

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



LUẬN VĂN THẠC SĨ  
LÊ NGUYỄN ANH HUY

**NHẬN DẠNG VÀ ĐỊNH DANH KHUÔN MẶT NGƯỜI  
THỜI GIAN THỰC VÀ SỬ DỤNG CAMERA 2D GIÁ RẺ**

NGÀNH: KỸ THUẬT ĐIỆN TỬ



Tp. Hồ Chí Minh, tháng 10/2017

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

LUẬN VĂN THẠC SĨ  
LÊ NGUYỄN ANH HUY

**NHẬN DẠNG VÀ ĐỊNH DANH KHUÔN MẶT NGƯỜI  
THỜI GIAN THỰC VÀ SỬ DỤNG  
CAMERA 2D GIÁ RẺ**

NGÀNH : KỸ THUẬT ĐIỆN TỬ - 60520203

Tp. Hồ Chí Minh, tháng 10/2017

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

LUẬN VĂN THẠC SĨ  
LÊ NGUYỄN ANH HUY

NHẬN DẠNG VÀ ĐỊNH DANH KHUÔN MẶT NGƯỜI  
THỜI GIAN THỰC VÀ SỬ DỤNG  
CAMERA 2D GIÁ RẺ

NGÀNH: KỸ THUẬT ĐIỆN TỬ - 60520203

Hướng dẫn khoa học:

TS. NGUYỄN VĂN THÁI

Tp. Hồ Chí Minh, tháng 10/2017

## QUYẾT ĐỊNH

### Về việc giao đề tài luận văn tốt nghiệp và người hướng dẫn năm 2017 HIỆU TRƯỞNG TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

Căn cứ Quyết định số 118/2000/QĐ-TTg ngày 10 tháng 10 năm 2000 của Thủ tướng Chính phủ về việc thay đổi tổ chức của Đại học Quốc gia TP. Hồ Chí Minh, tách Trường Đại học Sư phạm Kỹ thuật TP. Hồ Chí Minh trực thuộc Bộ Giáo dục và Đào tạo

Căn cứ Quyết định số 70/2014/QĐ-TTg ngày 10/12/2014 của Thủ tướng Chính phủ về ban hành Điều lệ trường Đại học

Căn cứ Thông tư số 15/2014/TT-BGDDT ngày 15/5/2014 của Bộ Giáo dục và Đào tạo về việc Ban hành Qui chế đào tạo trình độ thạc sĩ;

Căn cứ vào Biên bản bảo vệ Chuyên đề của ngành Kỹ thuật điện tử vào ngày 18/02/2017;

Xét nhu cầu công tác và khả năng cán bộ;

Xét đề nghị của Trưởng phòng Đào tạo,

### QUYẾT ĐỊNH:

**Điều 1.** Giao đề tài Luận văn tốt nghiệp thạc sĩ và người hướng dẫn Cao học năm 2017 cho:

Học viên : Lê Nguyễn Anh Huy MSHV: 1520705

Ngành : Kỹ thuật điện tử

Tên đề tài : Nhận dạng và định danh khuôn mặt người thời gian thực và sử dụng camera 2D giá rẻ

Người hướng dẫn : TS. Nguyễn Văn Thái

Thời gian thực hiện: từ ngày 27/02/2017 đến ngày 27/8/2017

**Điều 2.** Giao cho Phòng Đào tạo quản lý, thực hiện theo đúng Qui chế đào tạo trình độ thạc sĩ của Bộ Giáo dục & Đào tạo ban hành.

**Điều 3.** Trưởng các đơn vị, phòng Đào tạo, các Khoa quản ngành cao học và các Ông (Bà) có tên tại Điều 1 chịu trách nhiệm thi hành quyết định này.

Quyết định có hiệu lực kể từ ngày ký./. ✓

Nơi nhận :

- BGH (để biết);
- Như điều 2, 3;
- Lưu: VT, SĐH (3b).



**BIÊN BẢN CHẤM LUẬN VĂN TỐT NGHIỆP THẠC SĨ\_NĂM 2017**  
**NGÀNH: KỸ THUẬT ĐIỆN TỬ\_KHÓA 2015 - 2016 A**

Hội đồng chấm LVTN theo QĐ số: 1708/QĐ-DHSPKT-SĐH, ngày 12/10/2017

Có mặt : ..... **5** ..... Vắng mặt: ..... **0** .....

Chủ tịch Hội đồng : TS. Lê Mỹ Hà

Thư ký Hội đồng : TS. Nguyễn Thị Lưỡng

Học viên bảo vệ LVTN : Lê Nguyễn Anh Huy

MSHV: 1520705

Giảng viên hướng dẫn : TS. Nguyễn Văn Thái

Giảng viên phản biện : PGS.TS. Nguyễn Thanh Phương

TS. Vũ Quang Huy

Tên đề tài LVTN : *Nhận dạng và định danh khuôn mặt người thời gian thực và sử dụng camera 2D giá rẻ*

**I. KẾT QUẢ BẢO VỆ:**

TT	Thành viên hội đồng	Kết quả bảo vệ	Ghi chú
1	TS. Lê Mỹ Hà	6,5	
2	TS. Nguyễn Thị Lưỡng	6,5	
3	PGS.TS. Nguyễn Thanh Phương	6,0	
4	TS. Vũ Quang Huy	7,1	
5	PGS.TS. Dương Hoài Nghĩa	6,0	
<b>Tổng điểm</b>		<b>32,1</b>	
<b>Điểm trung bình</b>		<b>6,42</b>	

**II. KẾT LUẬN:**

(Thư ký hội đồng ghi rõ các ý kiến của thành viên hội đồng về việc chỉnh sửa, bổ sung những nội dung gì trong LVTN)

*Thạc sĩ Lê Mỹ Hà: Mục đích nghiên cứu là để xác định khuôn mặt người thời gian thực và sử dụng camera 2D giá rẻ.*

*Đề tài có ý nghĩa khoa học và ứng dụng.*

*Số thành phần tổng quan về tính hình nghiên cứu với những kết quả đạt được.*

*đã được công bố!*

*Làm thêm câu hỏi nghiên cứu để đánh giá phong cách*

*để xác định giá trị độ chính xác*

*Bổ sung tài liệu tham khảo chi tiết hơn*

Tp.Hồ Chí Minh, ngày 22 tháng 10 năm 2017

**THƯ KÝ HỘI ĐỒNG**

(Ký, ghi rõ họ và tên)

*TS. Lê Mỹ Hà*

*TS. Nguyễn Thị Lưỡng*



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

**PHIẾU NHẬN XÉT LUẬN VĂN THẠC SỸ**  
(Dành cho giảng viên phản biện)

**Tên đề tài luận văn thạc sĩ:** Nhận dạng và định danh khuôn mặt người thời gian thực và sử dụng camera 2D giá rẻ

**Tên tác giả:** LÊ NGUYỄN ANH HUY

**MSHV:** 1520705

**Ngành:** Kỹ thuật điện tử

**Khóa:** 2015

**Định hướng:** Ứng dụng

**Họ và tên người phản biện:** PGS.TS.Nguyễn Thanh Phương

**Cơ quan công tác:** Trường Đại học Công nghệ TPHCM

**Điện thoại liên hệ:** 0932757142

**I. Ý KIẾN NHẬN XÉT**

**1. Về hình thức & kết cấu luận văn:**

Luận văn gồm 66 trang chia làm 4 chương, hình thức trình bày gọn gàng định dạng hợp lý. Kết cấu hợp lý

**2. Về nội dung:**

**2.1. Nhận xét về tính khoa học, rõ ràng, mạch lạc, khúc chiết trong luận văn**

Luận văn trình bày mục tiêu rõ ràng, cơ sở lý thuyết phù hợp với nội dung thực hiện, logic chặt chẽ.

**2.2. Nhận xét đánh giá việc sử dụng hoặc trích dẫn kết quả NC của người khác có đúng qui định hiện hành của pháp luật sở hữu trí tuệ**

Việc trích dẫn tài liệu tham khảo đầy đủ và đúng qui định của luật sở hữu trí tuệ

**2.3. Nhận xét về mục tiêu nghiên cứu, phương pháp nghiên cứu sử dụng trong LVTN**

Mục tiêu nghiên cứu rõ ràng, phương pháp nghiên cứu hợp lý

**2.4. Nhận xét Tổng quan của đề tài**

Tổng quan đề tài trình bày khá tốt

**2.5. Nhận xét đánh giá về nội dung & chất lượng của LVTN**

Luận văn trình bày lý do thực hiện và cơ sở lý thuyết để thực hiện đề tài hợp lý. Tập mẫu dữ liệu đủ lớn để kiểm chứng. Nhìn chung đề tài đã đáp ứng yêu cầu của luận văn thạc sĩ kỹ thuật

**2.6. Nhận xét đánh giá về khả năng ứng dụng, giá trị thực tiễn của đề tài**

Đề tài có thể hoàn thiện và ứng dụng trong bài toán nhận dạng và định danh khuôn mặt trong hệ thống quản lý nhân viên

**2.7. Luận văn cần chỉnh sửa, bổ sung những nội dung gì (thiết sót và tồn tại):**

Cần bổ sung kết quả thực nghiệm và đánh giá độ chính xác để làm cơ sở kết luận

**II. CÁC VẤN ĐỀ CẦN LÀM RÕ**

(Các câu hỏi của giảng viên phản biện)

1. Độ chính xác trong định danh là bao nhiêu %?

2. Các yếu tố ảnh hưởng đến độ chính xác định danh?

**III. ĐÁNH GIÁ**

TT	Mục đánh giá	Đánh giá	
		Đạt	Không đạt
1	Tính khoa học, rõ ràng, mạch lạc, khúc chiết trong luận văn	x	
2	Đánh giá việc sử dụng hoặc trích dẫn kết quả NC của người khác có đúng qui định hiện hành của pháp luật sở hữu trí tuệ	x	
3	Mục tiêu nghiên cứu, phương pháp nghiên cứu sử dụng trong LVTN	x	
4	Tổng quan của đề tài	x	
5	Đánh giá về nội dung & chất lượng của LVTN	x	
6	Đánh giá về khả năng ứng dụng, giá trị thực tiễn của đề tài	x	

Đánh dấu chéo (x) vào ô muốn Đánh giá

### III. KẾT LUẬN

(Giảng viên phản biện ghi rõ ý kiến “Tán thành luận văn” hay “Không tán thành luận văn”)

Tán thành luận văn

TP.HCM, ngày tháng năm

**Người nhận xét**

(Ký & ghi rõ họ tên)

PGS.TS.Nguyễn Thanh Phương



## PHIẾU NHẬN XÉT LUẬN VĂN THẠC SỸ

(Dành cho giảng viên phản biện)

**Tên đề tài luận văn thạc sĩ:** Nhận dạng và định danh khuôn mặt người thời gian thực và sử dụng camera 2D giá rẻ

**Tên tác giả:** LÊ NGUYỄN ANH HUY

**MSHV:** 1520705

**Ngành:** Kỹ thuật điện tử

**Khóa:** 2015

**Định hướng:** Ứng dụng

**Họ và tên người phản biện:** TS.Vũ Quang Huy

**Cơ quan công tác:** Khoa Đào tạo chất lượng cao

**Điện thoại liên hệ:**

### I. Ý KIẾN NHẬN XÉT

#### 1. Về hình thức & kết cấu luận văn:

Luận văn bao gồm 4 chương và nội dung trình bày rõ ràng và tuân theo quy định.

#### 2. Về nội dung:

##### 2.1. Nhận xét về tính khoa học, rõ ràng, mạch lạc, khúc chiết trong luận văn

Nội dung trình bày rõ ràng súc tích. Một số lỗi chính tả và văn phong còn tồn tại trong nội dung luận văn

##### 2.2. Nhận xét đánh giá việc sử dụng hoặc trích dẫn kết quả NC của người khác có đúng qui định hiện hành của pháp luật sở hữu trí tuệ

Tác giả đã tham khảo các công trình và kết quả nghiên cứu trong và ngoài nước. Tuy nhiên các tài liệu trích dẫn chưa có tính cập nhật mới và nhiều.

##### 2.3. Nhận xét về mục tiêu nghiên cứu, phương pháp nghiên cứu sử dụng trong LVTN

Tác giả có phương pháp nghiên cứu hợp lý bao gồm thu thập tài liệu, tổng hợp, phân tích, đề xuất giải thuật, mô phỏng. Tuy nhiên kết quả mô phỏng, phân tích kết quả còn hạn chế, chưa có sự so sánh với các phương pháp khác.

##### 2.4. Nhận xét Tổng quan của đề tài

Phần tổng quan được trình bày rõ ràng, tuy nhiên có ít tài liệu được trích dẫn. Các nhận xét ưu nhược điểm của các thuật toán còn mang tính chung chung chưa trích dẫn rõ ràng và cụ thể.

##### 2.5. Nhận xét đánh giá về nội dung & chất lượng của LVTN

Tác giả đã đề xuất ứng dụng thuật toán kết hợp SHIFT và RANSAC trong xử lý nhận dạng khuôn mặt người, các kết quả đạt được là đáng trân trọng tuy nhiên còn giới hạn do nội dung mô phỏng, so sánh và phân tích kết quả còn hạn chế.

##### 2.6. Nhận xét đánh giá về khả năng ứng dụng, giá trị thực tiễn của đề tài

Tác giả đã chỉ ra các kết quả đạt được và một phần các hạn chế cần khắc phục

##### 2.7. Luận văn cần chỉnh sửa, bổ sung những nội dung gì (thiết sót và tồn tại):

- Nội dung trình bày còn ít và kết quả mô phỏng thực nghiệm chưa phong phú để đánh giá chính xác chất lượng thuật toán
- Một số hình trong chương 2 không đánh số

### II. CÁC VẤN ĐỀ CẦN LÀM RÕ

(Các câu hỏi của giảng viên phản biện)

1 Trong phần kết luận tác giả kết luận là thuật toán đề xuất cho tốc độ nhận dạng nhanh, tác giả hãy chứng minh cụ thể kết luận trên ( liên quan đến thời gian xử lý thuật toán trong máy tính).

2 Trình bày các giải pháp để giải quyết hạn chế của thuật toán đề xuất về cường độ ánh sáng

### III. ĐÁNH GIÁ

TT	Mục đánh giá	Đánh giá	
		Đạt	Không đạt
1	Tính khoa học, rõ ràng, mạch lạc, khúc chiết trong luận văn	x	
2	Đánh giá việc sử dụng hoặc trích dẫn kết quả NC của người khác có đúng qui định hiện hành của pháp luật sở hữu trí tuệ	x	
3	Mục tiêu nghiên cứu, phương pháp nghiên cứu sử dụng trong LVTN	x	
4	Tổng quan của đề tài	x	
5	Đánh giá về nội dung & chất lượng của LVTN	x	
6	Đánh giá về khả năng ứng dụng, giá trị thực tiễn của đề tài	x	

Đánh dấu chéo (x) vào ô muốn Đánh giá

### III. KẾT LUẬN

(Giảng viên phản biện ghi rõ ý kiến “**Tán thành luận văn**” hay “**Không tán thành luận văn**”)

Tán thành luận văn

TP.HCM, ngày tháng năm

**Người nhận xét**

(Ký & ghi rõ họ tên)



TS. Vũ Quang Huy

## LÝ LỊCH KHOA HỌC

## I. LÝ LỊCH SƠ LUỐC:

Họ & tên: Lê Nguyễn Anh Huy Giới tính: Nam  
Ngày, tháng, năm sinh: 16/05/1989 Nơi sinh: Quảng Nam  
Quê quán: Tam Kỳ - Quảng Nam Dân tộc: Kinh  
Chỗ ở riêng hoặc địa chỉ liên lạc: 130/3/13 Đường 2 – Tô 5 – Khu phố 9 –  
Phường Trường Thọ - Quận Thủ Đức – TP Hồ Chí Minh  
Điện thoại cơ quan: Điện thoại nhà riêng: 0938269304  
Fax: E-mail: anhhuyspkt@gmail.com

## **II. QUÁ TRÌNH ĐÀO TẠO:**

## 1. Trung học chuyên nghiệp:

Hệ đào tạo: Thời gian đào tạo từ ...../..... đến ...../ .....

Nơi học (trường, thành phố):

## Ngành hoc:

## 2. Đai học:

Hệ đào tạo: Chính quy Thời gian đào tạo từ 9/ 2008 đến 7/ 2012

Nơi học (trường, thành phố): Đại học Sư phạm Kỹ thuật TP HCM

## Ngành học: Kỹ thuật Điện - Điện tử

Tên đồ án, luận án hoặc môn thi tốt nghiệp: Thiết kế và thi công mô hình robot bám đối tượng

Ngày & nơi bảo vệ đồ án, luận án hoặc thi tốt nghiệp: Tháng 7/2012 Tại Đại học Sư Phạm Kỹ Thuật TP HCM

Người hướng dẫn: ThS. Trần Mạnh Sơn

### 3. Thạc sĩ:

Hệ đào tạo: Chính quy Thời gian đào tạo từ 05/2015 đến 10/2017

Nơi học (trường, thành phố): Đại học Sư Phạm Kỹ Thuật TP HCM

## Ngành học: Kỹ thuật Điện tử

Tên luận văn: Nhận dạng và định danh khuôn mặt người thời gian thực và sử dụng camera 2D giá rẻ

Ngày & nơi bảo vệ luận văn : 22/10/2017 tại Đại học Sư Phạm Kỹ Thuật  
TP. HCM

Người hướng dẫn: TS. Nguyễn Văn Thái

### **III. QUÁ TRÌNH CÔNG TÁC CHUYÊN MÔN KỂ TỪ KHI TỐT NGHIỆP ĐẠI HỌC:**

Thời gian	Nơi công tác	Công việc đảm nhiệm
Từ 6/2013 đến nay	Công ty Spitfire Controls VN	Kỹ sư Test

## LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

*Tp. Hồ Chí Minh, ngày 24 tháng 09 năm 2017*

Học viên thực hiện

**Lê Nguyễn Anh Huy**

## LỜI CẢM ƠN

Trước hết tôi xin gửi lời cảm ơn sâu sắc đến thầy TS. Nguyễn Văn Thái, người đã giúp đỡ tôi rất nhiều về định hướng nghiên cứu, hướng dẫn cho tôi trong suốt thời gian thực hiện đề tài này.

Tôi xin bày tỏ lời cảm ơn sâu sắc đến những cô giáo đã giảng dạy tôi trong thời gian học tại trường, những kiến thức mà tôi nhận được từ các thầy cô sẽ là hành trang giúp tôi vững bước trong tương lai.

Tôi cũng muốn gửi lời cảm ơn đến các anh chị và các bạn trong lớp đã giúp đỡ và cho tôi những lời khuyên bổ ích về chuyên môn trong quá trình nghiên cứu.

Vì kiến thức và thời gian có hạn nên chắc chắn sẽ không tránh khỏi những sai sót, tôi mong thầy cô và các bạn đóng góp những ý kiến quý báu để luận văn được hoàn thiện hơn.

Học viên thực hiện

**Lê Nguyễn Anh Huy**

## MỞ ĐẦU

Trong nền khoa học và kỹ thuật hiện nay, các công nghệ và kỹ thuật số đóng vai trò rất quan trọng trong việc hỗ trợ con người trong công việc và các vấn đề xã hội, đặc biệt là về xử lý ảnh. Lĩnh vực xử lý ảnh vẫn còn là một ngành khoa học rất mới mẻ so với các ngành khoa học khác nhưng nó đã là một lĩnh vực thu hút rất đông đảo nhà khoa học quan tâm và phát triển.

Công nghệ xử lý ảnh số đã hỗ trợ đắc lực trong rất nhiều các lĩnh vực như: giám sát an ninh, nhận dạng đối tượng, nhận dạng khuôn mặt, phát hiện chuyển động, theo dõi chuyển động, nhận dạng các khối u trong y học, hiệu chỉnh các ảnh và video,...

Nhận dạng khuôn mặt là một trong những ứng dụng rất phổ biến của công nghệ xử lý ảnh. Ứng dụng này được sử dụng trong các khu vực kiểm tra an ninh. Với việc đã được ứng dụng rộng rãi như vậy, nhận dạng khuôn mặt là một trong những đề tài rất thích hợp để các sinh viên và học viên bước đầu nghiên cứu và tìm hiểu về xử lý ảnh.

Hệ thống nhận dạng mặt người bao gồm hai bước: phát hiện khuôn mặt và định danh đối tượng. Công việc chính dựa vào các kỹ thuật rút trích đặc trưng cục bộ bất biến từ ảnh đối tượng và thực hiện đối sánh để định danh. Hiệu quả của hệ thống nhận dạng phụ thuộc vào các phương pháp sử dụng.

Khóa luận “Nhận dạng và định danh khuôn mặt người thời gian thực và sử dụng camera 2D giá rẻ” nhằm tìm hiểu phương pháp trích chọn đặc trưng ảnh và dùng những đặc trưng này so sánh với tập cơ sở dữ liệu được lưu trữ trước đó để định danh. Khóa luận bao gồm các nội dung sau:

Chương 1. Tổng quan về nhận dạng khuôn mặt

Chương 2. Cơ sở lý thuyết thuật toán

Chương 3. Xây dựng chương trình

Chương 4: Kết luận

# MỤC LỤC

	<b>TRANG</b>
Trang tựa	TRANG
Quyết định giao đề tài	
Lý lịch khoa học .....	i
Lời cam đoan.....	iii
Lời cảm ơn .....	iv
Mở đầu .....	v
Mục lục.....	vi
Danh mục các chữ viết tắt và ký hiệu .....	ix
Danh sách các hình.....	x
Danh sách các bảng .....	xii
<b>CHƯƠNG 1 TỔNG QUAN .....</b>	<b>1</b>
1.1 Tổng quan về đề tài.....	1
1.2 Mục đích của đề tài.....	4
1.3 Nhiệm vụ và giới hạn của đề tài.....	4
1.3.1 Nhiệm vụ của đề tài: .....	4
1.3.2 Giới hạn của đề tài .....	4
1.4 Phương pháp nghiên cứu .....	4
1.5 Giới thiệu thuật toán .....	6
1.5.1. SIFT .....	6
1.5.2. SURF.....	6
<b>CHƯƠNG 2 CƠ SỞ LÝ THUYẾT THUẬT TOÁN .....</b>	<b>7</b>
2.1 Đặc trưng Haar Like .....	7
2.2 Integral Image .....	8
2.3 Phương pháp AdaBoost.....	9
2.4 Mô hình phân tầng cascade .....	10
2.5. Thuật toán SIFT [4] [5] .....	12
2.5.1. Giới thiệu .....	12

2.5.2. Các nghiên cứu liên quan.....	13
2.5.3 Phát hiện cực trị trong không gian tỉ lệ.....	15
2.5.3.1. Phát hiện cực trị địa phương.....	17
2.5.3.2. Tần suất lấy mẫu tỉ lệ .....	18
2.5.3.3. Tần suất lấy mẫu trong miền không gian .....	20
2.5.4. Định vị các Keypoint .....	21
2.5.4.1 Loại trừ các điểm có tính tương phản kém.....	22
2.5.4.2. Loại bỏ điểm dư thừa theo biên.....	23
2.5.5. Gán hướng.....	24
2.5.6. Bộ mô tả hình ảnh cục bộ .....	26
2.5.6.1.Bộ mô tả .....	27
2.5.6.2. Kiểm thử bộ mô tả.....	28
2.5.6.3. Độ nhạy với biến đổi Affine.....	29
2.5.6.4. So khớp với cơ sở dữ liệu lớn .....	30
2.5.7 Đôi sánh đặc trưng SIFT .....	31
2.5.7.1 Độ đo tương tự và độ đo khoảng cách.....	31
2.5.7.2 Đôi sánh đặc trưng cục bộ bắt biến .....	32
2.5.7.3 Độ đo tương đồng cho đặc trưng cục bộ bắt biến .....	33
2.5.8. Ứng dụng cho nhận dạng đối tượng.....	33
2.5.8.1.So khớp Keypoint.....	34
2.5.8.2. Hiệu quả của việc đánh số các điểm lân cận gần .....	35
2.5.8.3.Cụm biến đổi Hough .....	36
2.5.8.4. Giải pháp cho các thông số Affine .....	37
2.5.9. Ví dụ nhận dạng .....	39
2.6 Thuật toán SURF .....	41
2.6.1 Giới thiệu .....	41
2.6.2 Bộ mô tả SURF .....	42
2.6.2.1 Bộ dò Fast-Hessian .....	42
2.6.2.2 Gán hướng cho điểm nổi bật và mô tả đặc trưng SURF.....	43

2.6.2.3 So khớp đặc trưng .....	45
2.7 Thuật toán RANSAC .....	46
2.7.1 Giới thiệu .....	46
2.7.2 Phương pháp .....	47
2.7.3 Thuật toán .....	47
2.7.4 Thông số.....	49
2.7.5 Bài toán thử nghiệm : .....	53
<b>CHƯƠNG 3 XÂY DỰNG CHƯƠNG TRÌNH .....</b>	<b>59</b>
3.1 Quá trình nhận dạng và định danh khuôn mặt.....	59
3.2 Thuật toán phát hiện đối tượng .....	59
3.3 Cài đặt chương trình .....	61
3.4 Chương trình mô phỏng .....	62
3.5 Định danh khuôn mặt.....	65
3.6 Thực nghiệm .....	65
<b>CHƯƠNG 4 KẾT LUẬN.....</b>	<b>70</b>
4.1 Kết luận chung.....	70
4.2. Kiến nghị .....	70
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>71</b>

## DANH MỤC CÁC CHỮ VIẾT TẮT VÀ KÝ HIỆU

<i>Chữ viết tắt</i>	<i>Giải thích</i>
FLANN	Fast Library for Approximate Nearest Neighbour Search
ANN	Viết tắt của thuật ngữ “Approximative Nearest Neighbour”
CSDL	Cơ sở dữ liệu.
DoG	Viết tắt của thuật ngữ “Difference-of-Gaussian”:
Gaussian	Hàm Gauss (Biểu đồ của một hàm Gauss là một đường cong đối xứng ).
Keypoint	Những điểm đặc trưng dùng trong quá trình nhận dạng ảnh.
<i>kNN</i>	<i>k Nearest Neighbor</i>
NBNN	Viết tắt của thuật ngữ “Naive Bayes Nearest Neighbor”
RANSAC	Viết tắt của thuật ngữ “Random Sample Consensus”
SIFT	Viết tắt của thuật ngữ “Scale Invariant Feature Transform”
SURF	Viết tắt của thuật ngữ “Speeded Up Robust Features”

# DANH SÁCH CÁC HÌNH

HÌNH	TRANG
<b>Hình 1.1:</b> Sơ đồ nhận dạng khuôn mặt.....	5
<b>Hình 2.1:</b> Các đặc trưng Haar Like cơ bản.....	7
<b>Hình 2.2:</b> Các đặc trưng Haar Like mở rộng.....	8
<b>Hình 2.3:</b> Tính giá trị ảnh tích phân tại điểm có tọa độ (x, y).....	8
<b>Hình 2.4:</b> Tính nhanh giá trị của vùng ảnh D.....	9
<b>Hình 2.5:</b> Máy phân lớp AdaBoost.....	11
<b>Hình 2.6:</b> Mô tả hàm Gaussian và hàm Difference-of-Gaussian (DoG) .....	16
<b>Hình 2.7:</b> Phát hiện cực trị của hàm DoG .....	18
<b>Hình 2.8:</b> Số lượng mẫu tỷ lệ trên mỗi Octave.....	19
<b>Hình 2.9:</b> Thứ tự làm mịn cho mỗi Octave .....	20
<b>Hình 2.10:</b> Các giai đoạn lựa chọn các điểm Keypoint.....	22
<b>Hình 2.11:</b> Đồ thị độ nhiễu của ảnh.....	25
<b>Hình 2.12:</b> Hướng phân bố trên ảnh và bộ mô tả các điểm Keypoint.....	27
<b>Hình 2.13:</b> Độ rộng của bộ mô tả (góc 50 độ, đồ nhiễu ảnh 4%).....	29
<b>Hình 2.14:</b> Sự ổn định của việc phát hiện vị trí các Keypoint .....	29
<b>Hình 2.15:</b> Số lượng Keypoint trong cơ sở dữ liệu .....	31
<b>Hình 2.16:</b> Đối sánh 2 ảnh quay về đối sánh 2 điểm đặc trưng.....	32
<b>Hình 2.17:</b> Tỷ lệ khoảng cách từ điểm điểm lân cận tới điểm kế tiếp.....	35
<b>Hình 2.18:</b> Ví dụ minh họa về thuật toán SIFT .....	38
<b>Hình 2.19:</b> Ví dụ về sự nhận dạng đối tượng .....	40
<b>Hình 2.20:</b> Xấp xỉ đạo hàm cấp 2 hàm Gaussian bằng hộp lọc.....	43
<b>Hình 2.21:</b> Lọc Haar wavelet để tính sự ảnh hưởng trên hai hướng x và y .....	44
<b>Hình 2.22:</b> Vùng hình tròn xung quanh và hướng đại diện cho điểm đặc trưng .....	44
<b>Hình 2.23:</b> 4x4 hình vuông con xung quanh điểm đặc trưng .....	45
<b>Hình 2.24:</b> So khớp đặc trưng .....	46
<b>Hình 2.25:</b> Tỉ lệ outlier trong tập dữ liệu.....	52
<b>Hình 3.1:</b> Sơ đồ nhận dạng khuôn mặt .....	59

<b>Hình 3.2:</b> Lưu đồ phát hiện đối tượng .....	60
<b>Hình 3.3:</b> Tập dữ liệu khuôn mặt.....	61
<b>Hình 3.4:</b> So khớp 2 ảnh dùng SIFT.....	62
<b>Hình 3.5:</b> So khớp 2 ảnh dùng SIFT kết hợp RANSAC .....	63
<b>Hình 3.6:</b> Thực hiện đối sánh ảnh từ camera và ảnh được lưu trong cơ sở dữ liệu .	64
<b>Hình 3.7:</b> Hai ảnh chứa khuôn mặt không có ngoại cảnh ở ánh sáng bình thường .	66
<b>Hình 3.8:</b> Hai ảnh chứa khuôn mặt có ngoại cảnh .....	66
<b>Hình 3.9:</b> Hai ảnh chứa khuôn mặt ở ánh sáng tối .....	67
<b>Hình 3.10:</b> So khớp giữa hai ảnh bị xoay .....	67

## DANH SÁCH CÁC BẢNG

BẢNG	TRANG
<b>Bảng 1.1:</b> Tỷ lệ nhận dạng trên các tập dữ liệu .....	3
<b>Bảng 1.2:</b> So sánh kết quả giữa các công nghệ .....	3
<b>Bảng 3.1:</b> Tỷ lệ nhận dạng trên các tập dữ liệu .....	69

## CHƯƠNG 1

# TỔNG QUAN

### 1.1 Tổng quan về đề tài

Nhận dạng khuôn mặt là một trong những lĩnh vực mới của xử lý ảnh. Và ngày nay nhận dạng được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống như nhận dạng trong lĩnh vực thương mại, hay phát hiện tội phạm trong lĩnh vực an ninh, hay trong lĩnh vực xử lý video, hình ảnh. Hiện nay có rất nhiều các phương pháp nhận dạng khác nhau được xây dựng để nhận dạng một người cụ thể trong thế giới thực. Tuy nhiên việc nhận dạng được một người trong thế giới thực là vô cùng khó khăn, bởi vì để nhận dạng được ta phải xây dựng được tập cơ sở dữ liệu đủ lớn và việc xử lý dữ liệu lớn này đòi hỏi phải nhanh và chính xác. Nhiệm vụ đặt ra là nghiên cứu và xây dựng một chương trình sử dụng phương pháp nhận dạng có độ chính xác cao mà khối lượng và thời gian tính toán lại ít.

Hệ thống nhận dạng mặt người bao gồm hai bước: phát hiện khuôn mặt và định danh tự động đối tượng. Công việc chính dựa vào các kỹ thuật rút trích đặc trưng cục bộ bất biến từ ảnh đối tượng và thực hiện đối sánh để định danh tự động. Hiệu quả của hệ thống nhận dạng phụ thuộc vào các phương pháp sử dụng.

Các nghiên cứu liên quan :

1. ThS. Châu Ngân Khánh *Khoa Kỹ thuật-Công nghệ và Môi trường, Trường Đại học An Giang* và TS. Đoàn Thanh Nghị *Khoa Kỹ thuật-Công nghệ và Môi trường, Trường Đại học An Giang* với đề tài “**NHẬN DẠNG MẶT NGƯỜI VỚI GIẢI THUẬT HAAR LIKE FEATURE – CASCADE BOOSTED CLASSIFIERS VÀ ĐẶC TRƯNG SIFT**” được đăng trên tạp chí Khoa học Trường đại học An Giang năm 2014.[1]

Đề tài đã được tiến hành đánh giá hiệu năng của hệ thống nhận dạng mặt người sử dụng thuật toán Haar Like Feature – Cascade of Boosted Classifiers và các đặc trưng SIFT. Hệ thống nhận dạng này được cài đặt bằng ngôn ngữ lập trình C/C++, sử dụng thư viện mã nguồn mở OpenCV của Intel (Bradski & Kaehler,

2012; Laganière, 2011), trên một máy tính cá nhân chạy hệ điều hành Linux với bản phân phối Ubuntu. Bước phát hiện mặt người thu được từ camera (webcam) được thực hiện thông qua việc huấn luyện mô hình phân tầng với mỗi tầng là một mô hình AdaBoost sử dụng bộ phân lớp yếu là cây quyết định với các đặc trưng Haar-Like (hỗ trợ bởi opencv\_createsamples và opencv\_haartraining của OpenCV) trên tập ảnh (mặt người và không phải mặt người). Phương pháp của đề tài trên được đánh giá trên các tập dữ liệu kiểm chuẩn AT&T (ORL) được giới thiệu bởi AT&T Laboratories Cambridge (1994); Face94, Face95, Face96, Grimace được giới thiệu bởi Spacek (2007a; 2007b; 2007c; 2007d); Jaffe được giới thiệu bởi Lyons, Miyuki Kamachi, và Jiro Gyoba (1998). Đầu tiên, chương trình sử dụng mô hình phân tầng đã huấn luyện để phát hiện mặt người, rút trích ra khuôn mặt. Sau đó, đã sử dụng lớp SiftFeatureDetector và SiftDescriptorExtractor từ thư viện OpenCV để rút trích các đặc trưng SIFT của tất cả các ảnh khuôn mặt (không phải ảnh gốc) và lưu vào cơ sở dữ liệu SIFT. Giải thuật kNN được sử dụng để tìm hai láng giềng gần nhất và láng giềng đảo ngược hoặc một láng giềng gần nhất tương ứng với thuật toán nhận dạng dựa trên so khớp SIFT và NBNN. Nghi thức kiểm tra trong thực nghiệm lấy ngẫu nhiên 2/3 tập dữ liệu làm tập học (hay cơ sở dữ liệu đối tượng), 1/3 tập dữ liệu còn lại làm tập kiểm tra. Đề tài thực hiện việc kiểm thử 5 lần, sau đó tính trung bình cộng để xác định giá trị lỗi tổng thể. Tiếp theo, so sánh các kết quả thu được từ việc sử dụng giải thuật kNN, kNN đảo ngược, NBNN. Việc rút trích được khuôn mặt người từ ảnh gốc đã giảm lược khá nhiều số lượng SIFT của đối tượng, nhờ vậy hệ thống đã tăng tốc đáng kể quá trình nhận dạng và đạt được độ chính xác cao hơn. Với ảnh có kích thước 500x500 pixels thì có khoảng 2000 SIFT. Nhưng khi trích xuất được khuôn mặt người trong ảnh thì số lượng SIFT trung bình còn khoảng 200 SIFT. Số lượng SIFT giảm đi không những không làm giảm độ chính xác của chương trình mà còn làm cho độ chính xác được tăng lên, vì các đặc trưng không hữu ích hoặc làm ảnh hưởng xấu đến kết quả nhận dạng đối tượng đã được loại bỏ. Thực nghiệm cho thấy số lượng SIFT không hữu ích là lớn hơn rất nhiều so với SIFT có ý nghĩa trong nhận dạng. Kết quả nhận dạng trên các tập dữ liệu kiểm thử

**Bảng 1.1:** Tỷ lệ nhận dạng trên các tập dữ liệu

Tập dữ liệu	Số lượng phân lớp	Tổng số ảnh	Tỷ lệ nhận dạng		
			kNN (%)	kNN đảo ngược (%)	NBNN (%)
AT&T	40	400	97.62	93.86	99.37
Face94	152	3040	100.00	99.37	100.00
Face95	72	1440	96.21	92.25	98.83
Face96	151	3016	94.35	86.05	99.20
Grimace	18	360	99.00	99.17	100.00
Jaffe	10	213	97.18	90.00	100.00

2. Vinay .A, Avani S Rao, Vinay S Shekhar, Akshay Kumar C, K N Balasubramanya Murthy, S Natarajan **Feature Extraction using ORB-RANSAC for Face Recognition** , PES University and PES Institute of Technology India 2015 [13]

Bài báo đã nêu ra dựa trên số lượng so khớp phù hợp từ kết quả thu được sau khi áp dụng FLANN, hai ngưỡng so khớp được thiết lập để được coi là phù hợp hoặc không phù hợp. Các ngưỡng cho việc nhận dạng thu được bằng cách lấy tỷ số inliers được trả về bởi thuật toán RANSAC cho tổng số so khớp tốt được trả về bởi FLANN. Đối với một số lượng so khớp tốt cố định, tỷ lệ của inliers so với tổng số so khớp sẽ cao hơn. Hình ảnh thử nghiệm được phân loại nếu nó vượt quá ngưỡng của ràng buộc đó. Bảng dưới đây mô tả ngưỡng cho tất cả ba cách tiếp cận cùng với giới hạn của các kết hợp tốt..

Table1. Thresholds and bounds of good matches.

TECHNIQUE	GOODMATCHES	THRESHOLD
SIFT-RANSAC	< 20	0.5
	>20	0.35
SURF-RANSAC	< 20	0.45
	>20	0.4
ORB -RANSAC	< 20	0.
	>20	0.4

Table2. Comparison results of SIFT-RANSAC, SURF-RANSAC, ORB-RANSAC.

PARAMETERS/TECHNIQUE	SIFT-RANSAC	SURF-RANSAC	ORB-RANSAC
Average time per image in sec	0.9276	0.6854	0.3962
Accuracy in %	69.72	65.27	75.08

**Bảng 1.2:** So sánh kết quả giữa các công nghệ

## 1.2 Mục đích của đề tài

Đề tài sẽ thực hiện việc nhận dạng khuôn mặt nhờ các đặc trưng Haar-like và thuật toán AdaBoost và mô hình phân tầng Cascade để định vị khuôn mặt, áp dụng thuật toán trích đặc trưng SIFT kết hợp RANSAC để so khớp với khuôn mặt trên tập ảnh đã được chọn lọc trước, đồng thời so sánh kết quả nhận dạng để định danh được đối tượng trong ảnh.

## 1.3 Nhiệm vụ và giới hạn của đề tài

### 1.3.1 Nhiệm vụ của đề tài:

Đề tài sẽ thực hiện các nhiệm vụ sau:

- Nghiên cứu tìm hiểu giải thuật Haar-like, Cascade of Boosted Classifiers
- Nghiên cứu, tìm hiểu phương pháp trích đặc trưng SIFT
- Nghiên cứu tìm hiểu thuật toán RANSAC
- Mô phỏng

### 1.3.2 Giới hạn của đề tài

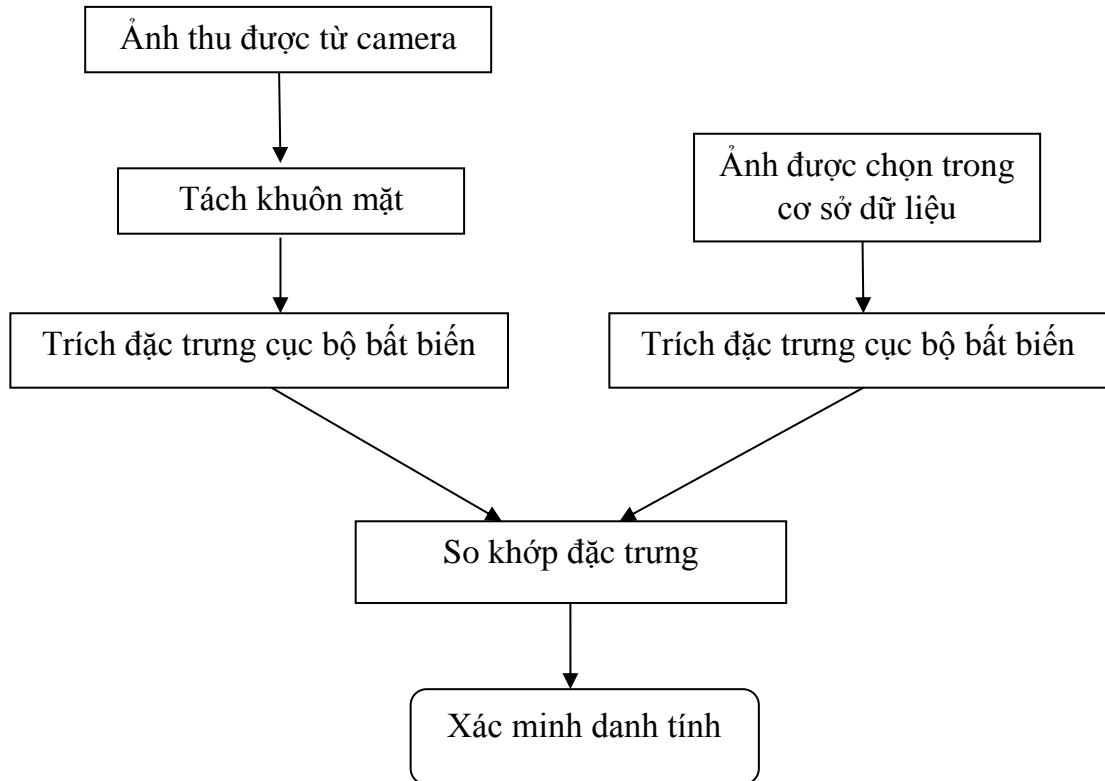
Do đề tài ở bước đầu nghiên cứu nên chỉ mới nhận dạng được khuôn mặt ở góc nhìn trực diện, góc nghiêng khuôn mặt nhỏ hơn khoảng 30 độ .

Trích các đặc trưng trên ảnh.

Định danh được đối tượng trong hình.

## 1.4 Phương pháp nghiên cứu

Vì điểm bất động là một dạng của bất biến nên các đặc trưng được trích chọn dựa vào các điểm bất động cũng bất biến nên nó thuận tiện trong việc so khớp và nhận dạng. Sau khi phát hiện các điểm quan tâm bất biến trong ảnh, bước tiếp theo là tính toán các đặc trưng dựa vào vị trí của các điểm bất động, bằng cách tạo ra các bộ mô tả cho các điểm này. Dựa trên bộ mô tả đã được xây dựng ta tiến hành so khớp giữa các đặc trưng của ảnh huấn luyện và ảnh đầu vào để đạt được kết quả mong muốn.

**Hình 1.1:** Sơ đồ nhận dạng khuôn mặt

Để thực hiện đề tài này, tác giả thực hiện các phương pháp nghiên cứu sau:

- Sử dụng các tài liệu tham khảo như sách giáo trình, các bài báo khoa học, các đồ án, luận văn tốt nghiệp của các trường đại học để nghiên cứu các cơ sở lý thuyết sử dụng trong đề tài
- Sử dụng các mã nguồn mở trên internet
- Mô phỏng các kết quả nghiên cứu

## 1.5 Giới thiệu thuật toán

### 1.5.1. SIFT

SIFT (Scale Invariant Feature Transform) là một phương pháp để chiết xuất các thuộc tính bất biến đặc biệt từ các hình ảnh và được sử dụng để thực hiện đối sánh tin cậy giữa các khung nhìn khác nhau của một đối tượng hay cảnh. Các thuộc tính này là bất biến đối với phép thay đổi tỉ lệ và phép quay ảnh và thể hiện rõ nét trong việc đối sánh một vùng con với phép biến đổi affine và sự thay đổi khung nhìn 3D cộng thêm nhiều và thay đổi trong chiều sáng. Các thuộc tính này rất đặc biệt và là một thuộc tính duy nhất có thể đối sánh chính xác trong một cơ sở dữ liệu lớn các thuộc tính trích xuất từ nhiều hình ảnh. Ngoài ra thuật toán này cũng được ứng dụng trong cách tiếp cận để nhận dạng đối tượng.

### 1.5.2. SURF

SURF (Speeded Up Robust Features) là bộ phát hiện và bộ mô tả các điểm quan tâm bất biến với tỷ lệ và góc xoay. Phương pháp này tương đương hoặc thậm chí nhanh hơn so với các phương pháp để xuất trước đây mà liên quan đến tính lặp đi lặp lại, tính riêng biệt và tính vững chắc, nó còn giúp việc tính toán và so sánh nhanh hơn.

SURF đạt được kết quả này bằng cách dựa trên những hình ảnh tích hợp có nhiều nếp cuộn hình ảnh thông qua việc xây dựng dựa trên các thế mạnh của các bộ phát hiện và bộ mô tả hàng đầu (ở đây sử dụng phương pháp ma trận Hessian để đo đạc cho bộ phát hiện và dựa trên phương pháp phân phôi cho các bộ mô tả) Bằng cách đơn giản hóa các phương pháp này sẽ cho ta các kết quả thiết yếu và dẫn tới việc liên kết các phát hiện và mô tả mới phù hợp.

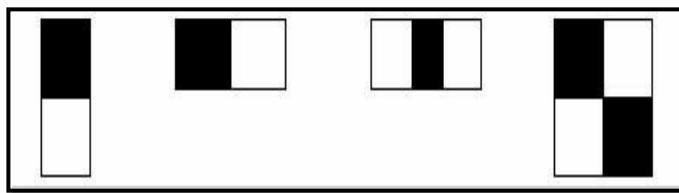
## CHƯƠNG 2

# CƠ SỞ LÝ THUYẾT THUẬT TOÁN

Trong luận văn này việc phát hiện khuôn mặt là sự kết hợp giữa một thuật toán tăng cường gọi là AdaBoost và đặc tính đáp ứng nhanh của các đặc trưng Haar. Đây là một phương pháp được xem là dựa trên cả hai cách phát hiện đối tượng dựa trên ảnh lẩn dựa trên đặc điểm hình học. Phương pháp này không chỉ sử dụng các thuật toán học (Learning Algorithm) để huấn luyện bộ phân lớp bằng các mẫu ví dụ đúng và không đúng mà người được chọn lựa trước cẩn thận, mà các đặc trưng được chọn ra bởi thuật toán hầu hết có liên quan trực tiếp đến các đặc trưng riêng biệt trên khuôn mặt người (độ tương phản của sóng mũi, vị trí cặp mắt ...). Kỹ thuật tăng tốc cải thiện hiệu suất của các bộ phân loại cơ sở bằng cách đặt lại trọng số cho các mẫu ví dụ dùng trong huấn luyện.

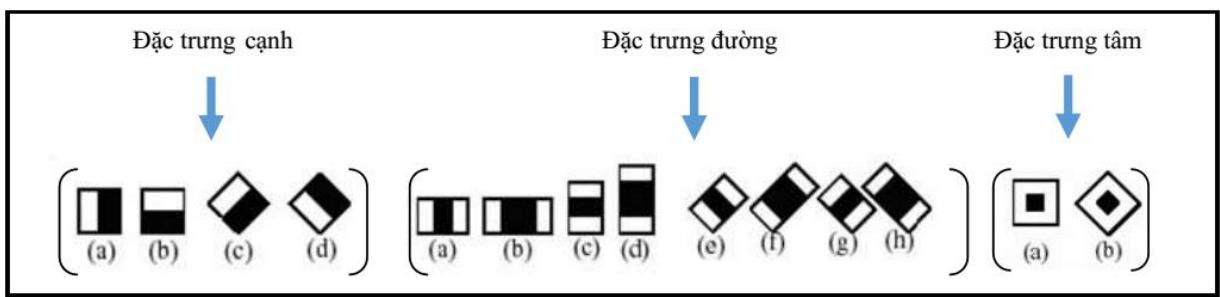
### 2.1 Đặc trưng Haar Like

Đặc trưng Haar Like được tạo thành bằng việc kết hợp các hình chữ nhật đen, trắng với nhau theo một trật tự, một kích thước nào đó dùng tính độ chênh lệch giữa các giá trị điểm ảnh trong các vùng kề nhau. Hình dưới đây mô tả 4 đặc trưng Haar Like cơ bản như sau:



**Hình 2.1:** Các đặc trưng Haar Like cơ bản

Để sử dụng các đặc trưng này cho việc phát hiện khuôn mặt, các đặc trưng Haar Like cơ bản trên được mở rộng (Lienhart, Kuranov, & Pisarevsky, 2002; Lienhart & Maydt, 2002) [2] thành các nhóm đặc trưng cạnh, đặc trưng đường và đặc trưng tâm.

**Hình 2.2:** Các đặc trưng Haar Like mở rộng

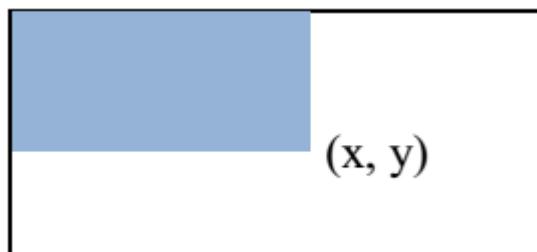
Dùng các đặc trưng trên, ta có thể tính được giá trị của đặc trưng Haar-like là sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng như trong công thức sau:

$$f(x) = \text{Tổng}_{\text{vùng đen}}(\text{các mức xám của pixel}) - \text{Tổng}_{\text{vùng trắng}}(\text{các mức xám của pixel})$$

Sử dụng giá trị này, so sánh với các giá trị của các giá trị pixel thô, các đặc trưng Haar-like có thể tăng hoặc giảm sự thay đổi bên trong hay bên ngoài lớp đồi tượng, do đó sẽ làm cho bộ phân loại dễ hơn. Để có thể tính nhanh các đặc trưng này, Viola và Jones (2001; 2004) giới thiệu khái niệm ảnh tích phân (Integral Image).

## 2.2 Integral Image

Integral Image [3] là một mảng hai chiều với kích thước bằng kích thước của ảnh cần tính giá trị đặc trưng Haar Like. Với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó. Bắt đầu từ vị trí trên bên trái đến vị trí dưới bên phải của ảnh, việc tính toán này đơn thuần chỉ dựa trên phép cộng số nguyên đơn giản, do đó tốc độ thực hiện rất nhanh. Dưới đây là mô tả cách tính ảnh tích phân.

**Hình 2.3:** Tính giá trị ảnh tích phân tại điểm có tọa độ (x, y)

Giá trị của ảnh tích phân tại điểm P có tọa độ (x,y) được tính như sau:

$$\text{ii}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{x}' \leq \mathbf{x}, \mathbf{y}' \leq \mathbf{y}} i(\mathbf{x}', \mathbf{y}') \quad (2.1)$$

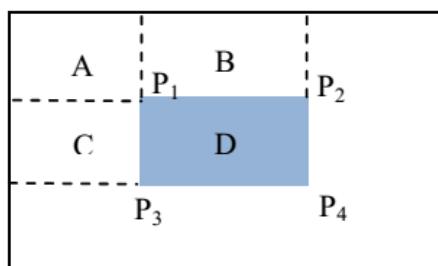
Trong đó  $\text{ii}(\mathbf{x}, \mathbf{y})$  là integral image và  $i(\mathbf{x}, \mathbf{y})$  là ảnh gốc

$$s(\mathbf{x}, \mathbf{y}) = s(\mathbf{x}, \mathbf{y} - 1) + i(\mathbf{x}, \mathbf{y})$$

$$\text{ii}(\mathbf{x}, \mathbf{y}) = \text{ii}(\mathbf{x} - 1, \mathbf{y}) + s(\mathbf{x}, \mathbf{y})$$

Với  $s(\mathbf{x}, -1) = 0$  và  $\text{ii}(-1, \mathbf{y}) = 0$

Dùng integral image việc tính tổng các giá trị mức xám của một vùng ảnh bất kỳ nào đó trên ảnh thực hiện theo cách sau, ví dụ tính giá trị của vùng D như sau:  $D = A + B + C + D - (A + B) - (A + C) + A$ .



**Hình 2.4:** Tính nhanh giá trị của vùng ảnh D

Ví dụ:

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	5

Ảnh ngõ vào

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	76

Integral image

Tiếp theo, sử dụng phương pháp máy học AdaBoost để xây dựng bộ phân loại mạnh với độ chính xác cao

### 2.3 Phương pháp AdaBoost

AdaBoost (Freund & Schapire, 1995) là một bộ phân loại mạnh phi tuyến phức, hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để tạo nên một bộ phân loại mạnh. AdaBoost sử dụng trọng số để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện cứ mỗi bộ phân loại yếu được xây dựng thì thuật

toán sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng bộ phân loại tiếp theo. Cập nhật bằng cách tăng trọng số của các mẫu nhận dạng sai và giảm trọng số của các mẫu được nhận dạng đúng bởi bộ phân loại yếu vừa xây dựng. Bằng cách này thì bộ phân loại sau có thể tập trung vào các mẫu mà bộ phân loại trước nó làm chưa tốt. Cuối cùng các bộ phân loại yếu sẽ được kết hợp lại tùy theo mức độ tốt của chúng để tạo nên một bộ phân loại mạnh.

Bộ phân loại yếu  $h_k$  được biểu diễn như sau:

$$h_k(x) = \begin{cases} 1 & \text{nếu } p_k f_k(x) < p_k \theta_k \\ 0 & \text{nếu ngược lại} \end{cases} \quad (2.2)$$

Với  $x$  là cửa sổ con cần xét

- $h_k$ : giá trị trả về của đặc trưng Haar-like thứ  $k$
- $p_k$ : hệ số chuẩn hóa
- $f_k$ : giá trị đặc trưng Haar-like thứ  $k$
- $\theta_k$ : ngưỡng

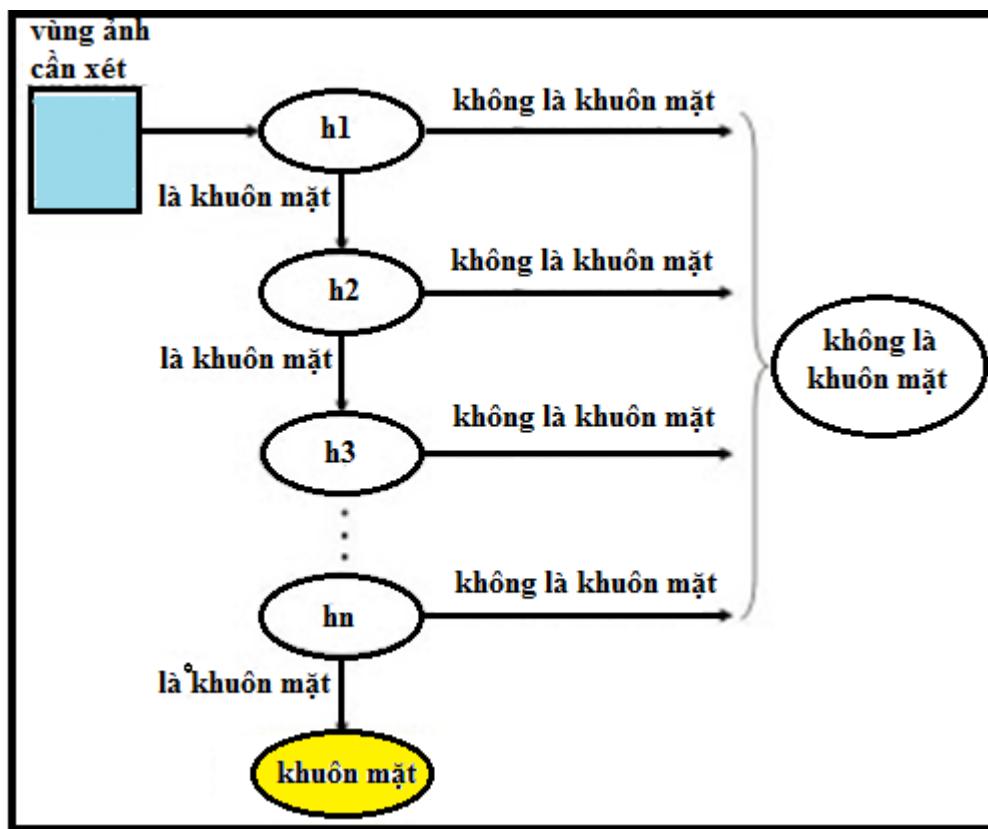
Công thức trên có thể được diễn giải như sau: nếu giá trị vector đặc trưng của mẫu cho bởi hàm  $f_k$  của bộ phân loại vượt qua một ngưỡng cho trước thì mẫu là object (đối tượng cần nhận dạng), ngược lại thì mẫu là background (không phải đối tượng).

#### 2.4 Mô hình phân tầng cascade

Cascade of Boosted Classifiers là mô hình phân tầng với mỗi tầng là một mô hình AdaBoost sử dụng bộ phân lớp yếu là cây quyết định với các đặc trưng Haar-Like.

Trong quá trình huấn luyện, bộ phân lớp phải duyệt qua tất cả các đặc trưng của mẫu trong tập huấn luyện. Việc này tốn rất nhiều thời gian. Tuy nhiên, trong các mẫu đưa vào, không phải mẫu nào cũng thuộc loại khó nhận dạng, có những mẫu background rất dễ nhận ra (gọi đây những mẫu background đơn giản). Đối với những mẫu này, chỉ cần xét một hay một vài đặc trưng đơn giản là có thể nhận dạng được chứ không cần xét tất cả các đặc trưng. Nhưng đối với các bộ phân loại thông thường thì cho dù mẫu cần nhận dạng là dễ hay khó nó vẫn phải xét tất cả các đặc

trung mà nó rút ra được trong quá trình học. Do đó, chúng tốn thời gian xử lý một cách không cần thiết.



**Hình 2.5:** Máy phân lớp AdaBoost

Mô hình Cascade of Classifiers được xây dựng nhằm rút ngắn thời gian xử lý, giảm thiểu nhận dạng lầm (false alarm) cho bộ phân loại. Cascade trees gồm nhiều tầng (stage hay còn gọi là layer), mỗi tầng là một mô hình AdaBoost với bộ phân lớp yếu là các cây quyết định. Một mẫu để được phân loại là đối tượng thì nó cần phải đi qua hết tất cả các tầng. Các tầng sau được huấn luyện bằng những mẫu âm negative (không phải mặt người) mà tầng trước nó nhận dạng sai, tức là nó sẽ tập trung học từ các mẫu background khó hơn, do đó sự kết hợp các tầng AdaBoost này lại sẽ giúp bộ phân loại giảm thiểu nhận dạng lầm. Với cấu trúc này, những mẫu background dễ nhận dạng sẽ bị loại ngay từ những tầng đầu tiên, giúp đáp ứng tốt nhất thời gian xử lý và vẫn duy trì được hiệu quả phát hiện khuôn mặt.

## 2.5. Thuật toán SIFT [4] [5]

### 2.5.1. Giới thiệu

Đối sánh một hình ảnh là một khía cạnh cơ bản của nhiều vấn đề trong thị giác máy tính bao gồm cả nhận dạng đối tượng hay cảnh và xử lý các cấu trúc 3D từ nhiều hình ảnh, âm thanh và theo dõi chuyển động. Trong một hình ảnh thì việc mô tả các thuộc tính mà làm cho chúng được nhận dạng trong các hình ảnh khác nhau của một đối tượng hay cảnh ở các khung nhìn khác nhau là vô cùng quan trọng. Các tính năng này là bất biến khi ta co giãn ảnh và xoay ảnh và một phần bất biến khi ta thay đổi trong chiếu sáng và hướng nhìn camera 3D. Chúng được định vị hóa tốt trong cả hai lĩnh vực không gian và miền tần số, giảm sự ảnh hưởng của sự lộn xộn trong hình ảnh hoặc nhiễu. Một số lượng lớn các thuộc tính có thể được chiết xuất từ các hình ảnh tiêu biểu với các thuật toán hiệu quả. Ngoài ra, các thuộc tính này là rất đặc biệt, trong đó cho phép một thuộc tính duy nhất có xác suất truy vấn cao đối với các thuộc tính trong một cơ sở dữ liệu lớn các thuộc tính và cung cấp một cơ sở cho nhận dạng đối tượng và bối cảnh.

Chi phí trích xuất các tính năng này được giảm thiểu bằng cách áp dụng phương pháp lọc cascade, trong đó các hoạt động tốn kém hơn chỉ được áp dụng tại các vị trí vượt qua kiểm tra ban đầu. Sau đây là các giai đoạn chính của tính toán được sử dụng để tạo ra các bộ các tính năng hình ảnh:

**Phát hiện cực trị Scale-Space:** Bước đầu tiên của tìm kiếm được tính trên tất cả các tỉ lệ và vị trí hình ảnh. Nó được thực hiện hiệu quả bằng cách sử dụng hàm DoG (Difference-of-Gaussian) để xác định các điểm quan tâm tiềm năng mà bất biến với các tỉ lệ và hướng.

**Định vị các Keypoint:** Tại mỗi điểm ứng viên địa phương sẽ có một mô hình chi tiết phù hợp để xác định vị trí và tỉ lệ. Keypoint được lựa chọn dựa trên sự ổn định của chúng trong các phép đo.

**Gán hướng:** Một hoặc nhiều hướng được gán cho mỗi keypoint cụt bộ dựa trên hướng gradient hình ảnh cục bộ. Mọi phép toán xử lý ở các bước sau này sẽ được thực hiện trên những dữ liệu ảnh đó đã được chuyển đổi liên quan đến phép

gán hướng và tỉ lệ địa phương hóa cho mỗi thuộc tính. Nhờ đó, tạo ra một sự bất biến trong các phép xử lý này.

**Bộ mô tả keypoint:** Các gradient cục bộ được chọn lựa trong các vùng xung quanh keypoint. Chúng được chuyển đổi thành đại diện địa phương quan trọng khi làm méo hình dạng và thay đổi trong chiều sáng. Cách tiếp cận này được đặt tên là các đặc trưng bất biến tỉ lệ (SIFT) vì nó biến đổi dữ liệu hình ảnh vào hệ tọa độ bất biến tỉ lệ liên quan đến các thuộc tính địa phương.

Với đối sánh ảnh và nhận dạng, các thuộc tính SIFT trước tiên được trích xuất từ một tập các ảnh tham chiếu và lưu trữ trong cơ sở dữ liệu. Một ảnh mới được đối sánh bằng cách so sánh các thuộc tính riêng lẻ từ ảnh mới với cơ sở dữ liệu và tìm thuộc tính đối sánh dựa trên khoảng cách Euclidean của các véc tơ thuộc tính. Thuật toán lảng giềng gần được sử dụng để có thể thực hiện các tính toán này nhanh chóng đối với cơ sở dữ liệu lớn.

Mỗi cụm Hough gồm ít nhất 3 thuộc tính giống với đối tượng và cần xác minh. Trước tiên một ước tính tối thiểu bình phương được thực hiện cho một xấp xỉ Affine với mỗi đối tượng. Bất kỳ thuộc tính hình ảnh nào khác phù hợp sẽ được nhận dạng và sự chênh lệch sẽ bị loại bỏ. Cuối cùng, ta sẽ có một tính toán chi tiết để tính xác suất để một tập hợp các thuộc tính chỉ ra sự hiện diện của một đối tượng, đem lại độ chính xác cho phép đối sánh. Đối sánh đối tượng qua các phép kiểm tra này có thể được xác định với độ tin cậy cao.

### 2.5.2. Các nghiên cứu liên quan

Việc phát triển đối sánh hình ảnh bằng cách sử dụng một tập hợp các điểm quan tâm địa phương có thể được truy ngược trở lại công việc của Moravec (1981) về việc sử dụng một máy dò góc. Các máy dò Moravec được cải thiện bằng cách Harris và Stephens (1988) làm cho nó có thể lắp lại nhiều hơn dưới các phép biến dạng hình ảnh nhỏ và gần biên. Harris cũng cho thấy hiệu quả của nó trong việc theo dõi chuyển động và khôi phục được cấu trúc 3D từ chuyển động (Harris, 1992), các góc dò Harris đã được sử dụng rộng rãi từ đó cho nhiều công việc đối sánh với hình ảnh khác. Các thiết bị dò thuộc tính này thường được gọi là máy dò

góc, họ không chỉ chọn góc mà hơn nữa là định vị bất kỳ hình ảnh có độ dốc lớn trong tất cả các hướng cùng ở cùng một tỉ lệ xác định.

Các máy dò góc Harris rất nhạy cảm với những thay đổi trong tỉ lệ ảnh, vì vậy nó không cung cấp một nền tảng tốt phù hợp với hình ảnh với kích cỡ khác nhau. Trước đó công trình của các tác giả (Lowe, 1999) cũng mở rộng cách tiếp cận thuộc tính cục bộ để đạt được tỉ lệ bất biến. Công việc này cũng mô tả một bộ mô tả địa phương mới cung cấp các thuộc tính đặc biệt hơn và ít nhạy cảm với biến dạng hình ảnh cục bộ như thay đổi khung nhìn 3D. Điều này cung cấp một nghiên cứu sâu hơn trong việc phân tích và trình bày một số cải tiến trong việc ổn định các thuộc tính bất biến.

Khung Affine cũng nhạy cảm với nhiều hơn so với các đặc điểm bất biến, vì vậy trong thực tế các thuộc tính Affine lặp lại ít hơn so với các đặc điểm bất biến trong biến dạng Affine với độ nghiêng 40 độ so với một bề mặt phẳng (Mikolajczyk, 2002). Hơn nữa bất biến Affine có thể không quan trọng đối với nhiều ứng dụng, ví dụ như thay đổi hướng nhìn là tốt nhất với vòng quay 30 độ trong khung nhìn (nghĩa là công nhận trong vòng 15 độ của điểm huấn luyện gần nhất) để nắm bắt những thay đổi không phẳng và các hiệu ứng tác động lên các đối tượng 3D.

Các phương pháp trên không phải là bất biến hoàn toàn, trong đó mô tả địa phương cho phép vị trí các tính năng tương đối chuyển đổi đáng kể với chỉ những thay đổi nhỏ trong mô tả. Cách tiếp cận này không chỉ cho phép các bộ mô tả được kết hợp chắc chắn trên một phạm vi bất biến affine đáng kể mà còn làm cho các tính năng mạnh hơn so với những thay đổi trong điểm nhìn 3D đối với bề mặt không phẳng. Mặt khác, sự bất biến affine là một tính chất có giá trị để so khớp các bề mặt phẳng với những thay đổi góc nhìn rất lớn và cần nghiên cứu sâu hơn về những cách tốt nhất để kết hợp điều này với bất biến không gian 3D một cách hiệu quả và ổn định. Một lớp các tính năng là những lớp có sử dụng đường viền hình ảnh hoặc ranh giới của vùng, điều này làm cho chúng không bị gián đoạn bởi các nền rườm rà gần ranh giới đối tượng. Matas và cộng sự, (2002) đã chỉ ra rằng các vùng cực trị ổn định

nhất có thể tạo ra một số lượng lớn các tính năng phù hợp với sự ổn định tốt. Mikolajczyk và cộng sự, (2003) đã phát triển một mô tả mới sử dụng các cạnh cục bộ trong khi bỏ qua các cạnh gần đó không liên quan, cung cấp khả năng tìm các tính năng ổn định thậm chí gần các ranh giới của các hình dạng hép chòng lên nền lộn xộn. Nelson và Selinger (1998) đã cho thấy kết quả tốt với các tính năng địa phương dựa trên các nhóm các đường viền hình ảnh. Tương tự, Pope và Lowe (2000) đã sử dụng các tính năng dựa trên việc xếp nhóm các đường viền hình ảnh có tính phân cấp đặc biệt hữu ích cho các đối tượng thiếu kết cấu chi tiết.

### **2.5.3 Phát hiện cực trị trong không gian tỉ lệ**

Như được mô tả trong phần giới thiệu, chúng ta sẽ phát hiện các keypoint bằng cách sử dụng một phương pháp lọc cascade sử dụng các thuật toán hiệu quả để xác định vị trí ứng cử viên sau đó được kiểm tra chi tiết hơn. Giai đoạn đầu tiên là phát hiện keypoint để tìm các khu vực và các tỉ lệ lặp đi lặp lại dưới các hướng nhìn khác nhau của cùng một đối tượng. Phát hiện địa điểm đó là bắt biến với tỉ lệ thay đổi của hình ảnh và có thể thực hiện bằng cách tìm kiếm các thuộc tính ổn định trên tất cả các tỉ lệ, có thể dùng một hàm liên tục của tỉ lệ được gọi là không gian tỉ lệ (Witkin, 1983). Nó đã được chứng minh bởi Koenderink (1984) và Lindeberg (1994) mà theo một loạt các giả định hợp lý thì chỉ có thể nhân rộng không gian là hàm Gaussian. Vì thế nên không gian tỉ lệ của một hình ảnh được định nghĩa như một hàm  $L(x, y, \sigma)$  được tạo ra từ phép nhân chập một biến tỉ lệ Gaussian  $G(x, y, \sigma)$  với một hình ảnh đầu vào  $I(x, y)$ :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.3)$$

Trong đó  $*$  là phép toán nhân chập giữa  $x, y$  và :

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.4)$$

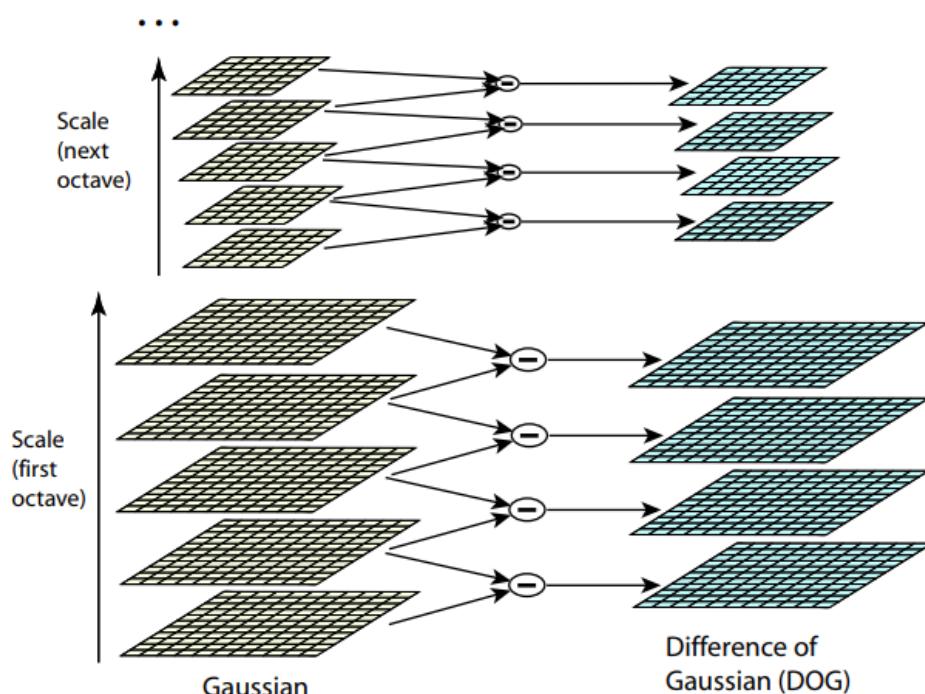
Để phát hiện địa điểm Keypoint ổn định và hiệu quả trong không gian tỉ lệ, Lowe đã đề xuất sử dụng không gian cực trị dùng các hàm Gaussian khác nhau với

các hình ảnh  $D(x, y, \sigma)$ , chúng có thể được tính toán từ sự khác biệt của hai tỉ lệ lân cận cách nhau bởi một số hằng số  $k$  không đổi:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (2.5)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma)$$

Có một số lý do cho việc lựa chọn hàm này. Đầu tiên nó là một hàm đặc biệt về hiệu suất để tính toán như những hình ảnh mịn  $L$  cần phải được tính toán trong bất kỳ bộ mô tả thuộc tính không gian tỉ lệ nào và  $D$  có thể được tính bằng cách đơn giản là trừ hình ảnh.



**Hình 2.6:** Mô tả hàm Gaussian và hàm Difference-of-Gaussian (DoG)

Ngoài ra, các hàm Gaussian khác nhau cung cấp một xấp xỉ gần Laplacian tỉ lệ. Bình thường Laplacian của Gaussian là  $\sigma^2 \nabla^2 G$  như nghiên cứu bởi Lindeberg (1994). Lindeberg cho thấy rằng Laplacian bình thường với các yếu tố  $\sigma^2$  là thực sự cần thiết cho tỉ lệ bất biến. Trong so sánh thử nghiệm chi tiết Mikolajczyk (2002) thấy rằng các cực đại và cực tiểu của  $\sigma^2 \nabla^2 G$  tạo nên các thuộc tính hình ảnh ổn định nhất so với một số các hàm hình ảnh khác chẳng hạn như gradient, Hessian hoặc hàm của góc Harris.

Mối quan hệ giữa D và  $\sigma^2 \nabla^2 G$  như sau:

$$\frac{dG}{d\sigma} = \sigma \nabla^2 G \quad (2.6)$$

Từ đây, chúng ta thấy rằng  $\nabla^2 G$  có thể được tính xấp xỉ để  $\partial G / \partial \sigma$  đạt sự khác biệt gần nhất về tỉ lệ tại  $k\sigma$  và  $\sigma$ :

$$\sigma \nabla^2 G = \frac{dG}{d\sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (2.7)$$

và do đó,

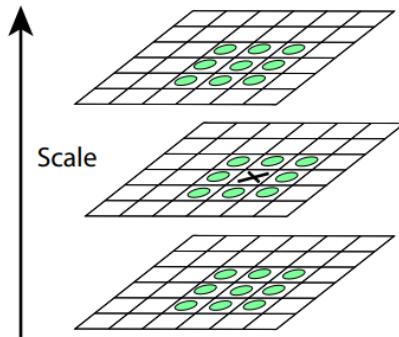
$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G \quad (2.8)$$

Điều này cho thấy rằng khi các hàm khác của hàm Gaussian có tỉ lệ khác nhau bởi một hằng số có quan hệ chặt chẽ với tỉ lệ  $\sigma^2$  cho tỉ lệ bát biến Laplacian. Các yếu tố ( $k - 1$ ) trong phương trình là một hằng số trên tất cả tỉ lệ và do đó không ảnh hưởng đến vị trí cực trị. Các lỗi xấp xỉ sẽ trả về 0 khi  $k$  tiến đến 1, nhưng trong thực tế, người ta đã tìm thấy rằng xấp xỉ gần như không có tác động đến sự ổn định của việc phát hiện cực trị hoặc địa phương hóa đối với sự khác biệt quan trọng về tỉ lệ, như  $k = \sqrt{2}$

Một cách tiếp cận hiệu quả để xây dựng  $D(x, y, \sigma)$  được thể hiện trong Hình 2.6. Hình ảnh ban đầu là từng bước kết hợp với Gaussian để tạo ra hình ảnh riêng biệt bởi hằng số  $k$  trong không gian tỉ lệ hiện xếp chồng lên nhau trong cột bên trái. Ở đây ta chọn cách phân chia từng octave của không gian tỉ lệ (tức là gấp đôi  $\sigma$ ) thành một số nguyên  $s$ , vì vậy  $k = 2^{m-1/s}$ . Chúng ta phải tạo ra  $s+3$  ảnh trong chồng hình ảnh mờ cho mỗi octave, vì thế cuối cùng việc phát hiện cực trị bao phủ một octave hoàn chỉnh. Tỉ lệ ảnh liền kề được trừ cho nhau để tạo sự khác biệt của ảnh Gaussian hiển thị bên phải. Khi một octave hoàn chỉnh đã được xử lý, chúng ta đổi mẫu hình Gaussian có giá trị khởi tạo gấp đôi  $\sigma$  (nó sẽ có 2 hình ảnh từ phía trên cùng của ngăn xếp) bằng cách lấy mỗi điểm ảnh thứ hai trong mỗi hàng và cột. Độ chính xác của mẫu so với  $\sigma$  là không có khác biệt so với thời điểm khởi tạo octave trước đó, trong khi các phép tính toán được giảm đi rất nhiều.

### 2.5.3.1. Phát hiện cực trị địa phương

Để phát hiện cực đại và cực tiểu địa phương của  $D(x, y, \sigma)$ , mỗi điểm mẫu được so sánh với tám điểm láng giềng của bức ảnh hiện tại và chín điểm láng giềng ở tỉ lệ trên và dưới (hình 2.7). Nó được chọn khi và chỉ khi nó lớn hơn tất cả các điểm láng giềng hoặc nhỏ hơn tất cả. Chi phí của việc kiểm tra này là khá thấp do thực tế hầu hết các điểm lấy mẫu sẽ được loại bỏ sau lần đầu kiểm tra.

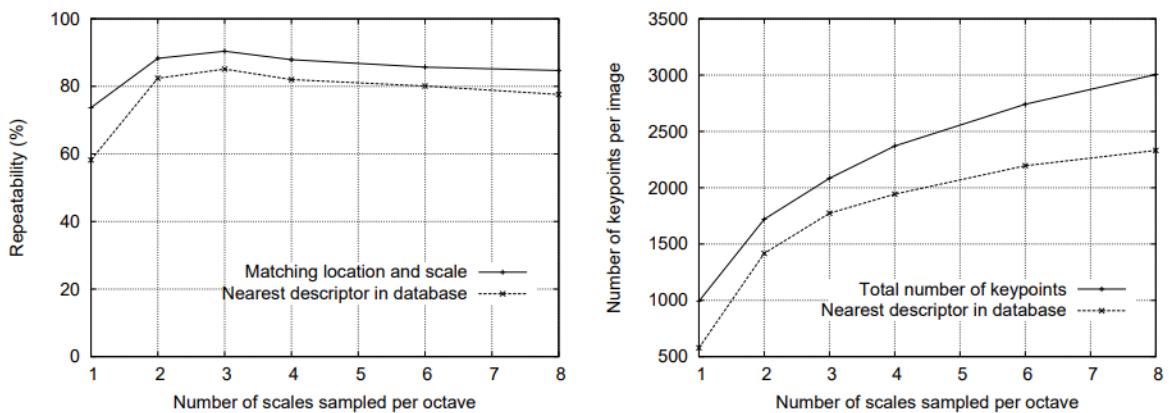


**Hình 2.7:** Phát hiện cực trị của hàm DoG

Vì vậy, chúng ta phải giải quyết một giải pháp chuyển đổi về hiệu năng. Trong thực tế, điều này có thể được minh chứng bằng các thí nghiệm. Các cực trị đó gần nhau là khá ổn định với những nhiễu loạn nhỏ của hình ảnh. Ta có thể xác định những thực nghiệm tốt nhất bằng cách nghiên cứu một loạt các tần số lấy mẫu và sử dụng các kết quả đáng tin cậy nhất trong một mô phỏng thực tế.

#### 2.5.3.2. Tần suất lấy mẫu tỉ lệ

Ta thực hiện việc đổi sánh dùng một bộ sưu tập 32 hình ảnh thực tế rất đa dạng, bao gồm cả ngoại cảnh, khuôn mặt người, hình ảnh trên không và hình ảnh công nghiệp (miền hình ảnh đã được tìm thấy hầu như không có ảnh hưởng đến bất kỳ kết quả nào). Mỗi hình ảnh sau đó đã phải chịu một loạt các biến đổi, bao gồm quay, thay đổi tỉ lệ, Affine, sự thay đổi về độ sáng và độ tương phản và bổ sung các nhiễu hình ảnh. Bởi vì những thay đổi này là tổng hợp, nó đã có thể dự đoán chính xác nơi mỗi thuộc tính trong một hình ảnh ban đầu sẽ xuất hiện trong hình ảnh chuyển đổi, cho phép đo lặp lại chính xác và độ chính xác vị trí cho mỗi thuộc tính.

**Hình 2.8:** Số lượng mẫu tỷ lệ trên mỗi Octave

Hình 2.8 cho thấy các kết quả mô phỏng được sử dụng để kiểm tra tác động của thay đổi số lượng tỉ lệ mỗi octave mà tại đó các chức năng chụp ảnh được lấy mẫu trước khi phát hiện cực trị. Trong trường hợp này, mỗi hình ảnh được lấy mẫu lại xoay sau bằng một góc ngẫu nhiên và nhân rộng bởi một số lượng ngẫu nhiên giữa 0,2 và 0,9 lần kích thước ban đầu. Keypoint từ các hình ảnh có độ phân giải giảm được đối sánh với những điểm đó từ các hình ảnh gốc vì thế tỉ lệ cho tất cả các keypoint được thể hiện trong ảnh đối sánh. Ngoài ra, 1% nhiều hình ảnh đã được bổ sung, nghĩa là mỗi điểm ảnh đã thêm vào một số ngẫu nhiên từ khoảng thống nhất [-0.01, 0.01] nơi các giá trị điểm ảnh nằm trong khoảng [0,1]

Dòng trên cùng trong đồ thị đầu tiên của Hình 2.8 cho thấy số phần trăm keypoint được phát hiện tại địa điểm đối sánh và tỉ lệ trong hình ảnh chuyển đổi. Đối với tất cả các ví dụ này, tỉ lệ đối sánh là  $\sqrt{2}$  của tỉ lệ chính xác và vị trí đối sánh là trong  $\sigma$  pixels,  $\sigma$  là tỉ lệ của các keypoint (định nghĩa phương trình (2.5) là độ lệch chuẩn của Gaussian nhỏ nhất được sử dụng trong hàm DOG). Các dòng thấp hơn trên biểu đồ này cho thấy số lượng các keypoint được đối sánh một cách chính xác đến một cơ sở dữ liệu gồm 40.000 keypoint sử dụng thủ tục đối sánh láng giềng gần để mô tả trong phần 2.1.6 (điều này cho thấy rằng một khi các keypoint được lặp đi lặp lại, nó có khả năng là hữu ích cho nhận dạng và phù hợp với nhiệm vụ đối sánh). Như biểu đồ này cho thấy, độ lặp lại cao nhất thu được khi lấy mẫu 3 thang mỗi octave.

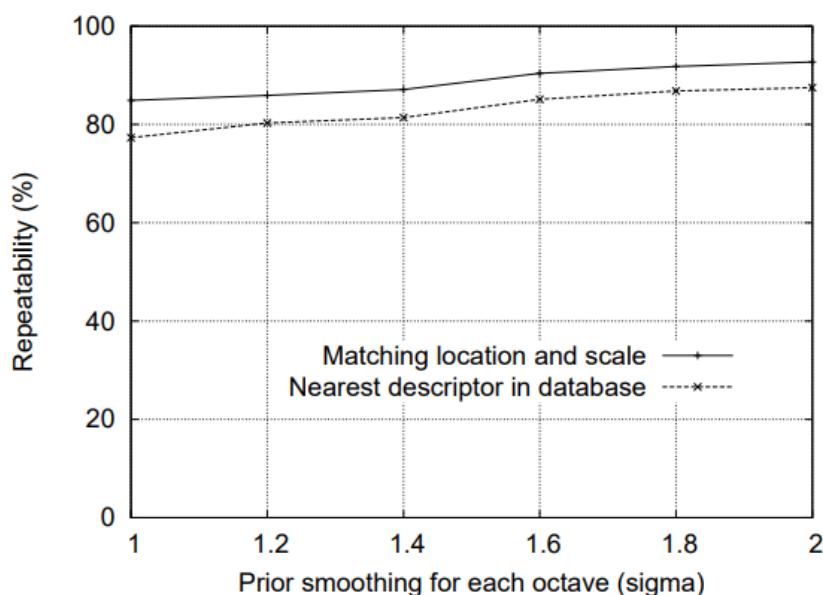
Số keypoint tăng lên với việc tăng tỉ lệ mẫu và tổng số các đối sánh đúng cũng

tăng. Từ thành công trong nhận dạng đối tượng thường phụ thuộc nhiều vào số lượng keypoint đối sánh đúng, và phần trăm đối sánh đúng cũng tăng, nhiều ứng dụng sẽ được tối ưu để sử dụng một số lượng lớn các mẫu tỉ lệ. Tuy nhiên, chi phí của việc tính toán cũng tăng lên với con số này, vì vậy mà ta lựa chọn sử dụng chỉ 3 mẫu tỉ lệ mỗi octave.

Các thí nghiệm cho thấy rằng hàm không gian tỉ lệ hàm DOG có một số lượng lớn các cực trị và nó sẽ rất tốn kém để phát hiện tất cả. Và điều may mắn là ta có thể phát hiện các tập con ổn định nhất và hữu ích ngay cả với một mẫu thô của tỉ lệ.

#### 2.5.3.3. Tần suất lấy mẫu trong miền không gian

Để xác định tần số lấy mẫu cho mỗi octave của không gian tỉ lệ thì phải xác định tần số lấy mẫu trong hình ảnh liên quan đến tỉ lệ của độ mịn. Giả sử rằng cực trị có thể được tự ý gần nhau, sẽ có một sự hoán đổi tương tự giữa tần số lấy mẫu và tỷ lệ phát hiện. Hình 2.9 cho thấy thực nghiệm của lượng làm mịn trước khi  $\sigma$  được áp dụng cho từng cấp hình ảnh trước khi xây dựng các không gian biểu diễn tỉ lệ cho một octave. Dòng trên cùng là lặp lại của phát hiện keypoint và kết quả cho thấy rằng khả năng lặp lại tiếp tục tăng với  $\sigma$ . Tuy nhiên, nếu chọn  $\sigma$  quá lớn thì lại mất nhiều thời gian, để tăng hiệu quả ta lựa chọn  $\sigma = 1.6$  cung cấp gần lặp lại tối ưu. Giá trị này đã được sử dụng cho các kết quả trong hình 2.8.



Hình 2.9: Thử tự làm mịn cho mỗi Octave

Tất nhiên, nếu ta làm mịn hình ảnh trước khi phát hiện cực trị, ta đang loại bỏ hiệu quả của các tần số không gian cao nhất. Vì vậy, để sử dụng đầy đủ các đầu vào, các hình ảnh có thể được mở rộng để tạo thêm nhiều điểm hơn mẫu đã có mặt trong bản gốc. Ta tiến hành nhân đôi kích thước của hình ảnh đầu vào sử dụng nội suy tuyến tính trước khi xây dựng các mức đầu tiên của kim tự tháp. Trong khi các hoạt động tương đương có thể có hiệu quả đã được thực hiện bởi việc dùng bộ lọc bù tập con điểm ảnh trên ảnh gốc, tăng gấp đôi hình ảnh dẫn đến việc thực hiện hiệu quả hơn. Ta giả định rằng các hình ảnh ban đầu có một vệt mờ tối thiểu  $\sigma = 0,5$  (mức tối thiểu cần thiết để ngăn chặn hiện tượng răng cưa tại đường biên ảnh), và do đó để tăng các điểm ảnh ta cần tăng gấp đôi giá trị  $\sigma = 1,0$ . Điều này có nghĩa rằng việc làm mịn bổ sung là cần thiết trước khi tạo ra các octave đầu tiên của không gian tỉ lệ. Việc tăng gấp đôi hình ảnh làm tăng số lượng các keypoint ổn định gần gấp 4.

#### 2.5.4. Định vị các Keypoint

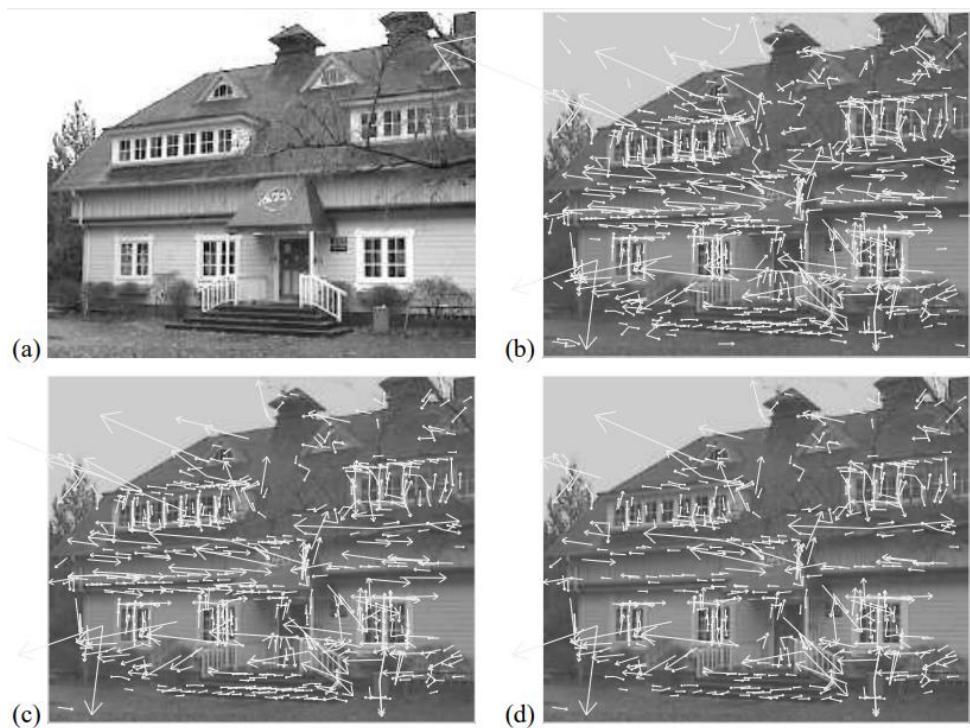
Khi một ứng viên keypoint đã được tìm thấy bằng cách so sánh một pixel với các điểm láng giềng của nó, bước tiếp theo là để thực hiện một cách chi tiết để các dữ liệu trong khu vực với vị trí, tỉ lệ và tỉ lệ của độ cong chính. Điều này cho phép các điểm được loại bỏ khi có độ tương phản thấp (và do đó nhạy cảm với nhiễu) hoặc ít được địa phương hóa dọc theo một cạnh.

Việc thực hiện ban đầu của phương pháp này (Lowe, 1999) chỉ đơn giản là định vị keypoint vào vị trí và tỉ lệ của các điểm mẫu trung tâm. Tuy nhiên, thời gian gần đây Brown đã phát triển một phương pháp (Brown và Lowe, 2002) cho một hàm bậc hai 3D vừa khít với các điểm lấy địa phương để xác định vị trí nội suy tối đa, và thí nghiệm của ông cho thấy rằng việc này cung cấp một sự cải thiện đáng kể phù hợp và ổn định. Cách tiếp cận của ông sử dụng các mở rộng Taylor (lên đến các phương trình bậc hai) của hàm tỉ lệ không gian,  $D(x, y, \sigma)$ , dịch chuyển sao mà nguồn gốc là ở vị trí mẫu:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (2.9)$$

Trong đó  $D$  và các dẫn xuất của nó được đánh giá ở vị trí mẫu và  $x = (x, y, \sigma)^T$  là phần bù đắp từ vị trí này. Các vị trí của các cực trị  $\hat{x}$  được xác định bằng cách lấy đạo hàm của hàm này đối với  $x$  và gán nó bằng 0, cho

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (2.10)$$



**Hình 2.10:** Các giai đoạn lựa chọn các điểm Keypoint

Theo đề xuất của Brown, Hessian và dẫn xuất của  $D$  được tính xấp xỉ bằng cách sử dụng những khác biệt của các điểm mẫu lân cận. Kết quả là hệ thống tuyến tính  $3 \times 3$  có thể được giải quyết với chi phí tối thiểu. Nếu phần bù lớn hơn 0,5 lần kích thước bất kỳ, điều đó có nghĩa là nó gần hơn với một mẫu khác. Trong trường hợp này, các điểm mẫu được thay đổi và suy diễn thay vì về điểm đó. Cuối cùng phần bù  $\hat{x}$  được thêm vào vị trí của điểm mẫu của nó để có được các ước tính nội suy cho vị trí của các cực trị.

#### 2.5.4.1 Loại trừ các điểm có tính tương phản kém

Các giá trị hàm tại cực trị  $D(\hat{x})$  rất hữu ích cho việc loại bỏ cực trị không ổn

định với độ tương phản thấp. Điều này có thể thu được bằng cách thay thế phương trình (2.10) vào (2.9), cho

$$\hat{D}(x) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (2.11)$$

Đối với các thí nghiệm này, tất cả các cực trị với một giá trị của  $|D(\hat{x})|$  ít hơn 0.03 sẽ bị loại bỏ (ta giả định các giá trị điểm ảnh trong khoảng [0,1]).

Hình 2.10 cho thấy những ảnh hưởng của lựa chọn keypoint trên một hình ảnh tự nhiên. Để tránh quá nhiều lọn xộn, một độ phân giải điểm ảnh thấp 233 x 189 được sử dụng và keypoint được hiển thị như là vectơ cho vị trí, tỉ lệ và hướng của mỗi keypoint (phân hướng được mô tả dưới đây). Hình 2.10(a) cho thấy những hình ảnh ban đầu được hiển thị ở độ tương phản giảm sau hình tiếp theo. Hình (b) hiển thị 832 keypoint trên tất cả các cực đại và cực tiểu tìm được của hàm DOG, trong đó hình (c) hiển thị 729 keypoint còn lại sau khi loại bỏ các giá trị  $d(x)$  nhỏ hơn 0.03.

#### 2.5.4.2. Loại bỏ điểm dư thừa theo biên

Sự ổn định không đủ để loại keypoint với độ tương phản thấp. Các hàm DOG sẽ có một đáp ứng mạnh mẽ dọc theo các biên, ngay cả khi các vị trí dọc theo các biên là khó xác định và do đó không ổn định với một lượng nhỏ của nhiễu.

Một điều khó định nghĩa trong hàm DOG sẽ có một độ cong chính lớn trên biên nhưng một lượng nhỏ theo hướng vuông góc. Các đường cong chính có thể được tính toán từ một ma trận Hessian  $H(2x2)$ , tính theo vị trí và tỉ lệ của các Keypoint:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2.12)$$

Các dẫn xuất được ước tính bằng cách lấy sự khác biệt của các điểm mẫu lân cận.

Các giá trị riêng của  $H$  là tỷ lệ thuận với độ cong chính của  $D$ . Từ cách tiếp cận được sử dụng bởi Harris và Stephens (1988), ta có thể tránh được việc tính toán

các giá trị đặc trưng, ta chỉ quan tâm đến tỷ lệ của chúng.

Cho  $\alpha$  là giá trị riêng với cường độ lớn nhất và  $\beta$  là nhỏ hơn. Sau đó, ta có thể tính tổng các giá trị đặc trưng từ các dấu 0.03, vết của H và kết quả từ việc xác định là:

$$\text{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta, \quad (2.13)$$

$$\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (2.14)$$

Trong trường hợp không chắc các yếu tố xác định là không tốt, độ cong có những dấu hiệu khác nhau thì điểm đó bị bỏ đi vì không có một cực trị. Cho  $r$  là tỷ số giữa độ lớn giá trị riêng lớn nhất và nhỏ hơn, do đó  $\alpha = r\beta$ . Vì vậy,

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r+1)^2}{r} \quad (2.15)$$

Chỉ phụ thuộc vào tỷ lệ của các giá trị đặc trưng hơn là giá trị riêng lẻ của nó. Số lượng  $(r+1)^2/r$  là ở mức tối thiểu khi hai giá trị riêng là bằng nhau và nó tăng theo  $r$ . Vì vậy, để kiểm tra tỷ lệ của độ cong chính là một ngưỡng  $r$  dưới đây chúng ta chỉ cần kiểm tra:

$$\frac{\text{Tr}(H)^2}{\text{Det}(H)} < \frac{(r+1)^2}{r} \quad (2.16)$$

Đây là tính toán rất hiệu quả với chưa đến 20 điểm nổi bật cần phải kiểm tra từng keypoint. Ở đây ta sử dụng một giá trị của  $r = 10$  trong đó loại bỏ keypoint có tỷ lệ giữa đường cong lớn hơn 10. Việc chuyển đổi từ hình 2.10 (c) và (d) cho thấy ảnh hưởng của hoạt động này.

### 2.5.5. Gán hướng

Bằng cách gán một hướng phù hợp với từng keypoint dựa trên các thuộc tính hình ảnh cục bộ, các bộ mô tả keypoint có thể liên quan đến hướng và do đó đạt được sự ổn định khi xoay hình ảnh. Tỉ lệ của các keypoint được sử dụng để chọn hình ảnh Gaussian mịn L với tỉ lệ gần nhất, vì thế tất cả các tính toán được thực hiện một cách bất biến tỉ lệ. Đối với mỗi hình ảnh mẫu L(x, y) ở tỉ lệ này, độ lớn

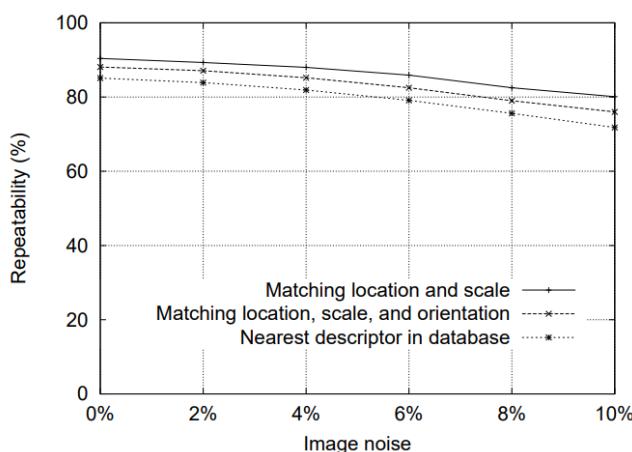
gradient  $m(x, y)$  và hướng  $\theta(x, y)$  được tính toán trước do sự khác biệt điểm ảnh:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2.17)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (2.18)$$

Một biểu đồ hướng được hình thành từ những hướng dốc của điểm lấy mẫu trong khu vực xung quanh các keypoint. Hướng biểu đồ tần số có 36 ngăn (bin) bao phủ 360 độ của hướng. Mỗi mẫu thêm vào biểu đồ được gán trọng số bằng độ lớn Gradient của nó và bởi một hình tròn trọng số Gaussian với  $\sigma$  gấp 1,5 lần so với tỉ lệ của các keypoint.

Hình 2.11 cho thấy sự ổn định vị trí, tỉ lệ, hướng và được gán hướng khác nhau với nhiễu ảnh. Trước những hình ảnh được quay và thu nhỏ lại bởi một lượng ngẫu nhiên, dòng đầu cho thấy sự ổn định của vị trí keypoint và gán tỉ lệ. Dòng thứ hai cho thấy sự ổn định phù hợp khi gán hướng (yêu cầu trong khoảng 15 độ). Khoảng cách giữa hai dòng trên cùng thể hiện việc gán hướng vẫn chính xác 95% ngay cả sau khi bổ sung  $\pm 10\%$  nhiễu ảnh (tương đương với một camera cung cấp ít hơn 3 bit chính xác). Các cách đo biến đổi hướng cho các đối sánh chính xác là khoảng 2,5 độ, tăng lên 3,9 độ cho 10% nhiễu. Điểm mấu chốt trong hình 2.11 cho thấy đối sánh đúng một mô tả chính xác keypoint đến một cơ sở dữ liệu của 40.000. Biểu đồ sau cho thấy các thuộc tính SIFT làm việc tốt ngay cả một lượng lớn các nhiễu pixel và các nguyên nhân chính gây lỗi là vị trí và tỉ lệ phát hiện ban đầu.

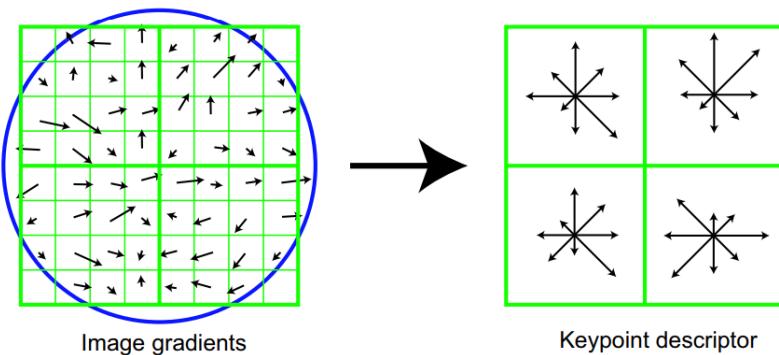


**Hình 2.11:** Đồ thị độ nhiễu của ảnh

### 2.5.6. Bộ mô tả hình ảnh cục bộ

Các phép xử lý trên đã được gán một vị trí ảnh, tỉ lệ và hướng đến mỗi điểm Keypoint. Những thông số ám chỉ sự lặp lại vị trí hệ tọa độ 2D trong đó mô tả các vùng ảnh cục bộ và do đó bất biến các thông số này. Bước tiếp theo là tính toán mô tả cho các khu vực hình ảnh cục bộ mà đặc biệt là chưa bất biến với các biến thể còn lại, chẳng hạn như thay đổi độ sáng hoặc thu – phóng ảnh, xoay.

Một cách tiếp cận là một mẫu cường độ ảnh cục bộ xung quanh keypoint ở tỉ lệ thích hợp, và để đối sánh chúng với các cách sử dụng biện pháp tương quan bình thường. Tuy nhiên, tương quan đơn giản của các bản vá lỗi hình ảnh rất nhạy cảm với những thay đổi, chẳng hạn như Affine hoặc thay đổi hướng nhìn 3D hay biến dạng mềm. Cách tiếp cận tốt hơn đã được chứng minh bởi Edelman, Intrator, và Poggio (1997). Họ đề xuất dựa trên một mô hình thị giác sinh học, đặc biệt là các tế bào thần kinh phức tạp trong vỏ não thị giác chính. Những tế bào thần kinh phức tạp đáp ứng với một gradient ở một hướng cụ thể và tần số không gian, nhưng vị trí của gradient trên võng mạc được phép thay đổi theo một lĩnh vực nhỏ hơn được cục bộ hóa một cách chính xác. Edelman et al. giả thuyết rằng chức năng của các tế bào thần kinh phức tạp này là cho phép đối sánh và nhận dạng của đối tượng 3D từ một vùng của hướng nhìn. Họ đã thực hiện thí nghiệm chi tiết sử dụng mô hình máy tính 3D của hình dạng đối tượng và động vật mà thấy phù hợp với gradients trong khi cho phép thay đổi vị trí của chúng tốt hơn khi xoay 3D. Ví dụ, nhận dạng chính xác cho các đối tượng 3D xoay theo chiều sâu bằng 20 độ tăng từ 35% cho mối tương quan của gradient đến 94% bằng cách sử dụng mô hình tế bào phức tạp. Việc mô tả dưới đây được lấy cảm hứng từ ý tưởng này, nhưng cho phép thay đổi vị trí bằng cách sử dụng một cơ chế tính toán khác nhau.



**Hình 2.12:** Hướng phân bố trên ảnh và bộ mô tả các điểm Keypoint

#### 2.5.6.1. Bộ mô tả

Hình 2.12 minh họa các tính toán của các bộ mô tả keypoint. Đầu tiên là độ lớn gradient và hướng được lấy mẫu xung quanh vị trí keypoint sử dụng tỉ lệ của các keypoint để lựa chọn cấp độ mờ Gaussian cho hình ảnh. Để đạt được hướng bất biến, tọa độ của các mô tả và độ dốc được xoay tương đối với hướng keypoint. Để đạt hiệu quả, gradient được tính toán trước ở tất cả các mức của các kim tự tháp như mô tả trong phần 2.1.5. Những minh họa bằng các mũi tên nhỏ ở mỗi vị trí lấy mẫu bên trái của Hình 2.12.

Bộ mô tả được hình thành từ một vector chứa các giá trị của tất cả các thực thể histogram tương ứng với chiều dài của mũi tên bên phải của Hình 2.12. Hình vẽ cho thấy một mảng  $2 \times 2$  biểu đồ hướng, trong khi các thí nghiệm dưới đây cho thấy rằng kết quả tốt nhất đạt được với một mảng  $4 \times 4$  biểu đồ với 8 hướng trong từng vùng. Do đó, các thí nghiệm này sử dụng một vector đặc trưng  $4 \times 4 \times 8 = 128$  phần tử cho mỗi Keypoint.

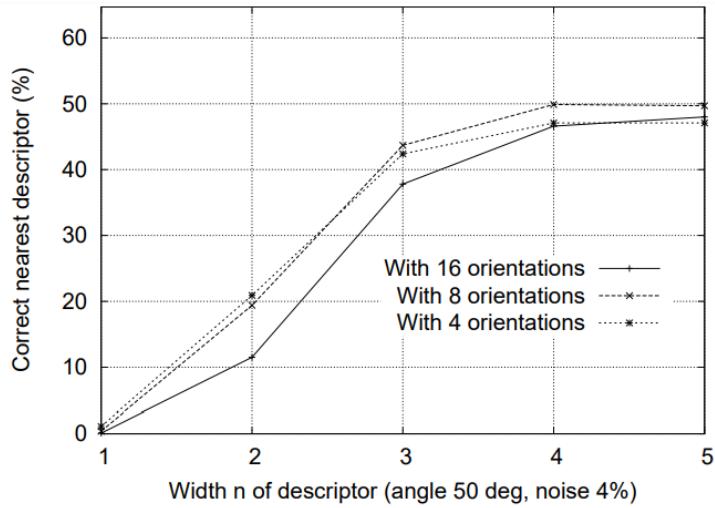
Khi thay đổi độ sáng trong đó một hằng số được thêm vào mỗi điểm ảnh hình ảnh thì sẽ không ảnh hưởng đến giá trị gradient khi chúng được tính từ sự khác biệt pixel. Do đó, các mô tả là bất biến để thay đổi Affine trong chiều sáng. Tuy nhiên, những thay đổi ánh sáng phi tuyến tính cũng có thể xảy ra do độ bão hòa của máy ảnh hoặc do sự thay đổi ánh sáng có ảnh hưởng đến bề mặt 3D với hướng khác nhau. Các hiệu ứng này có thể gây ra một sự thay đổi tương đối lớn cho một gradient, nhưng ít có khả năng ảnh hưởng đến hướng gradient. Do đó, ta sẽ làm giảm ảnh hưởng của độ dốc lớn bởi các giá trị ngưỡng trong các vector đặc trưng cho mỗi đơn vị, ngưỡng này

không được lớn hơn 0.2 và sau đó đưa về giá trị bình thường cho mỗi đơn vị chiều dài. Điều này có nghĩa là sự phù hợp với độ lớn cho gradient không còn là quan trọng và sự phân bố các hướng có trọng tâm hơn. Giá trị của 0.2 được xác định bằng thực nghiệm bằng cách sử dụng các hình ảnh có chứa sự chiếu sáng khác nhau đối với các đối tượng 3D.

#### **2.5.6.2. Kiểm thử bộ mô tả**

Có hai tham số có thể được sử dụng để thay đổi độ phức tạp của mô tả: số lượng hướng  $r$  trong biểu đồ và chiều rộng  $n$  của mảng  $n \times n$  các hướng của biểu đồ. Kích thước của vector mô tả kết quả là  $rn^2$ . Như sự phức tạp của mô tả phát triển, nó có thể phân biệt rõ hơn trong một cơ sở dữ liệu lớn, nhưng nó cũng sẽ nhạy cảm hơn với biến dạng hình và làm bế tắc công việc.

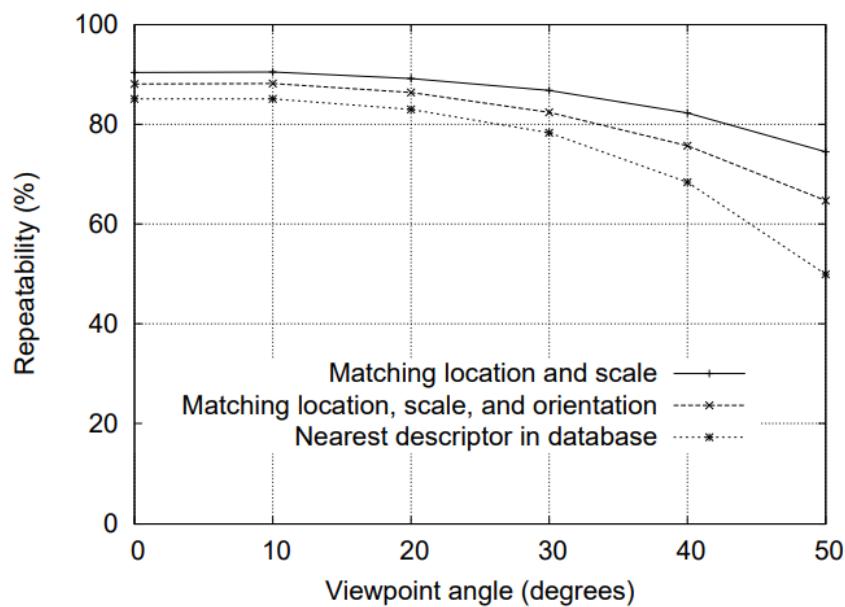
Hình 2.13 cho thấy kết quả thực nghiệm trong đó số các hướng và kích thước của các mô tả đã được thay đổi. Các đồ thị đã được tạo ra cho một chuyển đổi khung nhìn trong đó một mặt phẳng nghiêng 50 độ so với hướng nhìn và 4% nhiễu hình ảnh được thêm vào. Điều này là giới hạn gần của đối sánh đáng tin cậy, vì đây là những trường hợp khó hơn và trong các trường hợp này thì thực hiện mô tả là quan trọng nhất. Kết quả thể hiện số phần trăm keypoint được đối sánh đúng so với láng giềng gần nhất trong cơ sở dữ liệu của 40.000 keypoint. Đồ thị cho thấy một xu hướng biểu đồ duy nhất ( $n = 1$ ) là rất ít tại các điểm khác biệt, nhưng kết quả tiếp tục cải thiện lên đến một mảng  $4 \times 4$  của biểu đồ với 8 hướng. Khi số hướng tăng lên hoặc một mô tả lớn hơn có thể thực sự làm sai lệch việc đối sánh bằng cách làm cho các mô tả nhạy cảm hơn với sự biến dạng. Những kết quả này là tương tự nhau với thay đổi điểm nhìn và nhiễu, mặc dù trong một số trường hợp đơn giản sự khác biệt tiếp tục cải thiện (từ mức cao) với  $5 \times 5$  và kích thước bộ mô tả lớn. Ở đây ta sử dụng một mô tả  $4 \times 4$  với 8 hướng, dẫn đến các vector với 128 chiều. Trong khi số chiều của mô tả có vẻ nhiều và ta đã tìm thấy rằng nó luôn thực hiện tốt hơn so với mô tả dưới chiều trên một loạt các đối sánh phù hợp và các chi phí tính toán của so khớp vẫn thấp khi sử dụng các phương pháp láng giềng gần nhất.



Hình 2.13: Độ rỗng của bộ mô tả (góc 50 độ, độ nhiễu ánh 4%)

#### 2.5.6.3. Độ nhạy với biến đổi Affine

Độ nhạy của các mô tả trong thay đổi Affine được kiểm tra trong Hình 2.14. Biểu đồ thể hiện độ tin cậy của điểm keypoint và lựa chọn tỉ lệ, phân hướng, đối sánh lảng giềng gần nhất với một cơ sở dữ liệu như là một hàm số của phép quay theo chiều sâu so với hướng nhìn. Có thể thấy rằng mỗi giai đoạn tính toán đã làm giảm khả năng lặp lại với việc tăng biến dạng Affine nhưng các so khớp chính xác vẫn ở trên mức 50% với sự thay đổi 50 độ của hướng nhìn.

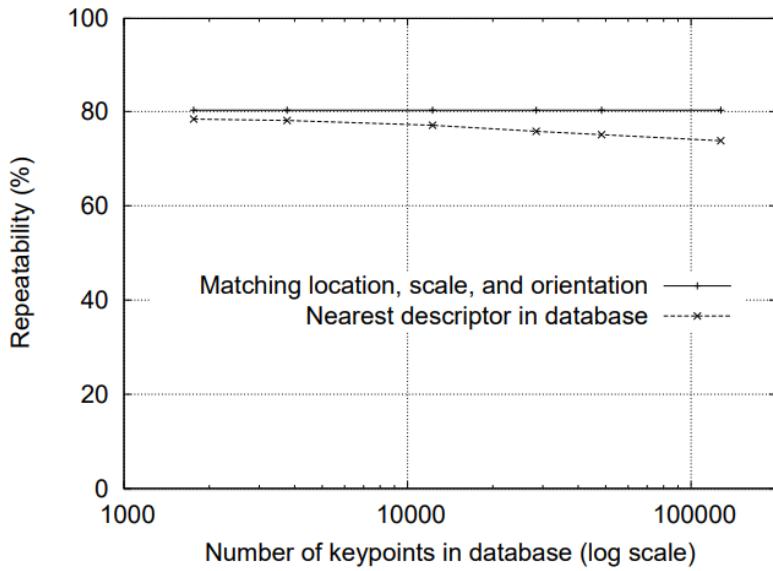


Hình 2.14: Sự ổn định của việc phát hiện vị trí các Keypoint

Để đạt được độ tin cậy khi đối sánh trên một khung nhìn rộng hơn, một trong các máy dò bát biến Affine có thể được dùng để chọn và lấy mẫu các khu vực ảnh. Như đã đề cập ở trên, không cách tiếp cận nào trong số những phương pháp biến đổi Affine bát biến thực sự, tất cả đều bắt đầu từ việc xác định thuộc tính ban đầu khi không bát biến affine. Điều đó thể hiện phương pháp tốt nhất về bát biến Affine. Mikolajczyk (2002) đã đề xuất và chạy thử nghiệm chi tiết với các máy dò Harris-Affine. Ông thấy rằng các keypoint lặp lại dưới dưới một góc nhìn 50 độ và nó vẫn đạt gần 40% dưới góc nhìn 70 độ, nó cung cấp hiệu suất tốt hơn cho những thay đổi Affine lớn. Nhưng nhược điểm là chi phí tính toán cao hơn nhiều, giảm số lượng các keypoint, và tính ổn định kém hơn cho những thay đổi Affine nhỏ do sai sót trong việc gán một khung Affine phù hợp dưới nhiễu. Trong thực tế, phạm vi cho phép quay cho các đối tượng 3D là ít hơn đáng kể hơn so với bề mặt phẳng, vì vậy Affine bát biến thường không phải là yếu tố hạn chế trong khả năng để phù hợp với sự thay đổi quan điểm trên. Nếu một phạm vi rộng của Affine bát biến là mong muốn, chẳng hạn như đối với một bề mặt được biết đến là phẳng, sau đó là một giải pháp đơn giản là áp dụng phương pháp tiếp cận của Pritchard và Heidrich (2003), trong đó thuộc tính SIFT bổ sung được tạo ra từ biến đổi Affine phiên bản 4 của hình ảnh huấn luyện tương ứng với thay đổi 60 độ của hướng nhìn, cho phép việc sử dụng các thuộc tính chuẩn SIFT và không phát sinh thêm chi phí khi các bức ảnh được nhận dạng, nhưng kết quả là tăng kích thước của cơ sở dữ liệu thuộc tính theo hệ số 3.

#### **2.5.6.4. So khớp với cơ sở dữ liệu lớn**

Một vấn đề còn quan trọng để đo sự khác biệt của thuộc tính là độ tin cậy của các biến đổi sánh như là một hàm như thế nào với số lượng các thuộc tính trong cơ sở dữ liệu đối sánh. Với cách sử dụng một cơ sở dữ liệu 32 ảnh với khoảng 40.000 keypoint, hình 2.15 cho thấy độ tin cậy của các đối sánh như một hàm của độ lớn cơ sở dữ liệu. Hình vẽ này đã được tạo ra bằng cách sử dụng một cơ sở dữ liệu lớn hơn 112 ảnh, với hướng nhìn xoay 30 độ và 2% nhiễu ảnh và lấy ảnh xoay ngẫu nhiên và thay đổi tỉ lệ.



**Hình 2.15:** Số lượng Keypoint trong cơ sở dữ liệu

Các đường nét đứt hiển thị một phần của thuộc tính ảnh mà những hàng xóm gần nhất trong cơ sở dữ liệu đối sánh đúng như là một hàm của kích thước cơ sở dữ liệu hiển thị trên một tỉ lệ logarit. Các điểm tận cùng bên trái là phù hợp với các thuộc tính từ một hình ảnh duy nhất, trong khi các điểm ngoài cùng bên phải là lựa chọn phù hợp từ một cơ sở dữ liệu của tất cả các thuộc tính từ 112 hình ảnh. Có thể thấy rằng độ tin cậy của đối sánh giảm như là một hàm của số lượng các sai số, nhưng tất cả các dấu hiệu cho thấy nhiều kết quả đúng sẽ tiếp tục được phát hiện ra khi kích thước cơ sở dữ liệu rất lớn.

Các dòng nét liền là tỷ lệ phần trăm của keypoint được nhận dạng tại vị trí đối sánh đúng và hướng trong hình ảnh chuyển đổi. Mỗi quan tâm của ta là khi khoảng cách giữa hai đường là nhỏ nghĩa là các đối sánh bị sai do việc khởi tạo các thuộc tính ban đầu và gán hướng chứ không phải do sự tính khác biệt về thuộc tính, thậm chí với kích thước cơ sở dữ liệu lớn.

### 2.5.7 Đối sánh đặc trưng SIFT

#### 2.5.7.1 Độ đo tương tự và độ đo khoảng cách

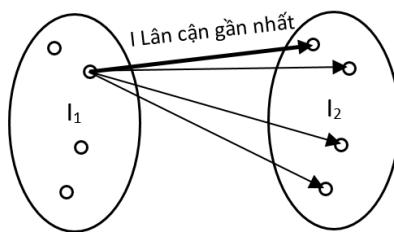
Độ đo tương tự là một trong những phương pháp tốt để máy tính phân biệt được các hình ảnh qua nội dung của chúng. Thông thường hệ thống tra cứu ảnh theo nội dung sẽ truy vấn hình ảnh bằng phương pháp đo tương tự dựa trên các đặc

trung, việc xác định nó có thể dưới nhiều hình thức như phát hiện biên, màu sắc, vị trí điểm ảnh... các phương pháp như histogram, màu sắc và phân tích histogram dòng cột sử dụng biểu đồ để xác định độ tương tự.

Do đó, độ đo có ý nghĩa quan trọng trong tra cứu ảnh dựa theo nội dung. Độ đo mang ý nghĩa quyết định kết quả tìm kiếm sẽ như thế nào, mức độ chính xác ra sao. Nhiều phép đo khoảng cách đã được khai thác trong việc tra cứu ảnh chúng bao gồm: khoảng cách Euclidean, khoảng cách Cosin, khoảng cách giao nhau của biểu đồ histogram, khoảng cách Minkowski... Trong mục này, một vài phép đo khoảng cách sẽ được mô tả và ước lượng. Mục đích của việc ước lượng này để tìm ra một phép đo tương đồng cho các bộ mô tả ước lượng hình dạng khác nhau.

#### 2.5.7.2 Đối sánh đặc trưng cục bộ bất biến

Trước hết để đối sánh các ảnh với nhau thì cần trích xuất tập keypoint tương ứng từ mỗi ảnh bằng các bước đã chỉ ra ở trên. Sau đó việc đối sánh sẽ thực hiện trên các tập keypoint này. Bước chính trong kỹ thuật đối sánh sẽ thực hiện tìm tập con keypoint so khớp nhau ở hai ảnh, để thực hiện việc này sẽ tìm các cặp keypoint trùng nhau lần lượt ở hai ảnh. Tập con các keypoint so khớp chính là vùng ảnh tương đồng. Việc đối sánh hai tập hợp điểm đặc trưng quy về bài toán tìm láng giềng gần nhất của mỗi điểm đặc trưng (hình 2.16).



**Hình 2.16:** Đối sánh 2 ảnh quay về đối sánh 2 điểm đặc trưng

Có 2 vấn đề cần được quan tâm :

Tổ chức tập hợp điểm cho phép tìm kiếm láng giềng một cách hiệu quả

Việc đối sánh phải đạt độ chính xác nhất định

Một phương pháp được đề xuất bởi D. Mount cho phép tìm kiếm nhanh các điểm lân cận được sử dụng[4], ANN là viết tắt của Approximative Nearest Neighbour. Nó cho phép tổ chức dữ liệu dưới dạng *kd-tree*, việc tìm kiếm láng giềng gần nhất

mang tính xấp xỉ trên *kd-tree*. Cụ thể là hai điểm trong không gian đặc trưng được coi là giống nhau nếu khoảng cách Euclidean giữa hai điểm là nhỏ nhất và tỉ số giữa khoảng cách gần nhất với khoảng cách gần nhì phải nhỏ hơn 1 ngưỡng cho trước

Giả sử cặp keypoint có bộ mô tả lần lượt là:

$$A = (a_1, a_2, a_3, \dots, a_{128}) \text{ và } B = (b_1, b_2, b_3, \dots, b_{128})$$

Thì khoảng cách Euclid giữa A và B được tính bằng công thức:

$$D(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

### 2.5.7.3 Độ đo tương đồng cho đặc trưng cục bộ bất biến

Một số độ đo tương đồng cho ảnh sử dụng đặc trưng SIFT

- Độ đo Cosin:

$$d(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \quad (2.19)$$

- Khoảng cách góc:

$$d(x, y) = \cos^{-1}(x \cdot y) \quad (2.20)$$

- Độ đo Euclide:

$$d(x, y) = \sqrt{\sum_{i=1}^n |x_i - y_i|^2} \quad (2.21)$$

- Độ đo Jensen-Shannon divergence :

$$d_{JSD}(H, H') = \sum_{m=1}^M H_m \log \frac{2H_m}{H_m + H'_m} + H'_m \log \frac{2H'_m}{H_m + H'_m} \quad (2.22)$$

Với H, H' là 2 biểu đồ biểu diễn các vector đặc trưng SIFT

### 2.5.8. Ứng dụng cho nhận dạng đối tượng

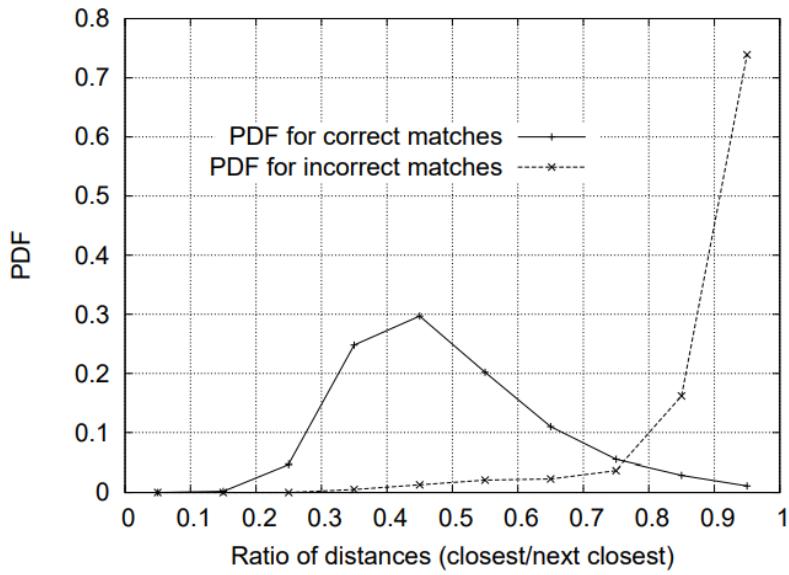
Nhận dạng đối tượng được thực hiện trước tiên bởi việc đối sánh từng keypoint độc lập với cơ sở dữ liệu của keypoint chiết xuất từ các hình ảnh huấn luyện. Nhiều đối sánh trong số những đối sánh đầu tiên sẽ là không chính xác, do thuộc tính không rõ ràng hoặc các thuộc tính phát sinh từ một nền lộn xộn. Do đó, các cụm ít nhất 3 thuộc tính đầu tiên được nhận dạng đúng về một đối tượng và từ

thể của nó, việc đối sánh theo những cụm thuộc tính có xác suất cao hơn nhiều so với các đối sánh đặc điểm riêng biệt. Sau đó, mỗi cụm được kiểm tra bằng cách thực hiện một mô hình hình học chi tiết và kết quả được sử dụng để xem xét xem đối sánh trên đúng hay sai.

#### **2.5.8.1. So khớp Keypoint**

Đối sánh các keypoint tốt nhất được tìm thấy bằng cách xác định điểm lân cận gần nhất với nó trong cơ sở dữ liệu của keypoint từ hình ảnh huấn luyện. Điểm lân cận gần nhất được định nghĩa là các keypoint với khoảng cách Euclidean tối thiểu đối với các vector mô tả bất biến như đã được mô tả trong phần sau.

Tuy nhiên, nhiều thuộc tính từ một hình ảnh sẽ không có bất kỳ đối sánh nào chính xác trong cơ sở dữ liệu chi phí đào tạo bởi vì nó phát sinh từ nền lõi hoặc không được phát hiện trong những hình ảnh huấn luyện. Đó là một cách hữu ích để loại bỏ dễ dàng các thuộc tính mà không có bất kỳ đối sánh tốt nào với cơ sở dữ liệu. Một nhược điểm cục bộ về khoảng cách đến các thuộc tính gần nhất là không hiệu quả vì có nhiều bộ mô tả khác nhau về một đối tượng. Biện pháp hiệu quả hơn thu được bằng cách so sánh khoảng cách của những điểm lân cận gần nhất đó với điểm lân cận gần nhất thứ hai. Nếu có nhiều hình ảnh huấn luyện của cùng một đối tượng, ta sẽ định nghĩa điểm lân cận thứ hai từ lân cận gần nhất được biết đến từ một đối tượng khác so với đối tượng đầu, chẳng hạn như bằng cách chỉ sử dụng các hình ảnh có chứa nhiều đối tượng khác nhau. Biện pháp này hoạt động tốt vì các đối sánh chính xác cần phải có số lượng đáng kể những điểm lân cận gần nhất hơn so với đối sánh không chính xác để đạt được đối sánh đáng tin cậy. Đối với đối sánh sai, có thể sẽ có một số lượng đối sánh sai khác trong khoảng cách tương tự do chiều cao của không gian đặc trưng.



**Hình 2.17:** Tỷ lệ khoảng cách từ điểm lân cận tới điểm kế tiếp

Hình 2.17 cho thấy giá trị của biện pháp này đối với dữ liệu hình ảnh thực tế. Hàm mật độ xác suất cho các đôi sánh chính xác và không chính xác được thể hiện trong trực tuyến gần nhất với điểm láng giềng gần nhất thứ hai của mỗi keypoint. Đôi sánh lân cận gần nhất là một kết hợp chính xác có một PDF (probability of distance from) mà tập trung tại một tỷ lệ thấp hơn nhiều so với các đôi sánh không chính xác. Để thực hiện nhận dạng đối tượng, ta lược bỏ tất cả các đôi sánh trong đó tỷ lệ khoảng cách lớn hơn 0,8, trong đó loại bỏ 90% trong những đôi sánh sai và loại bỏ ít hơn 5% trong những đôi sánh chính xác. Hình vẽ này được tạo ra bằng cách kết hợp các hình ảnh với tỉ lệ ngẫu nhiên và thay đổi hướng, xoay chiều sâu 30 độ và thêm 2% nhiễu hình ảnh đối với một cơ sở dữ liệu của 40.000 Keypoint.

#### 2.5.8.2. Hiệu quả của việc đánh số các điểm lân cận gần

Không có thuật toán nổi tiếng nào có thể xác định chính xác những điểm lân cận gần nhất của các điểm trong không gian mà hiệu hơn so với tìm kiếm vét cạn. Mô tả keypoint ta sử dụng một vector đặc trưng 128 chiều và các thuật toán tốt nhất chẳng hạn như cây kd (Friedman et al., 1977) sẽ nhanh hơn so với tìm kiếm vét cạn trong không gian khoảng 10 chiều (hoặc hơn). Do đó, ta sử dụng một thuật toán gần đúng, gọi là thuật toán Best-Bin-First (BBF) (Beis và Lowe, 1997). Thuật toán

trả về điểm lân cận gần nhất với xác suất cao.

Các thuật toán BBF sử dụng thứ tự tìm kiếm đã được chỉnh sửa cho thuật toán cây kd vì thế các vùng không gian đặc trưng được tìm trong các trật tự khoảng cách gần nhất của nó từ vị trí truy vấn, tìm kiếm ưu tiên này yêu cầu sử dụng đầu tiên được kiểm tra bởi Arya và Mount(1993), họ cung cấp nghiên cứu sâu về việc tính toán các thuộc tính (Arya et al., 1998). Việc tìm kiếm theo trật tự đòi hỏi việc sử dụng một hàng đợi ưu tiên dựa trên heap để xác định về hiệu quả của lệnh tìm kiếm. Một câu trả lời gần đúng có thể thực hiện với chi phí thấp bằng cách cắt đứt tìm kiếm sâu hơn nữa sau khi một số khu vực gần đó đã được tìm rồi. Trong việc thực hiện này, ta cắt đứt tìm kiếm sau khi kiểm tra lần đầu với 200 điểm láng giềng gần. Đối với một cơ sở dữ liệu của 100.000 keypoint, ta sẽ tăng tốc thuật toán tìm kiếm láng giềng gần nhất bằng cách tăng độ lớn gấp đôi và kết quả cho thấy sai số không quá 5% các đối sánh đúng.

#### **2.5.8.3. Cụm biến đổi Hough**

Để tối đa hóa hiệu suất của nhận dạng đối tượng cho các đối tượng nhỏ hoặc khả năng bέ tacute; cao, ta xác định các đối tượng với số lượng ít nhất có thể các đối sánh thuộc tính. Ta đã biết rằng việc nhận dạng là đáng tin cậy khi có 3 thuộc tính. Một hình ảnh chuẩn chứa 2.000 hoặc nhiều thuộc tính có thể đến từ nhiều đối tượng khác nhau và có sự lộn xộn nền. Trong khi kiểm tra tỷ lệ khoảng cách được mô tả trong Phần 2.1.7.1 đã cho phép chúng ta loại bỏ nhiều đối sánh sai phát sinh từ một nền lộn xộn, điều này không loại bỏ các đối sánh từ các đối tượng có giá trị khác, và chúng ta thường vẫn cần phải xác định tập con các đối sánh đúng có chứa ít hơn 1% inliers trong số 99%. Nhiều phương pháp nổi tiếng như RANSAC hoặc phương pháp tính trung bình nhỏ nhất của Squares hoạt động kém khi số phần trăm inliers rơi xuống thấp hơn 50%. May mắn thay, có thể thu được hiệu năng tốt hơn bằng cách phân nhóm các thuộc tính trong không gian bằng cách sử dụng biến đổi Hough (Hough, 1962; Ballard, 1981; Grimson 1990).

Mỗi keypoint ta sẽ đặc tả bằng 4 thông số: vị trí 2D, tỉ lệ, hướng và đối sánh mỗi keypoint trong cơ sở dữ liệu có một đặc tả về các thông số của keypoint liên

quan tới hình ảnh huấn luyện đã được tìm thấy. Do đó, chúng ta có thể tạo ra một biến đổi Hough để dự đoán vị trí, hướng, và tỉ lệ từ giả thuyết đối sánh. Dự đoán này có thể bị sai sót nhiều do sự biến đổi tương đối bởi 4 thông số xấp xỉ 6 độ trong không gian tự do cho mỗi đối tượng 3D và cũng không lý giải cho bất kỳ sự biến dạng nào. Do đó, ta sử dụng kích cỡ mỗi vùng rộng 30 độ để gán hướng, hệ số 2 cho tỉ lệ, và tối đa gấp 0,25 lần kích thước ảnh huấn luyện (bằng cách sử dụng tỉ lệ dự đoán) cho vị trí. Để tránh những vấn đề về ranh giới phân chia vùng, mỗi đối sánh keypoint dùng cho 2 vùng gần nhất ở mỗi hướng, tổng cộng 16 mục cho mỗi giả thuyết và tiếp tục mở rộng phạm vi tư thế.

#### 2.5.8.4. Giải pháp cho các thông số Affine

Các biến đổi Hough được sử dụng để xác định tất cả các cụm có ít nhất 3 mục trong một bin. Mỗi cụm như vậy sau đó tùy thuộc vào một thủ tục xác định hình học trong đó một giải pháp bình phương nhỏ nhất được thực hiện đối với các thông số Affine tốt nhất liên quan đến hình ảnh huấn luyện cho hình ảnh mới. Một biến đổi Affine chính xác cho vòng quay 3D của một bề mặt phẳng dưới phép chiếu trực giao, nhưng sự thiếu chính xác có thể xảy ra khi quay 3D của đối tượng không phẳng. Tuy nhiên, một giải pháp ma trận cơ bản đòi hỏi ít nhất 7 điểm phù hợp so với chỉ cần 3 cho các giải pháp Affine và trong thực tế đòi hỏi nhiều hơn các đối sánh mới ổn định tốt. Ta muốn thực hiện nhận dạng với ít nhất là 3 đối sánh thuộc tính, vì vậy giải pháp Affine cung cấp một điểm khởi đầu tốt hơn và ta có thể khoanh vùng cho các lỗi trong xấp xỉ Affine bằng cách cho phép các lỗi còn sót lớn. Đối với các ví dụ điển hình của các đối tượng 3D, một giải pháp Affine hoạt động tốt vì ta cho phép các lỗi còn sót lại lên đến 0,25 lần so với dự kiến. Một biến đổi Affine của một điểm mô hình  $[xy]^T$  đến một điểm ảnh  $[uv]^T$  có thể được viết như.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (2.23)$$

Khi mô hình biến đổi là  $[t_x \ t_y]^T$  và affine luân chuyển, tỉ lệ và độ căng được

thể hiện bởi các thông số  $m_i$ . Muốn giải quyết cho các thông số chuyển đổi, phương trình trên có thể được viết lại để chèn các ẩn số vào một vector cột:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ \dots & & \dots & & & \end{bmatrix} \begin{bmatrix} m \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad (2.24)$$

Phương trình này cho thấy một đôi sánh duy nhất, tuy nhiên có nhiều đôi sánh hơn nữa có thể được thêm vào, với mỗi đôi sánh có ít nhất 2 hàng cho các ma trận đầu tiên và cuối cùng. Ít nhất 3 đôi sánh là cần thiết cho một giải pháp.

Ta có thể viết hệ thống tuyến tính này như.

$$Ax = b \quad (2.25)$$



**Hình 2.18:** Ví dụ minh họa về thuật toán SIFT

Các giải pháp bình phương nhỏ nhất cho các tham số  $x$  có thể được xác định bằng cách giải quyết các phương trình tương đương:

$$x = [A^T A]^{-1} A^T b \quad (2.26)$$

Công thức trên làm tối thiểu tổng bình phương của khoảng cách từ vị trí mô hình dự đoán đến các địa điểm tương ứng trong hình ảnh. Trong cách tiếp cận này

bình phương nhỏ nhất có thể dễ dàng được mở rộng để giải quyết cho kiểu 3D (Lowe, 1991).

Các giá trị ngoại lai bây giờ có thể được loại bỏ bằng cách kiểm tra mối liên hệ giữa các thuộc tính hình ảnh và mô hình. Để thêm phần chính xác thì áp dụng giải pháp bình phương nhỏ nhất, bây giờ ta yêu cầu mỗi đối sánh phải ràng buộc trong vòng nửa phạm vi lỗi đã được sử dụng cho các thông số trong biến đổi Hough. Nếu ít hơn 3 điểm còn lại sau khi loại bỏ giá trị ngoại lai, sau đó đối sánh bị loại bỏ. Sau đó các giải pháp bình phương nhỏ nhất được giải quyết với các điểm còn lại, và quá trình lặp đi lặp lại.

Phương pháp này lần đầu tiên tính toán số lượng dự kiến của các đối sánh sai với mô hình đặt ra. Ta chấp nhận một mô hình nếu xác suất cuối cùng cho một luận điểm đúng là lớn hơn 0,98. Đối với các đối tượng mà ở các khu vực nhỏ của hình ảnh, 3 thuộc tính có thể là đủ để nhận dạng chính xác. Đối với các đối tượng lớn bao trùm một hình ảnh rất nhiều kết cấu, số lượng dự kiến của các đối sánh giả là cao hơn, và có thể cần đến đối sánh 10 thuộc tính.

#### **2.5.9. Ví dụ nhận dạng**

Ví dụ 1: Hình dưới đây cho thấy một ví dụ về sự nhận dạng đối tượng cho một hình ảnh lộn xộn và làm bế tắc khi chứa các đối tượng 3D. Các hình ảnh huấn luyện của một chiếc xe lửa đồ chơi và một con ếch được hiển thị bên trái.

Những hình ảnh ở giữa (kích thước 600x480 pixels) có chứa các phần của các đối tượng ẩn đằng sau và với nền lộn xộn rộng thì để phát hiện các đối tượng có thể không nhanh cả đối với tầm nhìn của con người. Các keypoint đã được sử dụng để nhận dạng nằm trong vùng hình vuông với một dòng chú thích để chỉ hướng.

Một ứng dụng tiềm năng của phương pháp này là đặt nhận dạng, trong đó một thiết bị di động hay một chiếc xe có thể xác định vị trí của mình bằng cách nhận dạng địa điểm quen thuộc. Hình 2.19 đưa ra một ví dụ về ứng dụng này, trong đó hình ảnh huấn luyện được thực hiện của một số điểm. Phần bên trái là bức tường bằng gỗ hoặc cây với thùng rác. Những hình ảnh thử nghiệm (kích thước 640x315 pixels) phía trên bên phải được lấy từ một khung hình xoay khoảng 30 độ xung

quanh hiện trường từ vị trí ban đầu, nhưng những hình ảnh địa điểm đào tạo dễ dàng được nhận biết.

Tất cả các bước của quá trình nhận thức có thể được thực hiện một cách hiệu quả, vì vậy tổng thời gian để nhận ra tất cả các đối tượng trong hình 12 hoặc 13 là ít



**Hình 2.19:** Ví dụ về sự nhận dạng đối tượng

một máy tính xách tay với máy quay video kèm theo, và thử nghiệm chúng rộng rãi trên một loạt các điều kiện. Nói chung, các bề mặt phẳng kết cấu có thể được xác định đáng tin cậy hơn một vòng quay ở độ sâu lên đến 50 độ trong bất kỳ hướng nào miễn là cung cấp đủ ánh sáng và không tạo ra ánh sáng chói quá mức. Đối với

các đối tượng 3D, phạm vi của vòng xoay trong chiều sâu để nhận dạng tốt chỉ trong khoảng 30 độ trong bất kỳ hướng nào và thay đổi ánh sáng gây rối hơn. Đối với những lý do này, nhận dạng đối tượng 3D được thực hiện tốt nhất bằng cách tích hợp các thuộc tính từ nhiều khung nhìn, chẳng hạn như với các vị trí cục bộ ta gom thuộc tính thành cụm(Lowe, 2001).

Những keypoint này cũng đã được áp dụng cho các vấn đề về định vị hóa robot và truy cập bản đồ (Se, Lowe và Little, 2001). Trong ứng dụng này, một hệ thống âm thanh nổi Trinocular được sử dụng để xác định ước tính 3D cho các địa điểm keypoint. Keypoint chỉ được sử dụng khi chúng xuất hiện trong tất cả 3 hình ảnh với sự chênh lệch phù hợp, dẫn đến rất ít giá trị ngoại lai. Khi di chuyển robot, nó định vị bản thân bằng cách sử dụng thuộc tính phù hợp với bản đồ 3D hiện tại, và sau đó từng bước bổ sung thêm các thuộc tính bản đồ trong khi cập nhật vị trí 3D của nó bằng cách sử dụng một bộ lọc Kalman.

## 2.6 Thuật toán SURF

### 2.6.1 Giới thiệu

Phương pháp SIFT đã giải quyết được những hạn chế còn tồn tại ở thuật toán tìm kiếm góc Harris và trở thành một trong những thuật toán trích chọn đặc trưng mạnh mẽ nhất. Dù vậy, tốc độ xử lý của SIFT vẫn còn rất chậm và không phù hợp với các ứng dụng thời gian thực

Để giải quyết bài toán này, người ta đã giới thiệu thuật toán trích chọn đặc trưng SURF (Speed Up Robust Features) có được sự cân bằng giữa yêu cầu tốc độ và sự chính xác. Đặc trưng tối ưu cả hai giai đoạn phát hiện đặc trưng và mô tả đặc trưng về mặt thời gian tính toán nhưng vẫn giữ được tính bền vững của đặc trưng. Bộ phát hiện đặc trưng của SURF sử dụng phép xấp xỉ trên ma trận Hessian và ảnh tích hợp (Integral Image) để làm giảm thời gian tính toán một cách đáng kể. Bộ mô tả đặc trưng tương tự như đặc trưng SIFT, sử dụng vector 64 chiều chứa thông tin biến thiên trên ảnh dựa trên sự phân phối bậc nhất Haar wavelet tác động trên trục x và y, kết hợp với ảnh tích lũy làm tăng tốc độ tính toán. SURF được mô tả bởi vector có số chiều ít hơn SIFT nên tốc độ so khớp nhanh hơn, tuy

nhiên độ bền vững vẫn được đảm bảo. Hơn thế nữa, bằng việc đánh chỉ mục dựa trên dấu của Laplacian, đặc trưng SURF không chỉ giữ tính bền vững cho đặc trưng mà còn làm tăng tốc độ so khớp (tăng gấp 2 trong trường hợp tốt nhất).

### **2.6.2 Bộ mô tả SURF**

Hiệu quả của SIFT tốt hơn nhiều so với các mô tả khác. Sự pha trộn của các thông tin sơ sài cục bộ hóa và sự phân bố của gradient liên quan đến các thuộc tính đường như mang lại sức mạnh đặc biệt tốt tránh ảnh hưởng của lỗi cục bộ hóa về tỉ lệ hoặc không gian. Việc sử dụng thể mạnh và hướng của gradient làm giảm ảnh hưởng của thay đổi trắc quang.

Bộ mô tả SURF được đề xuất dựa trên tính chất tương tự với độ phức tạp giảm dần

Thuật toán của kỹ thuật SURF gồm những bước dưới đây:

- Sử dụng bộ dò Fast-Hessian để xác định các điểm nổi bật
- Gán hướng cho các điểm nổi bật và mô tả đặc trưng SURF
- So khớp đặc trưng.

#### **2.6.2.1 Bộ dò Fast-Hessian**

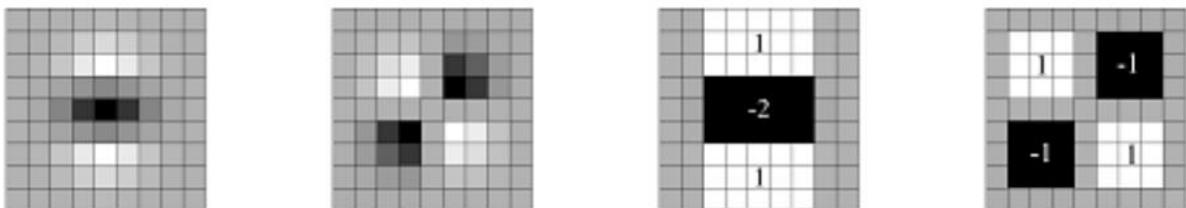
Bộ dò này căn cứ trên ma trận Hessian vì nó có hiệu suất tốt trong tính toán thời gian và độ chính xác. Tuy nhiên thay vì sử dụng một phát hiện Hessian cho việc lựa chọn vị trí và tỉ lệ (như đã được thực hiện trong các máy dò Laplace - Hessian), ở đây ta dựa vào các yếu tố quyết định của Hessian. Cho một điểm  $x=(x,y)$  trong một hình I, các ma trận Hessian  $H(x,\sigma)$  tại  $x$  ở tỉ lệ  $\sigma$  là được định nghĩa như sau.

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{yx}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (2.27)$$

Với  $L_{xx}(x, \sigma) = \frac{\partial^2}{\partial x^2} g(\sigma)$  là tích của đạo hàm bậc hai của hàm Guassian với ảnh I tại điểm  $x(x,y)$ , có tỉ lệ  $\sigma$ .

Nếu như SIFT xấp xỉ việc tính Laplacian của hàm Gaussian (LoG) bằng việc tính sai khác của hàm Gaussian (DoG) thì SURF xấp xỉ việc tính đạo hàm cấp 2 của

hàm Gaussian bằng các hộp lọc (box filters). Dưới đây là một ví dụ của việc tính xấp xỉ đạo hàm cấp hai của hàm Gaussian với hệ số tỉ lệ thấp nhất bằng hộp lọc:



**Hình 2.20:** Xấp xỉ đạo hàm cấp 2 hàm Gaussian bằng hộp lọc

Trong hình trên: Ảnh thứ nhất là đạo hàm ma trận đạo hàm cấp 2 Gaussian theo trục y, ảnh thứ hai theo trục x và trục y. Ảnh thứ ba và thứ tư lần lượt là các hộp lọc xấp xỉ với hai trường hợp của ảnh một và hai. Phép tích chập của ảnh với các hộp lọc này được thực hiện rất nhanh bằng việc sử dụng kết hợp với ảnh tích lũy. Ta xác định vị trí và hệ số tỉ lệ tương ứng của điểm đặc trưng dựa trên định thức của ma trận Hessian. Công thức tính xấp xỉ định thức ma trận Hessian

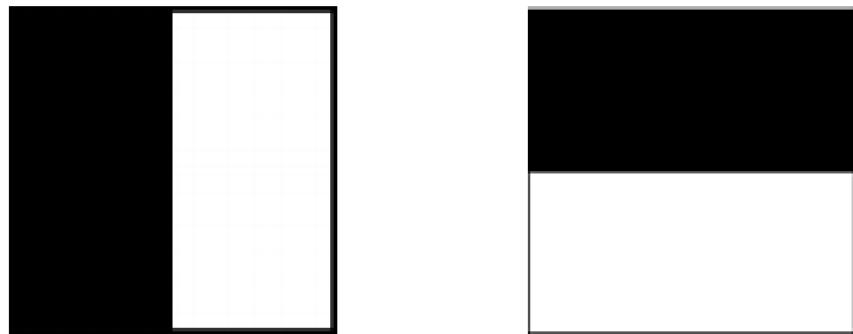
$$\det(H_{x \text{ và } y}) = D_{xx}D_{yy} - (w \cdot D_{xy})^2 \quad (2.28)$$

Trong đó w là trọng số cân bằng của biểu thức định thức ma trận Hessian tùy thuộc vào hệ số tỉ lệ.  $D_{xx}, D_{yy}, D_{xy}$  là các hộp lọc xấp xỉ Gaussian

Vị trí, tỉ lệ và không gian ảnh mà điểm đặc trưng được xác định một phép loại trừ phi cực đại trong một vùng  $3 \times 3 \times 3$

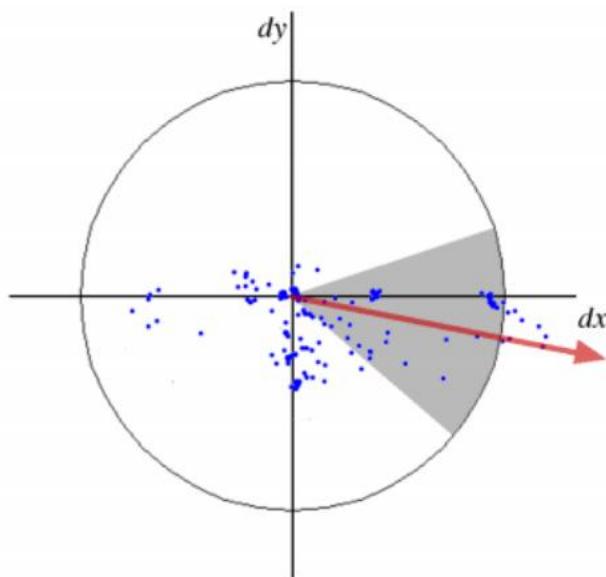
### 2.6.2.2 Gán hướng cho điểm nổi bật và mô tả đặc trưng SURF

Đầu tiên, ta phải xác định vùng hình xung quanh điểm đặc trưng vừa tìm được, gán một giá trị hướng duy nhất cho điểm đặc trưng. Kích thước của hình tròn phụ thuộc và hệ số tỉ lệ tương ứng trong không gian ảnh mà điểm đặc trưng tìm được. Ở đây các tác giả chọn bán kính của hình tròn là  $6s$ , trong đó  $s$  là tỉ lệ mà tại đó điểm đặc trưng được tìm thấy. Hướng của đặc trưng được tính bằng Haar wavelet tác động theo hai hướng x và y. Trong đó, vùng tối có trọng số  $-1$ , vùng sáng có trọng số  $+1$ . Kích thước của wavelet cũng phụ thuộc vào hệ số tỉ lệ  $s$ .



**Hình 2.21:** Lọc Haar wavelet để tính sự ảnh hưởng trên hai hướng x và y

Haar wavelet có thể được tính một cách nhanh chóng bằng cách sử dụng ảnh tích lũy tương tự như hộp lọc xấp xỉ của đạo hàm cấp 2 hàm Gaussian. Vector hướng nào trội nhất sẽ được ước lượng và gắn vào thông tin của điểm đặc trưng. Hình dưới đây sẽ mô tả hướng và vùng ảnh hưởng của đặc trưng.



**Hình 2.22:** Vùng hình tròn xung quanh và hướng đại diện cho điểm đặc trưng

Tiếp theo, ta xây dựng các vùng hình vuông xung quanh điểm đặc trưng men theo vector hướng vừa ước lượng được ở bước trước đó. Vùng hình vuông này được chia nhỏ thành  $4 \times 4$  hình vuông con để ghi nhận thông tin của trên miền không gian ảnh lân cận. Haar wavelet được rút trích trên toàn bộ không gian điểm ảnh. Wavelet

tác động trên hai hướng ngang và dọc được cộng dồn các giá trị  $d_x$  và  $d_y$  trên mỗi hình vuông con.

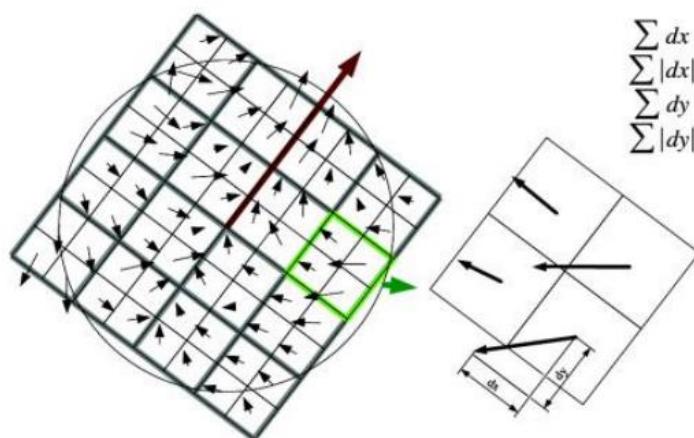
Hơn thế nữa, các giá trị tuyệt đối  $|d_x|$  và  $|d_y|$  cũng được cộng dồn để lấy thông tin về độ lớn của sự thay đổi cường độ sáng trên ảnh.

Như vậy mỗi hình vuông con sẽ được mô tả bởi một vector 4 chiều:

$$V = [\Sigma dx, \Sigma dy, \Sigma |dx|, \Sigma |dy|] \quad (2.29)$$

Như vậy vector mô tả cho tất cả  $4 \times 4$  hình vuông con là một vector 64 chiều ( $4 \times 4 \times 4$ ). Đây cũng chính là mô tả đặc trưng chuẩn của SURF (hay còn gọi là SURF-64). Ngoài ra còn có các phiên bản khác dựa trên cách chia hình vuông con như SURF – 36, SURF – 128...

Tuy nhiên thực nghiệm của các tác giả cho thấy rằng SURF – 64 cho tốc độ tính toán tốt nhất mà vẫn đảm bảo tính bền vững của đặc trưng. Haar wavelet bất biến với sự thay đổi của ánh sáng và sự tương phản khi ta chuẩn hóa vector mô tả đặc trưng về chiều dài đơn vị.

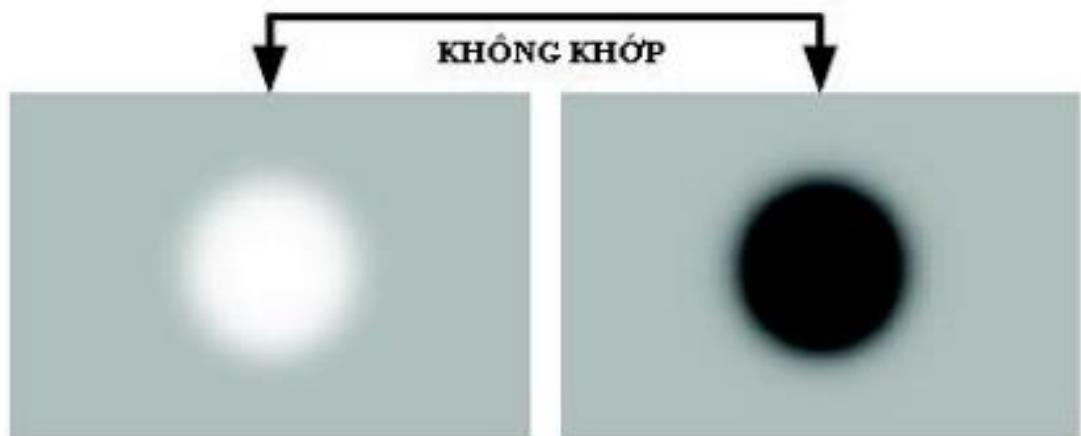


**Hình 2.23:**  $4 \times 4$  hình vuông con xung quanh điểm đặc trưng

#### 2.6.2.3 So khớp đặc trưng

Đặc điểm quan trọng của đặc trưng SURF là quá trình rút trích đặc trưng nhanh do sử dụng kỹ thuật ảnh tích lũy và phép loại trừ phi cực đại. Quá trình so khớp đặc trưng cũng nhanh hơn rất nhiều chỉ bằng một bước đánh chỉ mục đơn giản dựa trên dấu của Laplacian (trace của ma trận Hessian). Ta không phải tốn chi phí

tính toán trong bước này do trong quá trình phát hiện đặc trưng đã được tính sẵn. Dấu của Laplacian giúp phân biệt đóm sáng trên nền tối và đóm tối trên nền sáng. Điểm đặc trưng sáng chỉ có thể khớp với điểm đặc trưng sáng khác, tương tự cho đặc trưng tối. Kỹ thuật này có thể giúp cho quá trình so khớp nhanh gấp đôi trong trường hợp tốt nhất do không phải tốn chi phí tính toán dấu của Laplacian



**Hình 2.24:** So khớp đặc trưng

Nếu độ tương phản giữa hai điểm quan tâm khác nhau (tối trên nền sáng với sáng trên nền tối), ứng viên sẽ không được xem là so khớp có giá trị.

## 2.7 Thuật toán RANSAC

### 2.7.1 Giới thiệu

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. Là một phương pháp lặp đi lặp lại để ước lượng các tham số của mô hình toán học từ một tập hợp các dữ liệu quan sát có chứa các giá trị outliers, khi các outlier được cho là không ảnh hưởng đến các giá trị ước tính. Do đó, nó cũng có thể được hiểu như là một phương pháp phát hiện outliers.

RANSAC giả định rằng dữ liệu huấn luyện bao gồm các giá trị “inliers”, tức là dữ liệu mà sự phân bố có thể được giải thích bằng một số tham số mô hình, mặc dù có thể bị nhiễu và “outliers” là dữ liệu không phù hợp với mô hình. Vì vậy, sử dụng các outliers khi đào tạo mô hình sẽ làm tăng lỗi dự đoán cuối cùng vì chúng hầu như không có thông tin về mô hình. Hơn nữa, RANSAC chỉ tập trung vào mô hình

tham số chỉ với các giá trị inliers mà không quan tâm đến các giá trị outliers. Tuy nhiên, phân tách các dữ liệu như inliers và outliers sẽ là một giả định mạnh, nhưng nó là sự khác biệt chính của RANSAC từ các phương pháp khác. Ngoài ra, ngay cả khi giả định này không chứa cho một bộ dữ liệu, RANSAC sẽ không làm tổn hại đến việc ước lượng thông số, như trong điều kiện này nó sẽ xem xét toàn bộ tập dữ liệu như là các giá trị inliers và đào tạo mô hình với chúng.

Để loại bỏ các giá trị outliers trong quá trình huấn luyện, RANSAC sử dụng một bộ mẫu nhỏ để huấn luyện một mô hình hơn là sử dụng tất cả các dữ liệu và sau đó mở rộng bộ với các mẫu thích hợp khác. Bằng cách sử dụng một bộ nhỏ, nó tự động giả định rằng mô hình có thể được ước tính với một số lượng nhỏ các giá trị inliers. Tuy nhiên, nó là một giả định mềm và nó tổ chức cho hầu hết các trường hợp. Ví dụ để nắm bắt một chức năng tuyến tính, cần hai mẫu dữ liệu là đủ.

### **2.7.2 Phương pháp**

RANSAC thông nhất lựa chọn ngẫu nhiên một tập hợp các mẫu dữ liệu và sử dụng nó để ước lượng các tham số mô hình. Sau đó, nó xác định các mẫu nằm trong khả năng chịu lỗi của mô hình được tạo ra. Các mẫu này được coi là đã đồng ý với mô hình được tạo ra và được gọi là bộ đồng thuận của các mẫu dữ liệu đã chọn. Ở đây, các mẫu dữ liệu trong sự đồng thuận coi như inliers và phần còn lại như là outlier của RANSAC. Nếu tính các mẫu trong sự đồng thuận là đủ cao, nó sẽ tập hợp các mô hình cuối cùng của sự đồng thuận với việc sử dụng chúng. Nó lặp lại quá trình này cho một số lần lặp lại và trả về mô hình có sai số trung bình nhỏ nhất trong số các mô hình tạo ra.

Là một thuật toán ngẫu nhiên, RANSAC không đảm bảo để tìm ra mô hình tham số tối ưu đối với các giá trị inliers. Tuy nhiên, xác suất để đạt được giải pháp tối ưu có thể được giữ trên một giới hạn thấp hơn với việc gán các giá trị thích hợp cho các tham số thuật toán.

### **2.7.3 Thuật toán**

Trên thực tế, RANSAC không phải là một thuật toán hoàn chỉnh và tự làm việc. Trong thực tế, nó là một thuật toán bổ sung sử dụng một mô hình và chức

năng khoảng cách để ước lượng các tham số mô hình một cách mạnh mẽ trong khi có các giá trị outliers tồn tại. Nó chỉ đơn giản áp dụng một sự tách outliers đối với bộ dữ liệu và chỉ tập trung vào mô hình bằng các phương pháp tối ưu và chức năng khoảng cách của mô hình. Do đó, trước khi sử dụng RANSAC, một mô hình, một chức năng khoảng cách và một thuật toán tối ưu hóa đã được xác định.

Thuật toán được mô tả tổng quan như sau:

Cho:

data – một tập hợp các điểm dữ liệu quan sát được

model – mô hình có thể được trang bị cho các điểm dữ liệu

n – số lượng tối thiểu các giá trị dữ liệu cần thiết để phù hợp với mô hình

k – số lần lặp lại tối đa cho phép trong thuật toán

t – giá trị ngưỡng để xác định khi một điểm dữ liệu phù hợp với mô hình

d – số lượng các giá trị dữ liệu gần nhau cần thiết để khẳng định rằng một mô hình phù hợp với dữ liệu

Giá trị trả về:

bestfit – các tham số mô hình phù hợp nhất với dữ liệu (hoặc null nếu không có mô hình tốt được tìm thấy)

iterations = 0

bestfit = null

besterr = something really large

while iterations < k

{

maybeinliers = n giá trị được chọn ngẫu nhiên từ dữ liệu

maybemodel = model các tham số được gắn với maybeinliers

alsoinliers = null

Cho mỗi điểm trong dữ liệu không phải trong maybeinliers

{

if điểm phù hợp với maybemodel với một lỗi nhỏ hơn t thì thêm điểm vào alsoinliers

}

if số lượng các phần tử trong các alsoinliers > d {

% điều này ngụ ý rằng ta có thể đã tìm thấy một mô hình tốt

% bây giờ kiểm tra nó tốt như thế nào

bettermodel = model các tham số phù hợp cho tất cả các điểm trong maybeinliers và alsoinliers

thiserr = một thước đo của mô hình tốt như thế nào phù hợp với những điểm này

if thiserr < besterr {

bestfit = bettermodel

besterr = thiserr

}

}

tăng iterations

}

Return bestfit

\* Lưu đồ thuật toán được tham khảo từ

[https://en.wikipedia.org/wiki/Random\\_sample\\_consensus](https://en.wikipedia.org/wiki/Random_sample_consensus)

#### 2.7.4 Thông số

Giống như được mô tả trong phần đầu của thuật toán, RANSAC cần một số tham số được xác định trước cho kích thước của tập con mẫu (n), ngưỡng dung sai (t), ngưỡng đồng thuận tối thiểu (d) và số lần lặp lại (k). Ngoài ra, điều quan trọng là ước tính tỷ lệ inliers (w) trong tập dữ liệu, để tính toán một số các tham số này.

Vì Ransac là một thuật toán ngẫu nhiên nên cần phải ước tính chính xác các thông số, để tăng khả năng tìm ra mô hình tối ưu, trong khi vẫn giữ những lợi ích tính toán của một thuật toán ngẫu nhiên so với thuật toán xác định đầy đủ. Dưới đây là một số khám phá để tính toán các tham số này.

➤ **Tỷ lệ Inliers**

Mặc dù nó không phải là một tham số trực tiếp trong thuật toán, tỷ lệ inliers được sử dụng trong tính toán các tham số thuật toán và thậm chí nó có thể ảnh hưởng đến sự phức tạp của thuật toán ẩn. Do đó, sẽ có lợi nếu có một số thông tin về tỷ lệ outliers trong bộ dữ liệu trước khi chạy RANSAC.

➤ **Kích thước mẫu phân nhóm**

Kích cỡ của mẫu là số lượng các mẫu được chọn ngẫu nhiên bởi RANSAC để mô hình ban đầu tại mỗi lần lặp. Nó liên quan trực tiếp với mô hình dự định phù hợp với bộ dữ liệu. Ransac sử dụng số lượng tối thiểu các mẫu cần thiết để xác định mô hình làm kích cỡ tập hợp con mẫu. ví dụ. để phù hợp với một mô hình tuyến tính nó chọn 2 mẫu dữ liệu hoặc để phù hợp với một hình tròn mô hình nó chọn 3 mẫu dữ liệu như là 3 điểm sẽ là đủ để xác định một vòng tròn.

$n$  = số lượng mẫu tối thiểu để xác định mô hình

Ta có thể nghĩ rằng sử dụng nhiều mẫu dữ liệu hơn thì tập con nhỏ nhất sẽ là thuận lợi, vì có thể thu được một ước tính tốt hơn và chính xác hơn của mô hình. Tuy nhiên, có nhiều mẫu hơn trong tập hợp con mẫu sẽ làm tăng không gian tìm kiếm cho việc chọn tập hợp con. Vì vậy, để giữ xác suất tìm ra mô hình tối ưu ở cùng mức độ, chúng ta cần phải thử thêm tập con mẫu. Do đó, sự gia tăng số lượng lặp lại là cần thiết, chủ yếu làm tăng sự phức tạp về mặt tính toán, vượt trội hơn những ưu điểm của việc có một tập con lớn hơn. Đây là mối quan hệ giữa kích thước tập con và số lần lặp lại ảnh hưởng trực tiếp đến sự phức tạp.

➤ **Ngưỡng dung sai lỗi**

Ngưỡng dung sai lỗi được RANSAC sử dụng để xác định xem mẫu dữ liệu có đồng ý với mô hình hay không. Các mẫu dưới ngưỡng này sau đó sẽ tạo ra sự đồng

thuận cho mô hình đó, sẽ là các giá trị đầu vào của tập dữ liệu nếu tìm thấy đúng mô hình. Do đó, nó nên được lựa chọn theo các lỗi gaussian trong inliers.

➤ ***Nguưỡng đồng thuận tối thiểu (Minimum Consensus Threshold)***

Nguưỡng đồng thuận tối thiểu là số lượng tối thiểu các mẫu có thể được chấp nhận như một sự đồng thuận hợp lệ để tạo ra mô hình cuối cùng cho lần lặp đó. Vì RANSAC cố gắng nắm bắt các giá trị đầu vào có cấu kết đồng thuận, số lượng mẫu trong một sự đồng thuận hợp lệ có liên quan trực tiếp với số inliers trong tập dữ liệu. Do đó, RANSAC sử dụng một giá trị ngưỡng tương đương hoặc nhỏ hơn một chút so với số inliers để chấp nhận sự đồng thuận là hợp lệ

Nếu tổng số mẫu trong tập dữ liệu là [Data Set]

$$d \approx w . [Data\ Set]$$

➤ ***Số lần lặp***

Các thuật toán xác định toàn diện sẽ thử mọi tập hợp con có thể có để tìm ra một tập hợp con tốt nhất, nhưng thực tế nó không chỉ là tính toán không khả thi, mà còn không cần thiết. Do đó, thay vì một cách xác định, RANSAC chọn tập hợp con mẫu ngẫu nhiên. Tuy nhiên, cũng rất quan trọng để xác định số lượng các lựa chọn ngẫu nhiên này để có được một xác suất cao rằng RANSAC sẽ chọn một tập con mẫu mà không bao gồm các ngoại lệ.

Số lần lặp lại dự kiến để chạy thành công với xác suất xác định có thể được tính như sau:

Xác suất lựa chọn đầu vào:

$$P(inlier) \equiv w$$

Xác suất của việc chọn một tập hợp con n mẫu mà không có giá trị outlier:

$$P(subset\ with\ no\ outlier) \equiv w^n$$

Xác suất của việc chọn một tập hợp con n mẫu mà có giá trị outlier:

$$P(subset\ with\ outliers) \equiv 1 - w^n$$

Xác suất của việc chọn một tập hợp con n mẫu mà có giá trị outlier trong tất cả k lần lặp:

$$P(k\ subset\ with\ outliers) \equiv (1 - w^n)^k$$

Xác suất của một hoạt động không thành công

$$P(\text{fail}) \equiv (1 - w^n)^k$$

Xác suất của một hoạt động thành công

$$P(\text{success}) \equiv 1 - (1 - w^n)^k$$

Số lần lặp lại mong đợi

$$\Rightarrow k = \frac{\log(1 - P(\text{success}))}{\log(1 - w^n)} \quad (2.30)$$

Do đó, với xác định một xác suất thích hợp  $P(\text{success})$  theo tính thực tế mong muốn, số lần lặp có thể được ước tính.

Ví dụ. đây là một số số tính toán lặp đi lặp lại cho RANSAC :

$$[\text{Data Set}] \equiv 12, n \equiv 2, P(\text{success}) = 0.99$$

$$w = 0.95 \Rightarrow k = 2$$

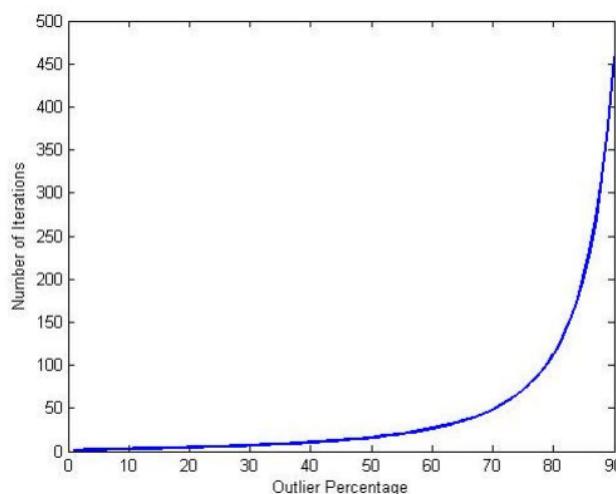
$$w = 0.75 \Rightarrow k = 6$$

$$w = 0.6 \Rightarrow k = 11$$

$$w = 0.5 \Rightarrow k = 17$$

Hơn nữa, số lần lặp lại có thể tăng đáng kể khi tỷ lệ outlier trong tập dữ liệu tăng lên.

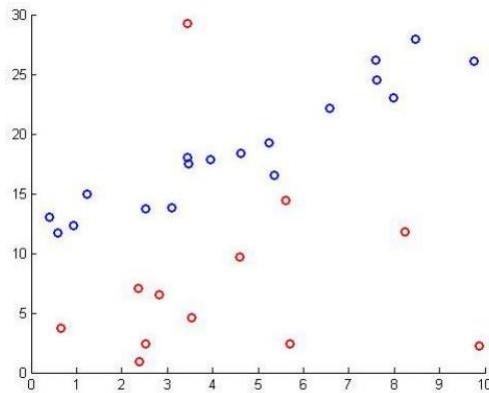
Với  $n \equiv 2, P(\text{success}) = 0.99$



**Hình 2.25:** Tỉ lệ outlier trong tập dữ liệu

### 2.7.5 Bài toán thử nghiệm :

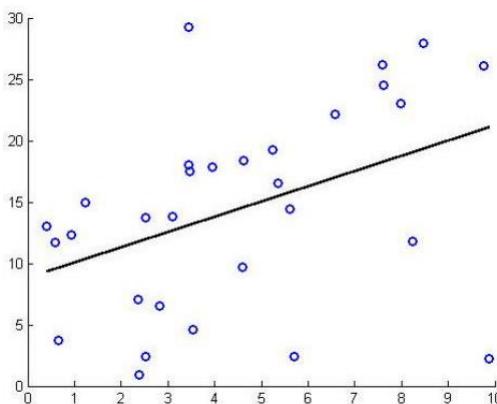
#### *Ước lượng chức năng tuyến tính*



30 mẫu dữ liệu được tạo ra với  $y = 2x + 5$  và  $w = 0,6$  với tổng sai sót để gây ra các outlier và với tiếng ồn gaussian trắng trên inliers. Outliers được tô màu đỏ, inliers là màu xanh.

#### Hồi quy tuyến tính

Đây là kết quả của phương pháp hồi quy tuyến tính mà không có RANSAC. Do không xem xét các mẫu có sai sót, nên mẫu phù hợp được tạo ra sẽ bị ảnh hưởng rất lớn và sai lệch bởi các outlier.

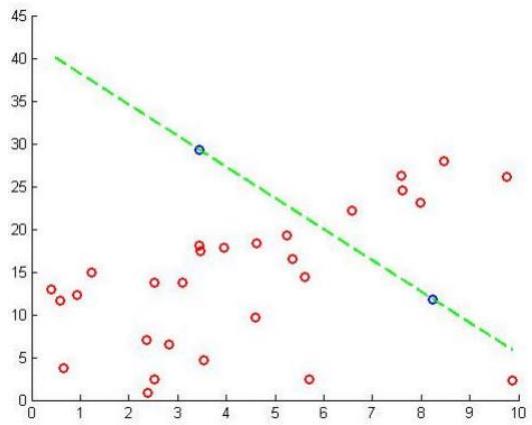


#### Các ví dụ về sự lặp RANSAC

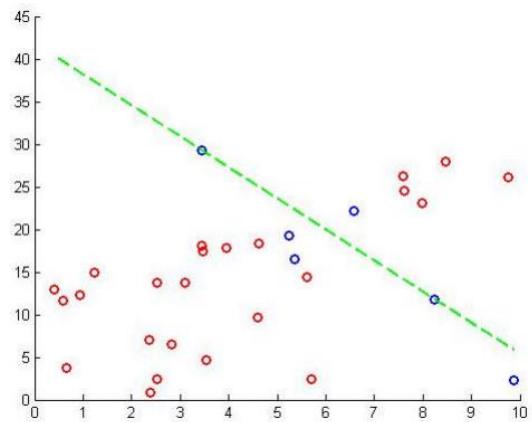
Dưới đây là một số đồ thị trên tập dữ liệu từ các lần lặp lại của RANSAC.

#### Lần lặp 1

Trong lần lặp lại này, tập hợp con ngẫu nhiên bao gồm hai giá trị outlier.

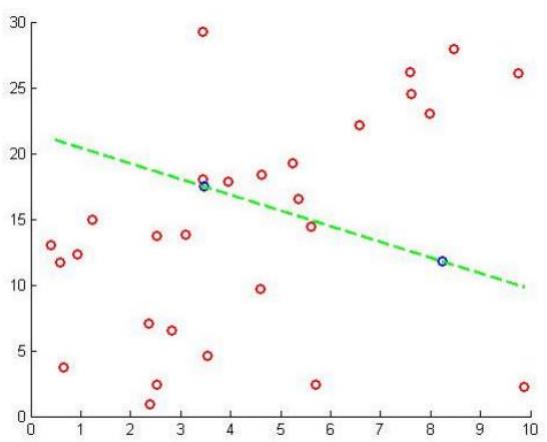


Vì mô hình được tạo ra bởi outlier, nó chắc chắn không phải là một ước tính tốt cho dòng. Do đó nó sẽ không vượt qua ngưỡng thỏa thuận tối thiểu như mong muốn.

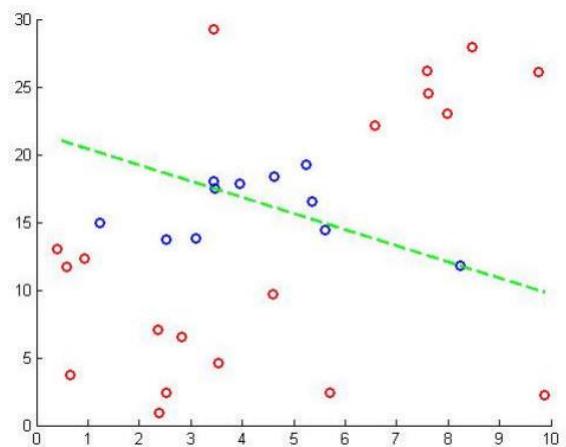


### Lần lặp 2

Trong lần lặp lại này, tập con mẫu ngẫu nhiên bao gồm một giá trị inlier và một giá trị outlier.

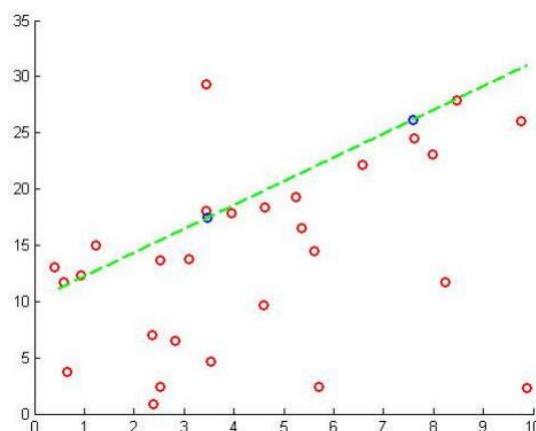


Do sự tồn tại của một outlier trong tập hợp con mẫu, mô hình được tạo ra không có đủ hỗ trợ, vì vậy nó sẽ không vượt qua ngưỡng đồng thuận tối thiểu

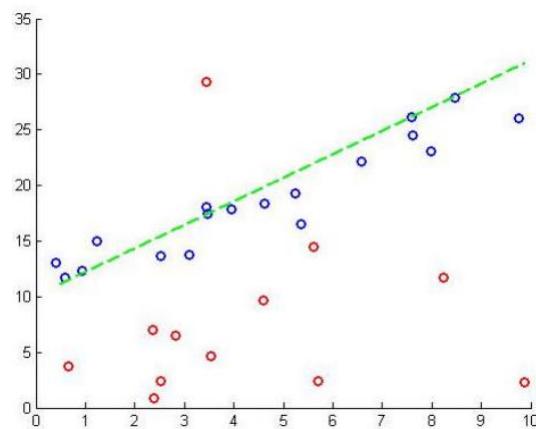


### Lần lặp 3

Trong lần lặp lại này, tập hợp con ngẫu nhiên bao gồm hai giá trị inlier



Do tập hợp con mẫu chỉ bao gồm các giá trị inlier, với ngưỡng chịu dung sai phù hợp, tất cả các giá trị inlier có thể được thu thập như là sự đồng thuận.



Có tất cả các inlier trong sự đồng thuận, mô hình này vượt qua ngưỡng đồng thuận tối thiểu. Sau đó, một mô hình cuối cùng được tính bằng cách sử dụng tất cả các mẫu dữ liệu trong sự đồng thuận với một tối ưu hóa leastsquare.

Đến đây, RANSAC bỏ qua tất cả các outlier và đào tạo các mô hình chỉ sử dụng inliers.

❖ Ảnh hưởng của tham số

- Số lần lặp lại

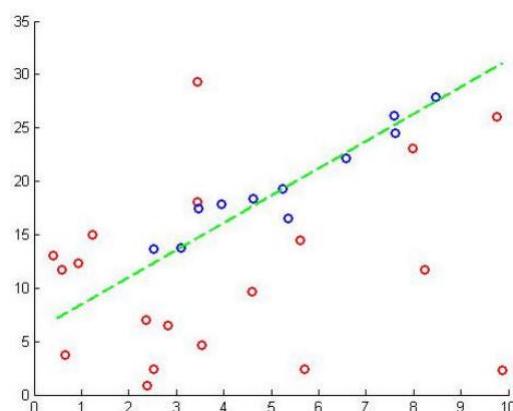
$$W = 0.6 ; P(\text{success}) = 0.99 ; n = 2$$

$$\Rightarrow k = \frac{\log(1 - P(\text{success}))}{\log(1 - w^n)} = \frac{\log(1 - 0.99)}{\log(1 - 0.6^2)} \approx 11$$

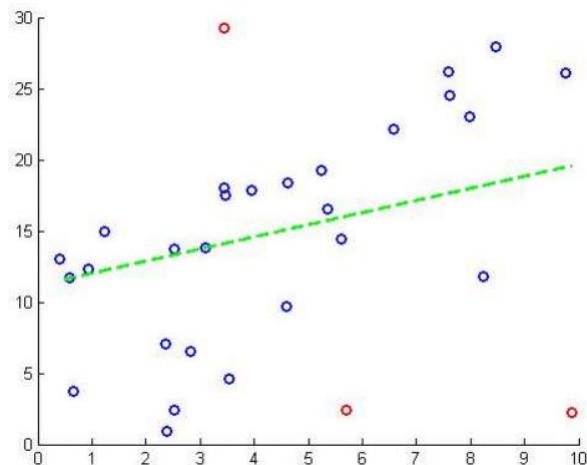
- Ngưỡng dung sai lỗi

Ngưỡng dung sai lỗi nên được chọn phù hợp với cài đặt. Nếu không, RANSAC sẽ không tìm được kết quả chính xác.

Nếu t được chọn nhỏ hơn cần thiết, ngay cả RANSAC đã chọn hai giá trị inliers cho tập con mẫu, nó sẽ không đạt được sự đồng thuận bao gồm tất cả các inliers.

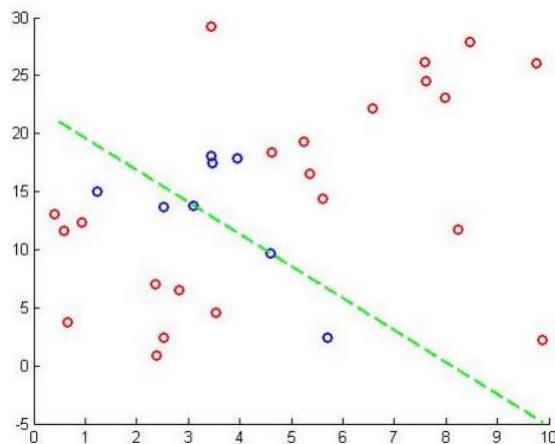


Hoặc nếu t được lựa chọn lớn, sự đồng thuận tạo ra sẽ bao gồm một số outliers ngay cả khi tập con mẫu chỉ bao gồm các giá trị inliers.

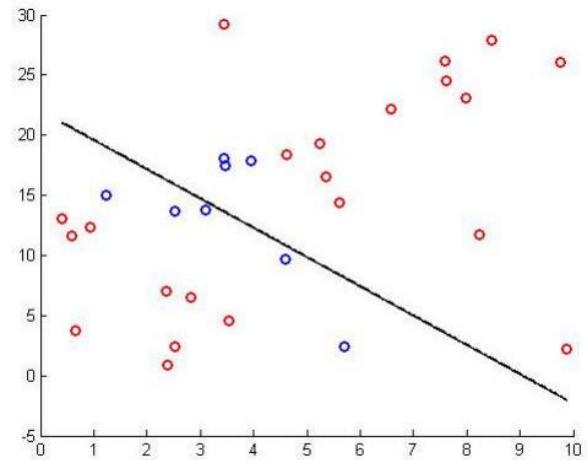


- NguỒng đồng thuận tối thiểu

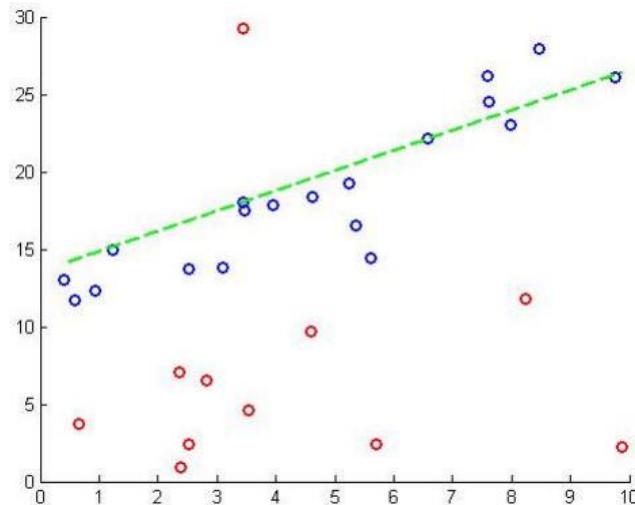
NguỒng đồng thuận tối thiểu cũng có một vai trò quan trọng đối với hành vi của RANSAC. Nếu  $d$  được chọn nhỏ, một số dòng ngẫu nhiên phù hợp với một phần của các mẫu sẽ được coi là một sự đồng thuận hợp lệ.



Một mô hình cuối cùng sẽ được tạo ra với các tham số của thiết lập sai này.



Hoặc, nếu  $d$  được chọn quá lớn, thậm chí một ước tính tốt sẽ không vượt qua ngưỡng đồng thuận tối thiểu và không có kết quả mô hình cuối cùng nào được tìm thấy bởi RANSAC.



Tóm lại quá trình thực hiện thuật toán RANSAC được mô tả như dưới đây:

Từ tập dữ liệu đầu vào gồm có nhiều và không nhiều ta chọn dữ liệu ngẫu nhiên, tối thiểu để xây dựng mô hình

Tiến hành xây dựng mô hình với dữ liệu đó, sau đó đặt ra một ngưỡng dùng để kiểm chứng mô hình.

Gọi tập dữ liệu ban đầu trừ đi tập dữ liệu để xây dựng mô hình là tập dữ liệu kiểm chứng. Sau đó, tiến hành kiểm chứng mô hình đã xây dựng bằng tập dữ liệu kiểm chứng. Nếu kết quả thu được từ mô hình vượt quá ngưỡng, thì điểm đó là nhiễu, còn không đó sẽ là ngược lại.

Quá trình này sẽ được lặp đi lặp lại trong vài lần. Với được tính theo công thức

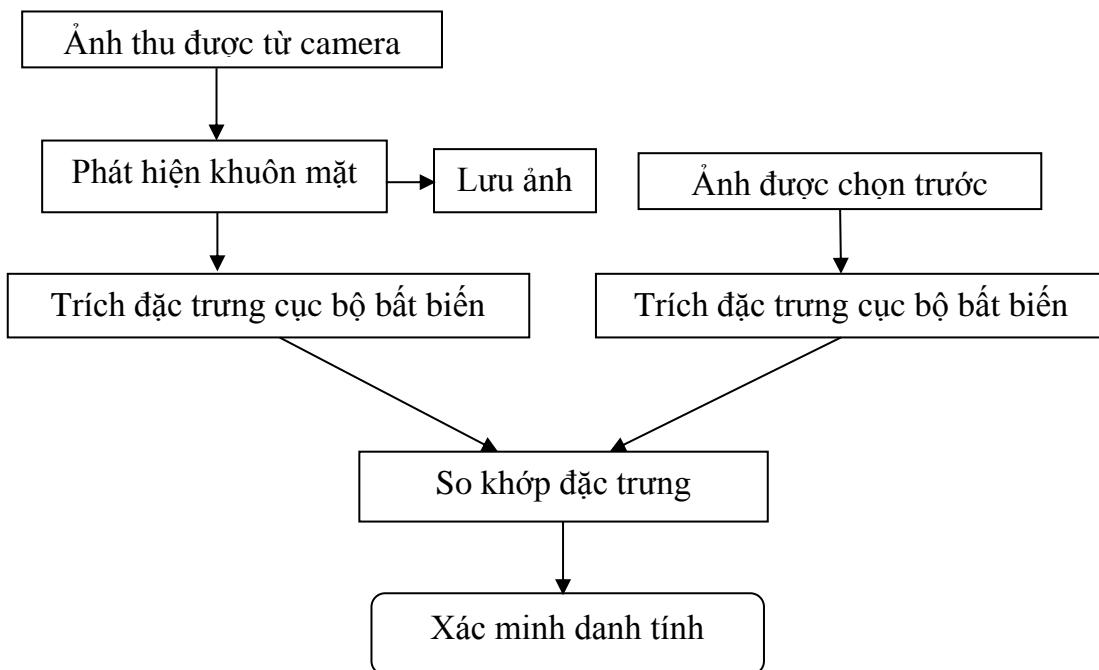
trên. Tại mỗi vòng lặp giá trị của sẽ được tính lại.

Kết quả là mô hình nào có số dữ liệu không nhiễu nhiều nhất sẽ được chọn là mô hình tốt nhất.

## CHƯƠNG 3

# XÂY DỰNG CHƯƠNG TRÌNH

### 3.1 Quá trình nhận dạng và định danh khuôn mặt

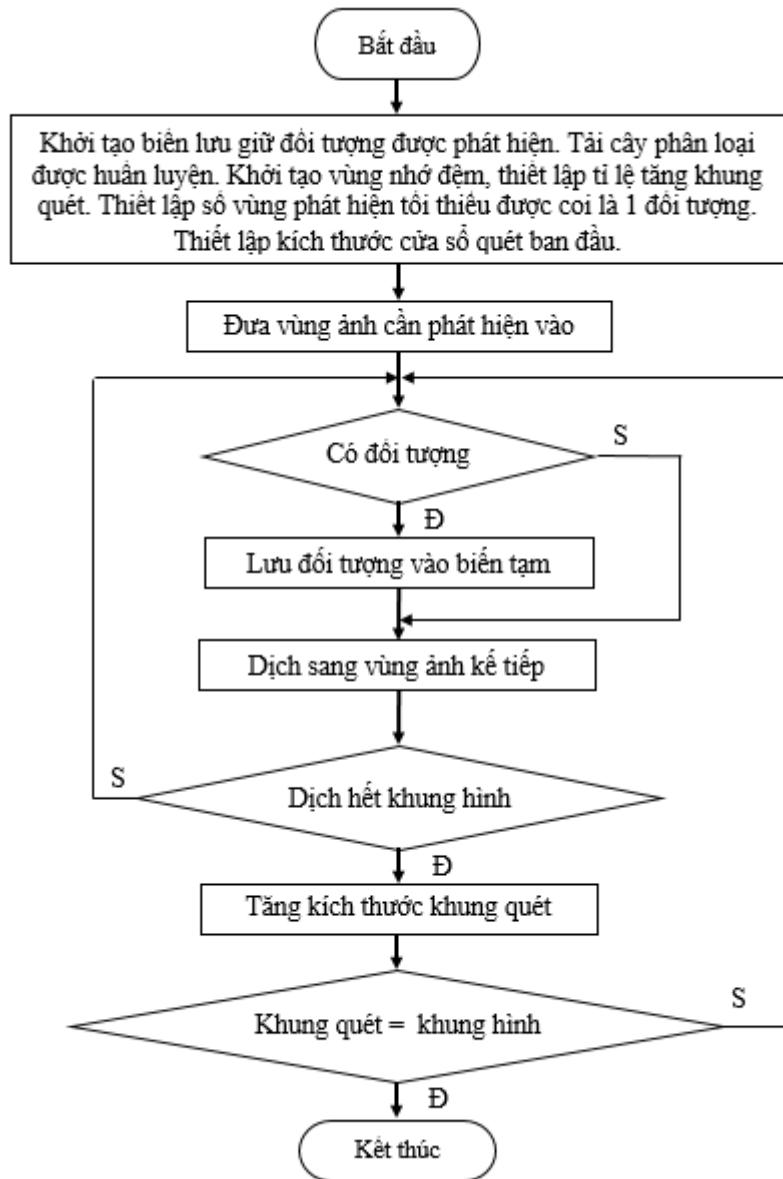


**Hình 3.1:** Sơ đồ nhận dạng khuôn mặt

Từ camera ta thu được một ảnh, dựa vào các đặc trưng Haar Like và mô hình Cascade ta xác định khuôn mặt trong ảnh. Để tăng độ nhận danh chính xác khuôn mặt trong ảnh, ta cần giảm bỏ các yếu tố không cần thiết trong ảnh, trong luận văn này tôi cắt khuôn mặt ra so sánh với khuôn mặt trong cơ sở dữ liệu.

### 3.2 Thuật toán phát hiện đối tượng

Đối tượng trong luận văn này là khuôn mặt, tập huấn luyện “haarcascade\_frontalface\_default.xml” được lấy trong thư viện Emgu CV

**Hình 3.2:** Lưu đồ phát hiện đối tượng

- Bước 1: Sau khi đã khởi tạo các giá trị ban đầu cho thuật toán như trên lưu đồ, vùng ảnh cần phát hiện đối tượng được đưa vào phân tích (kích thước ban đầu do người lập trình qui định, thường là 20x20 Pixel). Vùng ảnh này bắt đầu từ góc trên bên trái của khung hình. Sau đó vùng ảnh này sẽ được đưa qua cây phân loại đã được huấn luyện bằng thuật toán Adaboost và đặc trưng Haarlike trước đó. Qua bước này chương trình sẽ biết được liệu trong vùng ảnh đưa vào có đối tượng cần phát hiện hay không. Nếu có thì đối tượng sẽ được lưu vào một biến tạm.

- Bước 2: vùng ảnh ban đầu sẽ được dịch sang bên trái 1 đến 2 pixel và bắt đầu lại bước 1 cho đến khi nó dịch hết toàn màn hình. Các đối tượng phát hiện được cũng được lưu vào biến tạm.
- Bước 3: chương trình tăng kích thước của vùng ảnh ban đầu. Tỉ lệ tăng phụ thuộc vào thông số mà người lập trình thiết lập. Nếu là 1.1 thì tỉ lệ tăng là 10%, nếu là 1.2 thì tỉ lệ tăng là 20%,... Sau đó thực hiện lại bước 1 và bước 2 như trên đã đề cập cho đến khi kích thước của vùng ảnh bằng kích thước của khung hình thì dừng chương trình.

Đây là tập dữ liệu thu được từ camera



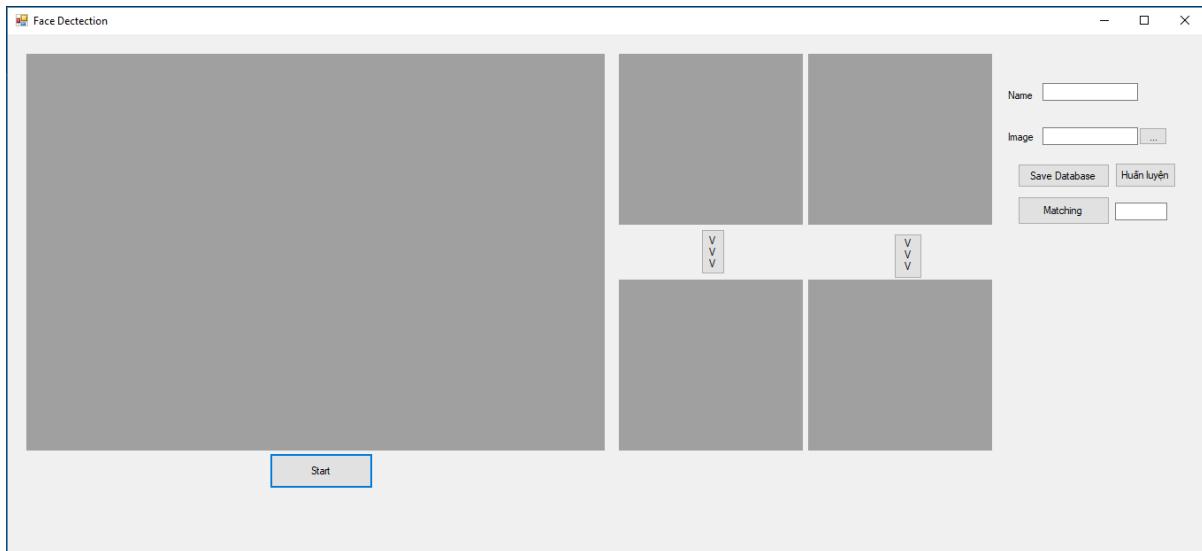
**Hình 3.3:** Tập dữ liệu khuôn mặt

### 3.3 Cài đặt chương trình

Chương trình được cài đặt trên Dell Inspiron 7537 hệ điều hành Windows 10, Ram 6G và bộ xử lý CORE i5-4210U CPU @ 1.7GHz.

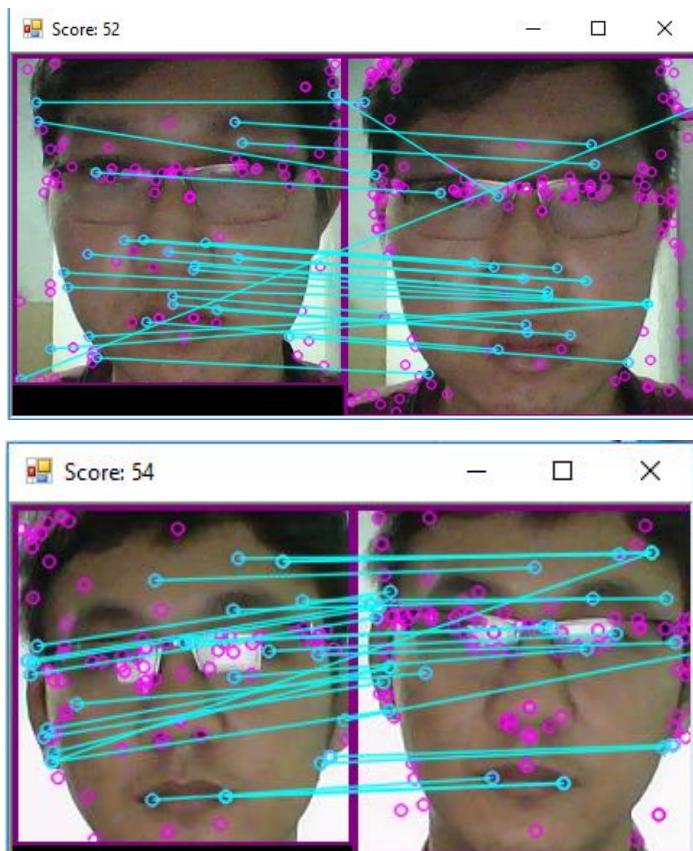
Sử dụng Visual Studio 2013 và thư viện Emgu CV 3.1.0

### 3.4 Chương trình mô phỏng



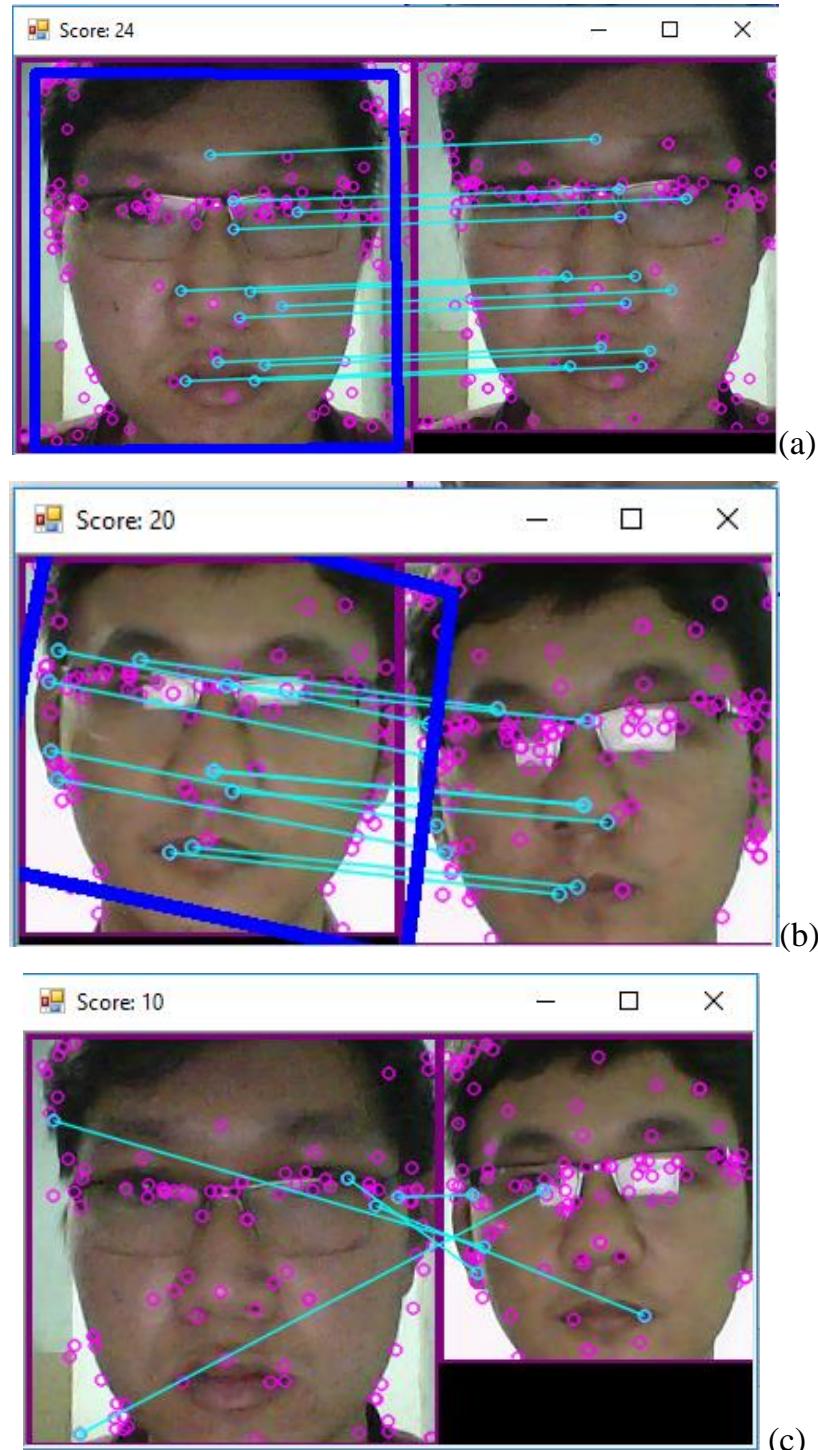
- Một số kết quả

Đây là hai ảnh so khớp các đặc trưng dùng SIFT, chưa sử dụng RANSAC , ta thu được số lượng so khớp nhiều bao gồm luôn các outlier



**Hình 3.4:** So khớp 2 ảnh dùng SIFT

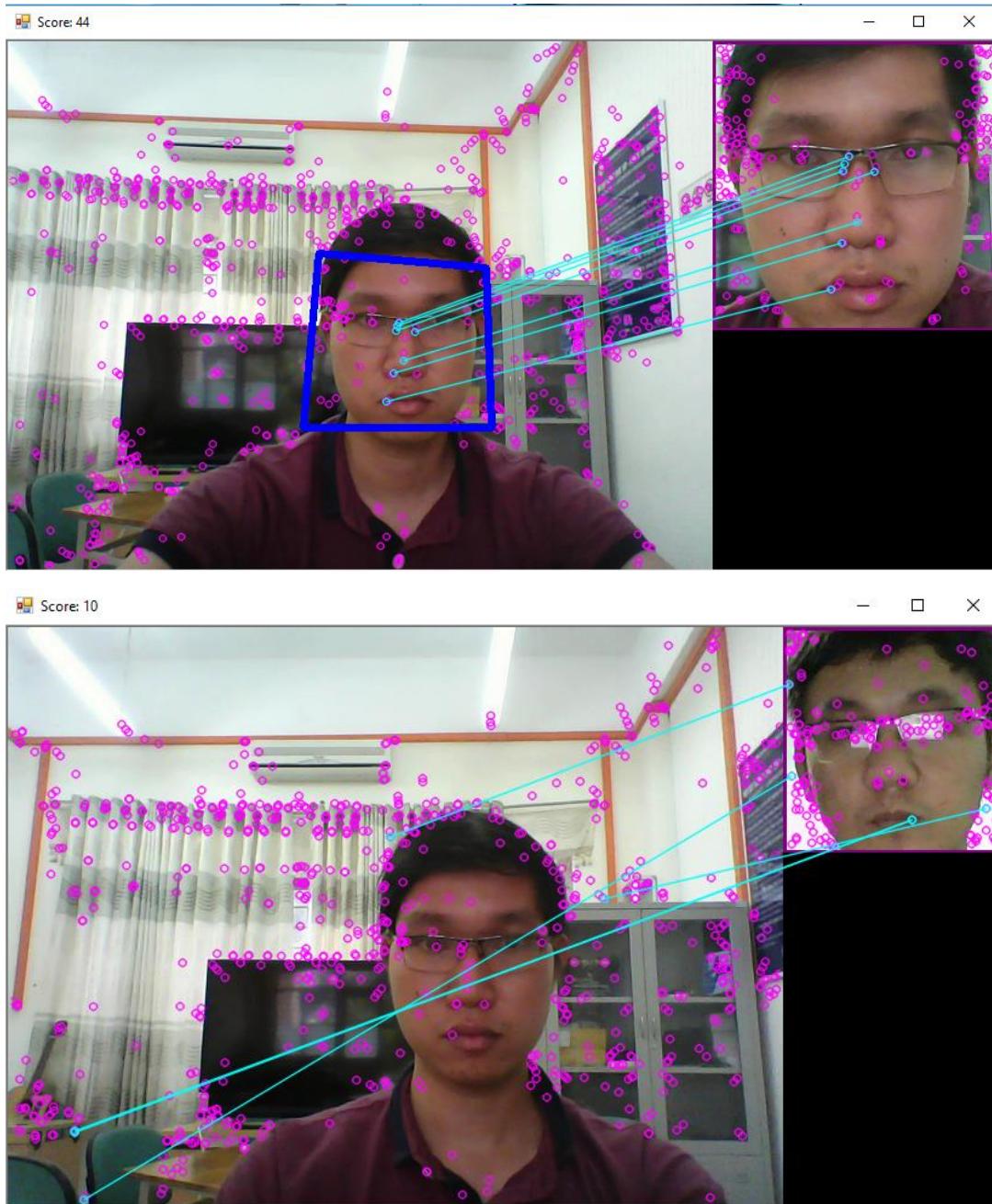
Đây là hai ảnh so khớp các đặc trưng dùng SIFT và sử dụng RANSAC, ta thu được số lượng so khớp nhiều ít hơn nhưng đã loại bỏ được các outlier. Tùy vào giá trị cài đặt ngưỡng Ransac mà ta thu được số lượng so khớp.



**Hình 3.5:** So khớp 2 ảnh dùng SIFT kết hợp RANSAC

Hình 3.5 a Kết quả đối sánh giữa 2 ảnh cùng một người ta có được 24 điểm so khớp. Hình 3.5 (b) Ảnh bị biến đổi bởi phép xoay, ảnh trên thu được số 20 điểm so khớp. Hình 3.5 (c) với hai khuôn mặt khác nhau vẫn thu được 10 điểm so khớp.

Với ảnh bị biến đổi bởi phép xoay, co dãn và che lấp một phần, chương trình đã đối sánh chính xác . Điều này cho thấy SIFT bắt biến với phép xoay, thu phóng và không yêu cầu tính toàn vẹn của ảnh.



**Hình 3.6:** Thực hiện đối sánh ảnh từ camera và ảnh được lưu trong cơ sở dữ liệu

### 3.5 Định danh khuôn mặt

Dựa vào các kết quả mô phỏng ta có thể thấy rằng dùng các đặc trưng Haar và thuật toán Adaboost ta phát hiện được khuôn mặt trong camera. Ta so khớp khuôn mặt đó lần lượt với từng khuôn mặt trong cơ sở dữ liệu. Nếu số lượng so khớp của khuôn mặt nào trong danh sách là lớn nhất, ta có thể kết luận đó là người trong ảnh. Để tránh tình trạng một khuôn mặt bất kỳ không có trong cơ sở dữ liệu ta vẫn thu được số lượng so khớp giữa 2 ảnh nhưng không chính xác, ta nên cài đặt giá trị score để tránh tình trạng nhận danh nhầm.

### 3.6 Thực nghiệm

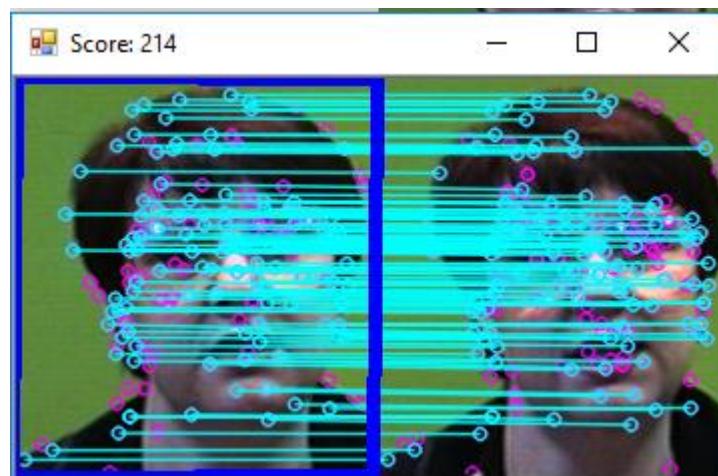
Chương trình đã sử dụng bộ hình ảnh khuôn mặt face94, face95, face96 và grimace của Tiến sĩ Libor Spacek với :

- Tổng số cá nhân: 395
- Số hình ảnh trên mỗi cá nhân: 20
- Tổng số hình ảnh: 7900
- Giới tính: chứa hình ảnh của các đối tượng nam và nữ
- Chủng tộc: chứa hình ảnh của những người có nguồn gốc chủng tộc khác nhau
- Phạm vi tuổi: hình ảnh chủ yếu là sinh viên năm nhất, do đó, phần lớn các cá nhân là từ 18-20 tuổi, nhưng một số cá nhân lớn tuổi cũng có mặt.
- Kính: Có
- Gáu: Có
- Định dạng hình ảnh: JPEG màu 24bit
- Máy ảnh đã sử dụng: Máy quay S-VHS
- Ánh sáng: nhân tạo, hỗn hợp vonfram và huỳnh quang trên không

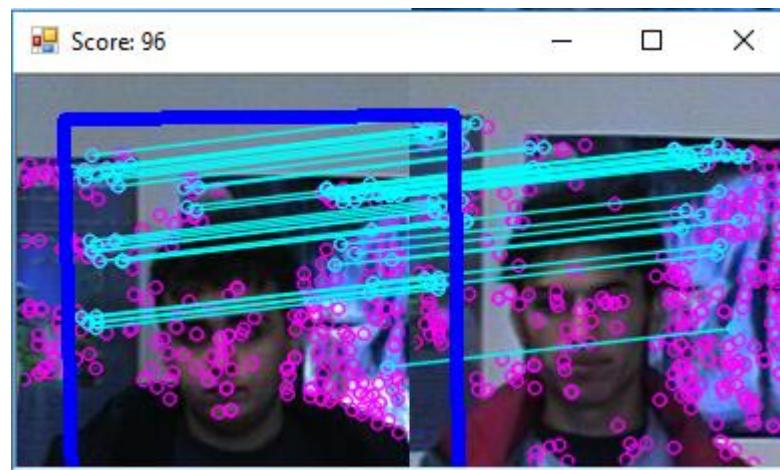
Ta được kết quả sau:

- Các ảnh chứa khuôn mặt trong điều kiện ánh sáng bình thường không có tác động bởi ngoại cảnh thì số lượng so khớp của hai ảnh cao trên 100 điểm và các điểm đó đều nằm trên khuôn mặt.

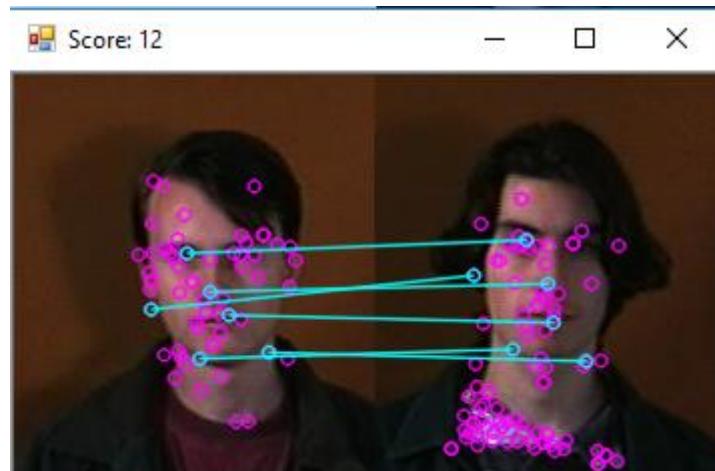
- Các ảnh có chứa khuôn mặt và chịu tác động bởi ngoại cảnh thì số lượng so khớp của hai ảnh cũng cao nhưng có nhiều điểm không nằm trên khuôn mặt.
- Các ảnh có chứa khuôn mặt trong điều kiện ánh sáng tối thì số lượng so khớp của hai ảnh thấp và dễ bị nhầm lẫn.



**Hình 3.7:** Hai ảnh chứa khuôn mặt không có ngoại cảnh ở ánh sáng bình thường  
Số lượng so khớp nằm hoàn toàn nằm trong vùng mình cần nhận dạng và kết quả nhận dạng là chính xác.

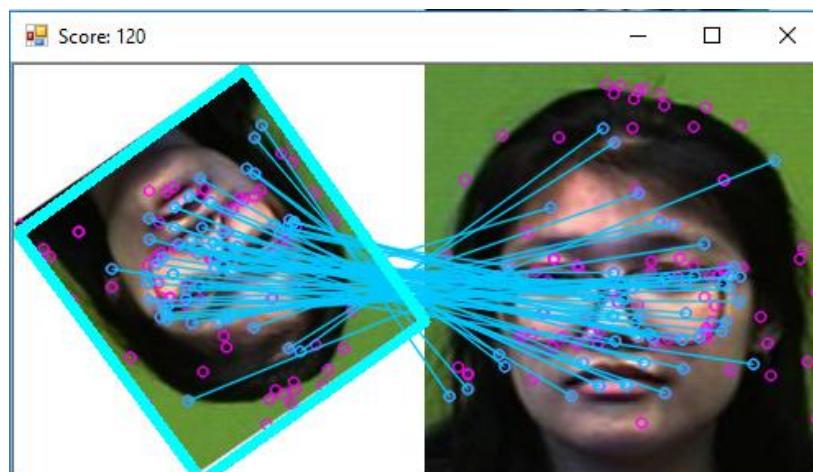


**Hình 3.8:** Hai ảnh chứa khuôn mặt có ngoại cảnh  
Số lượng so khớp giữa hai ảnh là nhiều nhưng những điểm này không nằm trên đối tượng mình cần nhận dạng.



**Hình 3.9:** Hai ảnh chứa khuôn mặt ở ánh sáng tối

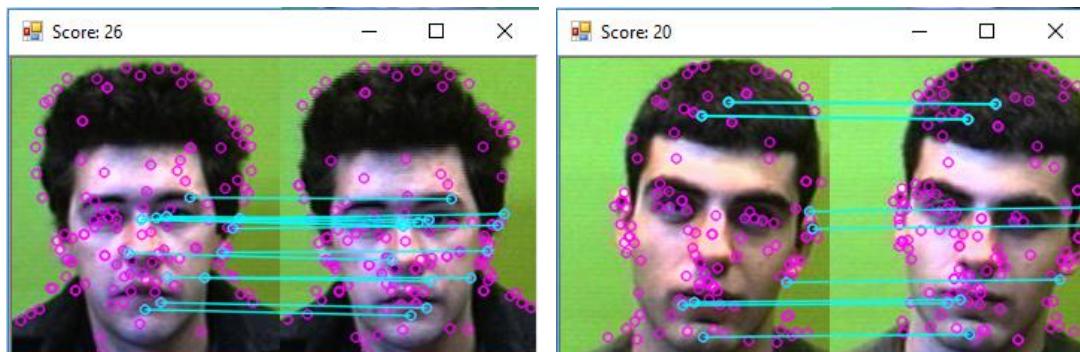
Số lượng so khớp giữa hai hình ít và các điểm so khớp gần như tại các vị trí giống nhau. Nhưng đây là 2 khuôn mặt khác nhau, kết quả nhận dạng là không chính xác.

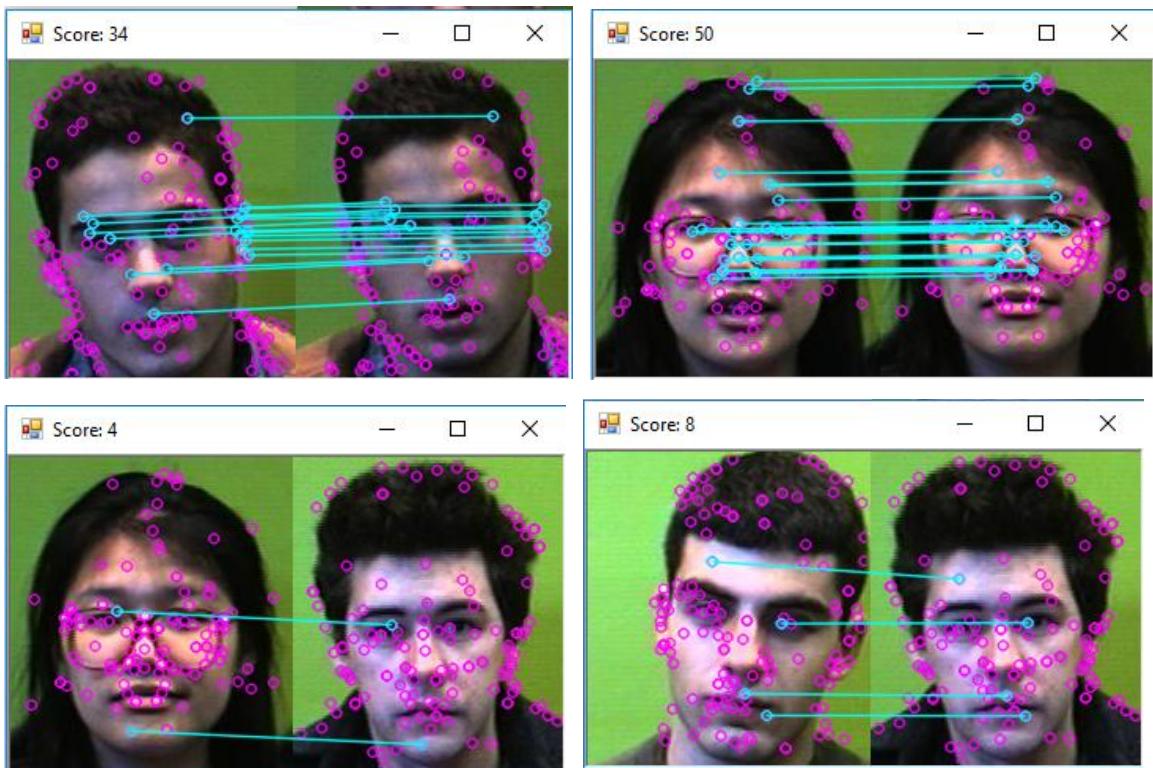


**Hình 3.10** So khớp giữa hai ảnh bị xoay

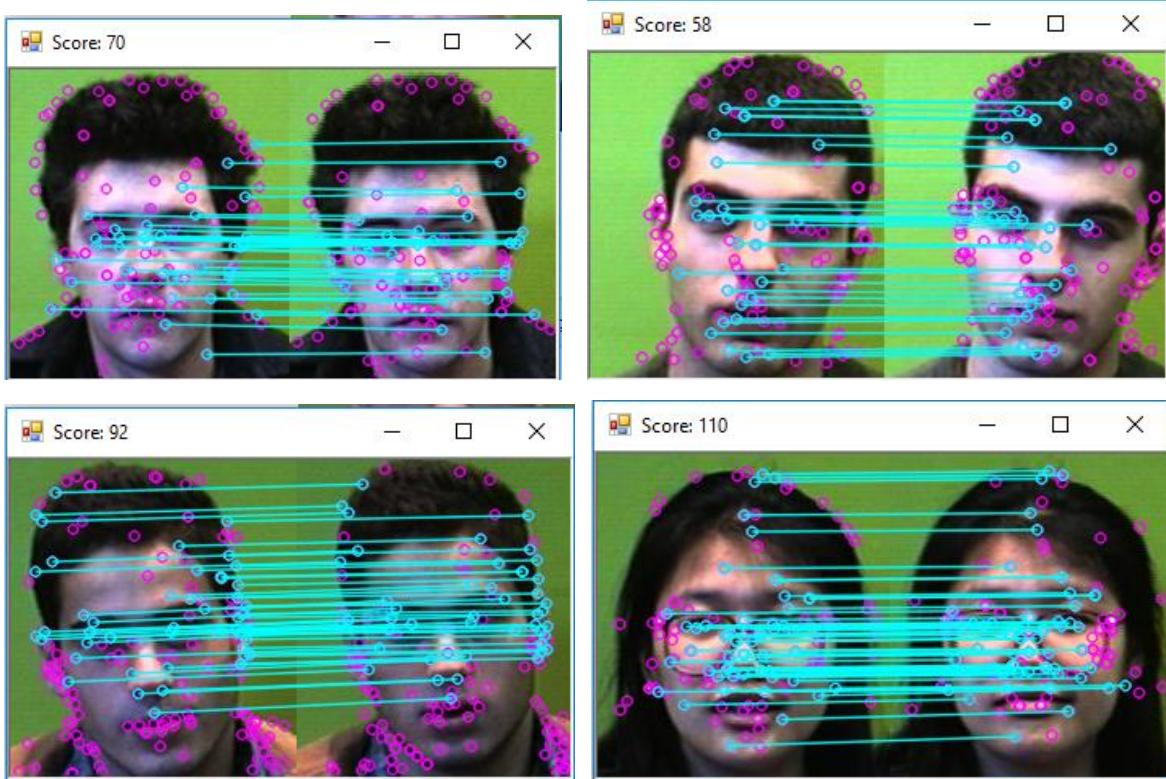
So khớp giữa hai ảnh không bị ảnh hưởng khi ảnh bị xoay hay co giãn.

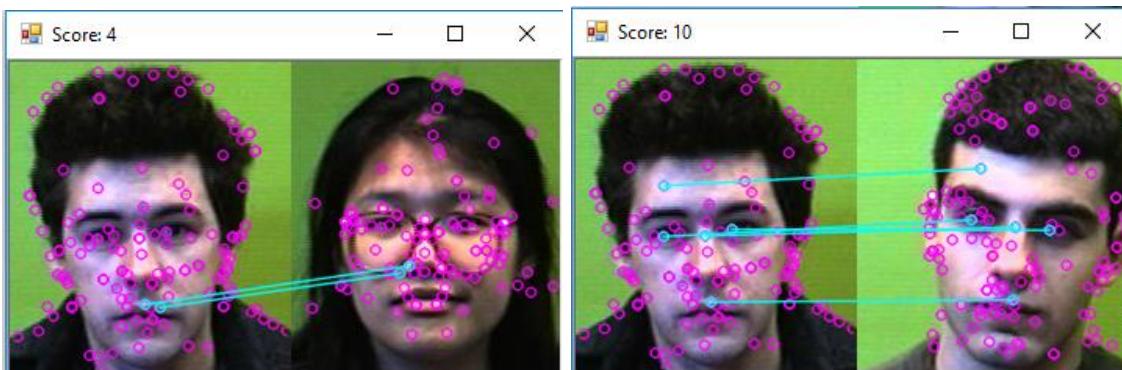
- So khớp hai ảnh với ngưỡng RANSAC 0.5





- So khớp hai ảnh với ngưỡng RANSAC 1.5





Dựa vào các kết quả thu được ta có thể thấy giá trị ngưỡng RANSAC làm cho số lượng so khớp giữa hai ảnh thay đổi. Khi giá trị ngưỡng RANSAC càng cao thì số lượng so khớp giữa hai ảnh tăng lên, có thể xảy ra trường hợp số lượng so khớp giữa hai ảnh không cùng một người lớn hơn giá trị cài đặt, dẫn đến định danh nhầm. Hoặc giá trị ngưỡng RANSAC nhỏ thì số lượng so khớp thấp, do đó số lượng so khớp giữa hai ảnh cùng một người có thể nhỏ hơn giá trị cài đặt, dẫn đến tình trạng là không định danh được.

Tập dữ liệu	Tổng số ảnh	Ngưỡng RANSAC	Tỷ lệ nhận dạng
Face94	3040	0.5	86.84%
		1.5	93.42%
Face95	1440	0.5	58.3%
		1.5	72.2%

**Bảng 3.1:** Tỷ lệ nhận dạng trên các tập dữ liệu

## CHƯƠNG 4

# KẾT LUẬN

### 4.1 Kết luận chung

SIFT + RANSAC là một thuật toán rất mạnh và phức tạp trong các bài toán đối sánh ảnh. Trong luận văn này tôi đã tìm hiểu và cài đặt thuật toán với đầy đủ các bước cơ bản của SIFT , xây dựng chương trình ứng dụng mô phỏng việc đối sánh ảnh tương tự sử dụng SIFT và dùng RANSAC để giảm bớt các đối sách không đúng

#### Ưu điểm

- + Phát hiện được khuôn mặt qua webcam.
- + Tốc độ phát hiện đối tượng và nhận dạng nhanh.

Tuy nhiên, đề tài cũng còn tồn tại các hạn chế:

- + Độ chính xác nhận dạng phụ thuộc nhiều vào cường độ ánh sáng

### 4.2. Kiến nghị

Luận văn đã nghiên cứu chi tiết các thuật toán, cách thức hoạt động và ưu nhược điểm của từng thuật toán. Trong thời gian tới tác giả sẽ cố gắng nghiên cứu có thể sử dụng các phương pháp trích đặc trưng khác kết hợp với nhau để nâng cao hiệu quả nhận dạng.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Châu Ngân Khánh và Đoàn Thanh Nghị. *Nhận Dạng Mặt Người Với Giải Thuật Haar Like Feature – Cascade Of Boosted Classifiers Và Đặc Trung SIFT*. Tạp chí khoa học trường đại học An Giang Quyển 3 (2), trang 15 – 24 năm 2014
- [2] Rainer Lienhart, Alexander Kuranov, Vadim Pisarevsky. *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection*, MRL Technical Report 2002
- [3] P.Viola, M.Jones, *Rapid Object Detection using a Boosted Cascade of Simple Features*, Computer Vision and Vision and Pattern Recognition. In CVPR 2001, Proceeding of the 2001 IEEE Computer Society Conference on (Volume:1 ), Page(s):I-511 - I-518 vol.1,2001
- [4] David G. Lowe, *Distinctive Image Featuresfrom Scale-Invariant Keypoints*, Computer Science Department, University of British Columbia 2004
- [5] Nguyễn Thị Lan. Luận văn tốt nghiệp “*Truy vấn thông tin dựa trên việc đối sánh ảnh qua các đặc điểm bất biến*”
- [6] Kamarul Hawari Ghazali. *Feature Extraction technique using SIFT keypoints descriptors*. The International Conference on Electrical and Engineering and Informatics Institut technology Bandung, Indonesia, june 17-19, 2007
- [7] Nguyễn Thị Hoàn. *Phương pháp trích chọn đặc trưng ảnh trong thuật toán học máy tìm kiếm ảnh áp dụng vào bài toán tìm kiếm sản phẩm*, Đại học quốc gia Hà Nội. 2010
- [8] Faraj Alhwarin, Chao Wang, Danijela Risti -Durrant, Axel Gräser, *Improved SIFT-Features Matching for Object Recognition, Institute of Automation*, University of Bremen. 2008
- [9] Harris C. and Stephens M. , *A combined corner and edge detector*, Proceedings of the Alvey Vision Conference.1998

- [10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, *SURF: Speeded Up Robust Features*
- [11] Orhan-Sonmez RANSAC, 2006 (“<http://www.math-info.univ-paris5.fr/~lomn/Cours/CV/SeqVideo/Material/RANSAC-tutorial.pdf>”)
- [12] Wikipedia - RANSAC (<http://en.wikipedia.org/wiki/RANSAC>)
- [13] Vinay .A, Avani S Rao, Vinay S Shekhar, Akshay Kumar C, K N Balasubramanya Murthy, S Natarajan *Feature Extraction using ORB-RANSAC for Face Recognition* , PES University and PES Institute of Technology India 2015
- [14] Massimiliano Di Mella and Francesco Isgr`o *Face Recognition from Robust SIFT Matching* Dipartimento di Ingegneria Elettrica E Delle Tecnologie Dell’Informazione, Universit`a Degli Studi di Napoli Federico II, Napoli, Italy 2015

# NHẬN DẠNG VÀ ĐỊNH DANH KHUÔN MẶT NGƯỜI SỬ DỤNG THUẬT TOÁN SIFT - RANSAC

<sup>(1)</sup>Lê Nguyễn Anh Huy, <sup>(2)</sup>Nguyễn Văn Thái

<sup>(1)</sup> Trường đại học Sư phạm Kỹ thuật TP.HCM

<sup>(2)</sup> Trường đại học Sư phạm Kỹ thuật TP.HCM

## TÓM TẮT

Bài báo này xây dựng ứng dụng xử lý ảnh – thị giác máy tính vào việc phát hiện và nhận dạng khuôn mặt từ camera 2D. Đầu tiên là dùng các đặc trưng Haar like và Adaboost để phát hiện khuôn mặt người trong khung ảnh. Tính năng trích đặc trưng cục bộ bát biến SIFT (*Scale Invariant Feature Transform*) là thuật toán được sử dụng để phát hiện và mô tả các tính năng cục bộ, các tính năng chuyển đổi và xoay trong các hình ảnh. Khi đã xác định được khuôn mặt thành công, chương trình sẽ trích các đặc trưng SIFT của khuôn mặt để tìm kiếm các điểm hấp dẫn (key-points) và tạo ra bộ mô tả SIFT, kết hợp với thuật toán RANSAC (*Random Sample Consensus*) như là một bước sau xử lý để loại bỏ các key-points và nhiễu (outliers) và do đó làm tăng hiệu quả trong việc đưa ra một hệ thống mạnh mẽ để nhận ra hình ảnh khuôn mặt. So khớp các đặc trưng các khuôn mặt, so sánh số lượng so khớp để xác định khả năng tương đồng giữa hai ảnh.

**Từ khóa :** Nhận dạng khuôn mặt, SIFT , RANSAC

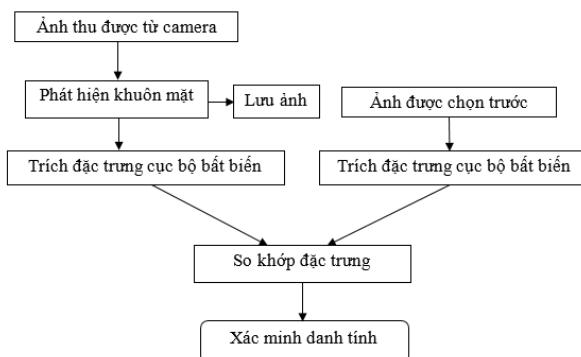
## ABSTRACT

This article builds the image-processing computer vision app for face detection and recognition from a 2D camera. The first is the use of Haar like and Adaboost features to detect the human face in the photo frame. SIFT (Scale Invariant Feature Transform) is an algorithm used to detect and describe local features, converting and rotating features in images. Once the face has been determined successfully, the program extracts the facial features of SIFT to search for key-points and creates a SIFT descriptor, combined with the RANSAC algorithm (Random Sample Consensus) as a step-by-step process to remove key-points and outliers and increase the efficiency of delivering a powerful system for recognizing facial images. Matching the features of the faces, comparing the number of matches to determine the similarity between the two images.

**Từ khóa :** Face detection, SIFT , RANSAC

## I. GIỚI THIỆU

Nhận dạng khuôn mặt là một trong những lĩnh vực mới của xử lý ảnh. Vào ngày nay nhận dạng được ứng dụng rộng rãi trong nhiều lĩnh vực của đời sống như nhận dạng trong lĩnh vực thương mại, phát hiện tội phạm trong lĩnh vực an ninh, hay trong lĩnh vực xử lý video, hình ảnh. Hiện nay có rất nhiều các phương pháp nhận dạng khác nhau được xây dựng để nhận dạng một người cụ thể trong thế giới thực. Hệ thống nhận dạng mặt người bao gồm hai bước: phát hiện khuôn mặt và định danh đối tượng. Công việc chính của bài báo này là dựa vào các kỹ thuật rút trích đặc trưng cục bộ bắt biến từ ảnh đối tượng và thực hiện đối sánh để định danh.



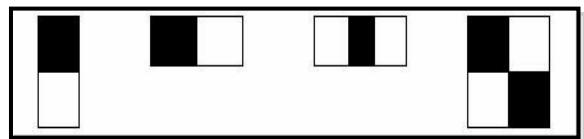
Hình 1.1 : Sơ đồ nhận dạng khuôn mặt

## II. CƠ SỞ LÝ THUYẾT

### 2.1 Đặc trưng Haar Like

Đặc trưng Haar Like [3] được tạo thành bằng việc kết hợp các hình chữ nhật đen, trắng với nhau theo một trật tự, một kích thước nào đó dùng tính độ chênh lệch giữa các giá trị điểm ảnh trong các vùng kè

nhau. Hình dưới đây mô tả 4 đặc trưng Haar Like cơ bản như sau:



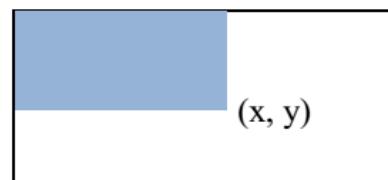
Hình 2.1: Các đặc trưng Haar Like cơ bản

Dùng các đặc trưng trên, ta có thể tính được giá trị của đặc trưng Haar-like là sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng như trong công thức sau:

$$f(x) = \text{Tổng}_{\text{vùng đen}}(\text{mức xám của pixel}) - \text{Tổng}_{\text{vùng trắng}}(\text{mức xám của pixel})$$

### 2.2 Integral Image

Integral Image [3] là một mảng hai chiều với kích thước bằng kích thước của ảnh cần tính giá trị đặc trưng Haar Like. Với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó. Bắt đầu từ vị trí trên bên trái đến vị trí dưới bên phải của ảnh, việc tính toán này đơn thuận chỉ dựa trên phép cộng số nguyên đơn giản, do đó tốc độ thực hiện rất nhanh

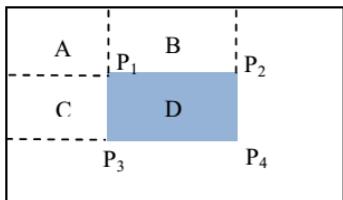


Hình 2.2: Tính giá trị ảnh tích phân tại điểm có tọa độ (x, y)

Giá trị của ảnh tích phân tại điểm P có tọa độ (x,y) được tính như sau:

$$\text{ii}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{x}' \leq \mathbf{x}, \mathbf{y}' \leq \mathbf{y}} \mathbf{i}(\mathbf{x}', \mathbf{y}')$$

Dùng integral image việc tính tổng các giá trị mức xám của một vùng ảnh bất kỳ nào đó trên ảnh thực hiện theo cách sau, ví dụ tính giá trị của vùng D như sau:  $D = A + B + C + D - (A + B) - (A + C) + A$ .



Hình 2.3: *Tính nhanh giá trị của vùng ảnh D*

### 2.3 Phương pháp AdaBoost

AdaBoost [2] là một bộ phân loại mạnh phi tuyến phức, hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để tạo nên một bộ phân loại mạnh. Bộ phân loại yếu  $h_k$  được biểu diễn như sau:

$$h_k(x) = \begin{cases} 1 & \text{nếu } p_k f_k(x) < p_k \theta_k \\ 0 & \text{nếu ngược lại} \end{cases}$$

Với  $x$  là cửa sổ con cần quét,  $h_k$  là giá trị trả về của đặc trưng Haar-like thứ  $k$ ,  $p_k$  là hệ số chuẩn hóa  $f_k$  là giá trị đặc trưng Haar-like thứ  $k$   $\theta_k$  là ngưỡng.

Công thức trên có thể được diễn giải như sau: nếu giá trị vector đặc trưng của mẫu cho bởi hàm  $f_k$  của bộ phân loại vượt qua một ngưỡng cho trước thì mẫu là object (đối tượng cần nhận dạng), ngược lại thì mẫu là background (không phải đối tượng).

### 2.4 Mô hình phân tầng cascade

Cascade of Boosted Classifiers [2] [3] là mô hình phân tầng với mỗi tầng là một mô hình AdaBoost sử dụng bộ phân lớp

yếu là cây quyết định với các đặc trưng Haar-Like.

Mô hình Cascade of Classifiers được xây dựng nhằm rút ngắn thời gian xử lý, giảm thiểu nhận dạng lầm (false alarm) cho bộ phân loại. Cascade trees gồm nhiều tầng (stage hay còn gọi là layer), mỗi tầng là một mô hình AdaBoost với bộ phân lớp yếu là các cây quyết định. Một mẫu để được phân loại là đối tượng thì nó cần phải đi qua hết tất cả các tầng.

### 2.5 Thuật toán SIFT

#### 2.5.1 Phát hiện điểm cực trị

Các điểm hấp dẫn với đặc trưng SIFT tương thích với các cực trị địa phương của bộ lọc difference-of-Gaussian (DoG) ở các tỉ lệ khác nhau. Định nghĩa không gian tỉ lệ của một hình ảnh là hàm  $L(x,y,k\sigma)$  được mô tả như sau:

$$L(x,y,k\sigma) = G(x,y,k\sigma) * I(x,y)$$

Với  $G(x,y,k\sigma)$ : biến tỉ lệ Gaussian

$I(x,y)$ : Ảnh đầu vào

$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x^2+y^2)/2\sigma^2}$$

Để phát hiện điểm Keypoint [4] [5] [6] ổn định và hiệu quả trong không gian tỉ lệ, Lowe đã đề xuất sử dụng không gian cực trị dùng các hàm Gaussian khác nhau với các hình ảnh  $D(x, y, \sigma)$ , chúng có thể được tính toán từ sự khác biệt của hai tỉ lệ lân cận cách nhau bởi một số hằng số  $k$  không đổi:

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) * I(x,y)$$

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma)$$

Hàm sai khác DoG có thể được sử dụng để tạo ra một xấp xỉ gần với đạo hàm bậc hai Laplace có kích thước chuẩn của hàm Gaussian ( $\sigma^2 \nabla^2 G$ ) do tác giả Lindeberg đề xuất năm 1994. Ông đã chỉ ra rằng việc chuẩn hóa đạo hàm bậc hai với hệ số  $\sigma^2$  là cần thiết cho bất biến đo trờ nên đúng. Cụ thể, ông đã công bố rằng các giá trị cực đại và cực tiểu của  $(\sigma^2 \nabla^2 G)$  chính là những giá trị có tính ổn định nhất (bất biến cao)

Mối quan hệ giữa D và  $\sigma^2 \nabla^2 G$  như sau:

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G$$

Từ đây, chúng ta thấy rằng  $\sigma^2 \nabla^2 G$  có thể được tính xấp xỉ để  $\partial G / \partial \sigma$  đạt sự khác biệt gần nhất về tỉ lệ tại  $k\sigma$  và  $\sigma$ :

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

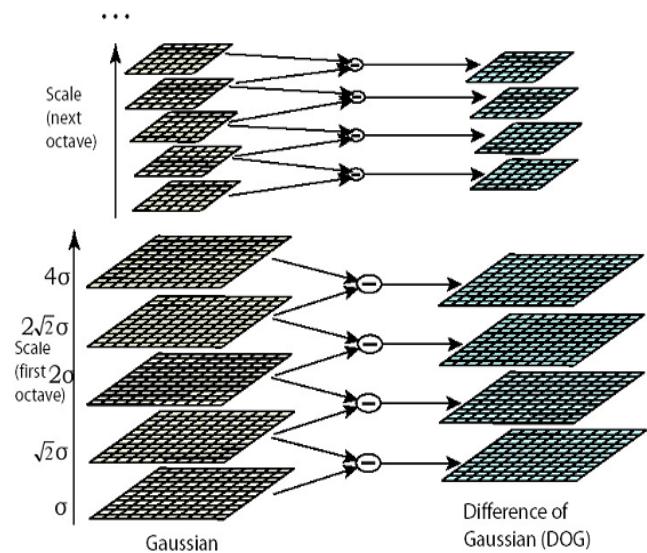
Do đó:

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G$$

Từ công thức này, ta thấy khi mà hàm sai khác DoG được tính toán tại các tham số đo lěch nhau một hằng số, thì ta có thể sử dụng DoG để xấp xỉ đạo hàm bậc hai Laplace của Gaussian. Vì hệ số  $(k-1)$  trong phương trình trên là hằng số trong mọi không gian đo nên nó sẽ không ảnh hưởng đến việc tìm các vị trí cực trị.

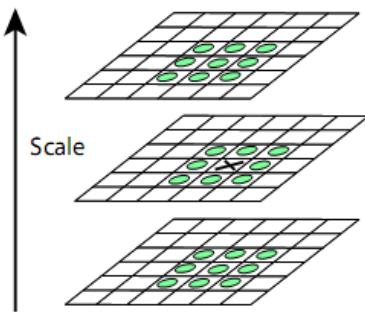
Như vậy, bước đầu tiên của giải thuật SIFT phát hiện các điểm hấp dẫn với bộ lọc Gaussian ở các tỉ lệ khác nhau và các ảnh DoG từ sự khác nhau của các ảnh kè mờ.

Các ảnh cuộn được nhóm thành các octave (mỗi octave tương ứng với giá trị gấp đôi của  $\sigma$ ). Giá trị của  $k$  được chọn sao cho số lượng ảnh mờ (blured images) cho mỗi octave là cố định. Điều này đảm bảo cho số lượng các ảnh DoG cho mỗi octave không thay đổi. Các điểm hấp dẫn được xác định là các cực đại hoặc cực tiểu của các ảnh DoG qua các tỉ lệ. Mỗi điểm ảnh trong



Hình 2.4: Biểu đồ mô phỏng việc tính toán các DoG ảnh từ các ảnh kè mờ

DoG được so sánh với 8 điểm ảnh láng giềng của nó ở cùng tỉ lệ đó và 9 láng giềng kè ở các tỉ lệ ngay trước và sau nó. Nếu điểm ảnh đó đạt giá trị cực tiểu hoặc cực đại thì sẽ được chọn làm các điểm hấp dẫn tiềm năng



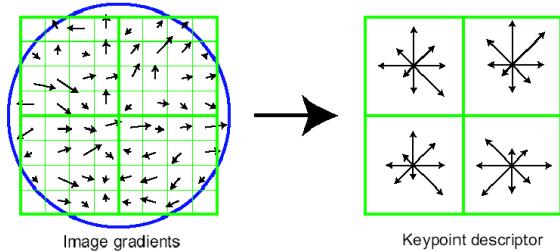
Hình 2.5: Quá trình tìm điểm cực trị trong các hàm sai khác DoG

### 2.5.2 Định vị các Keypoint

Mỗi điểm hấp dẫn tiềm năng sau khi được chọn sẽ được đánh giá xem có được giữ lại hay không. Các điểm hấp dẫn có độ tương phản thấp và một số điểm hấp dẫn độc theo các cạnh không giữ được tính ổn định khi ảnh bị nhiễu sẽ bị loại bỏ. Các điểm hấp dẫn còn lại sẽ được xác định hướng.

### 2.5.3 Mô tả các điểm hấp dẫn

Các phép xử lý trên đã thực hiện dò tìm và gán tọa độ, kích thước, và hướng cho mỗi điểm nổi bật. Các tham số đó yêu cầu một hệ thống tọa độ cục bộ 2D có thể lặp lại được để mô tả vùng ảnh cục bộ và nhờ vậy tạo ra sự bất biến đối với các tham số đó. Bước này sẽ tính toán một bộ mô tả [4] [6] cho một vùng ảnh cục bộ mà có tính đặc trưng cao (bất biến với các thay đổi khác nhau về độ sáng, thu - phóng ảnh, xoay).

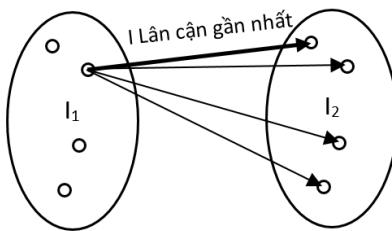


Hình 2.6 : Biểu diễn các vector đặc trưng

Ảnh trái là mô phỏng biên độ gradient và hướng tại mỗi mẫu ảnh trong một vùng lân cận với điểm keypoint. Các giá trị đó tập trung trong một cửa sổ gaussian (nằm bên trong vòng tròn). Các mẫu này sau đó được gom lại thành một lược đồ hướng mô tả văn tắt nội dung trong 4x4 vùng con như được mô tả ở bên phải với độ dài của mỗi hàng tương ứng với tổng biên độ gradient gần hướng đó bên trong một vùng

### 2.5.4 Đối sánh đặc trưng cục bộ bất biến

Để đối sánh các ảnh với nhau thì cần trích xuất tập keypoint tương ứng từ mỗi ảnh bằng các bước đã chỉ ra ở trên. Sau đó việc đối sánh sẽ thực hiện trên các tập keypoint này. Bước chính trong kĩ thuật đối sánh sẽ thực hiện tìm tập con keypoint so khớp nhau ở hai ảnh, để thực hiện việc này sẽ tìm các cặp keypoint trùng nhau lần lượt ở hai ảnh. Tập con các keypoint so khớp chính là vùng ảnh tương đồng. Việc đối sánh hai tập hợp điểm đặc trưng quy về bài toán tìm láng giềng gần nhất của mỗi điểm đặc trưng



Hình 2.7 : Đối sánh 2 ảnh quay về đối sánh 2 điểm đặc trưng

Có 2 vấn đề cần được quan tâm :

Tổ chức tập hợp điểm cho phép tìm kiếm lảng giềng một cách hiệu quả và việc đối sánh phải đạt độ chính xác nhất định. Một phương pháp được đề xuất bởi D. Mount cho phép tìm kiếm nhanh các điểm lân cận được sử dụng[4], ANN là viết tắt của Approximative Nearest Neighbour. Nó cho phép tổ chức dữ liệu dưới dạng *kd-tree*, việc tìm kiếm lảng giềng gần nhất mang tính xấp xỉ trên *kd-tree*. Cụ thể là hai điểm trong không gian đặc trưng được coi là giống nhau nếu khoảng cách Euclidean giữa hai điểm là nhỏ nhất và tỉ số giữa khoảng cách gần nhất với khoảng cách gần nhì phải nhỏ hơn 1 ngưỡng cho trước

Giả sử cặp keypoint có bộ mô tả lần lượt là:

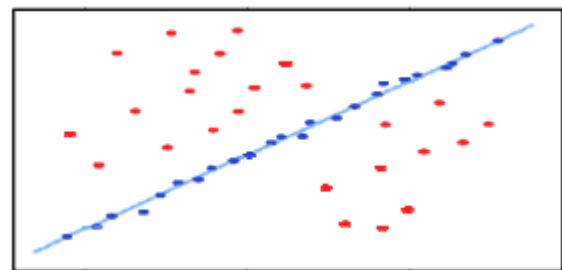
$$A = (a_1, a_2, a_3, \dots, a_{128}) \text{ và } B = (b_1, b_2, b_3, \dots, b_{128})$$

Thì khoảng cách Euclid giữa A và B được tính bằng công thức:

$$D(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

## 2.6 Thuật toán RANSAC

RANSAC [13] là một kỹ thuật ước tính mạnh mẽ cao phù hợp với bất kỳ mô hình nào bằng cách loại bỏ các giá trị ngoại biên (outliers) trong tập dữ liệu nhất định. Nó hoạt động dựa trên nguyên tắc rút gọn và có khả năng tính toán hiệu quả ngay cả khi có sự hiện diện của số lượng lớn các outlier (hơn 50%) và cũng có thể xử lý dữ liệu cấu trúc đa dạng. Một minh họa thực hiện bởi RANSAC, nơi mà các outlier (có màu đỏ) không ảnh hưởng đến kết quả cuối cùng và bị loại bỏ trong hình 2.8



Hình 2.8 : Đường phù hợp trong RANSAC

RANSAC [11] [12] 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 bảo xác suất p (thường rơi vào giá trị 0.99) của tập dữ

liệu mẫu ngẫu nhiên không chứa dữ liệu nhiễu.

Gọi  $u$  là ước lượng dữ liệu không nhiễu

$v = 1-u$  là ước lượng dữ liệu nhiễu

$m$  là số lượng dữ liệu đầu vào cần xây dựng mô hình. Khi đó ta có:

$$1-p = (1-u^m)^k$$

$k$  sẽ được tính theo công thức:

$$k = \frac{\log(1-p)}{\log(1 - (1-v)^m)}$$

Kết quả thu được sẽ là mô hình cần xây dựng phù hợp nhất với dữ liệu đầu vào, tập các dữ liệu nhiễu và tập các dữ liệu không nhiễu.

Quá trình thực hiện thuật toán RANSAC được mô tả như dưới đây:

Từ tập dữ liệu đầu vào gồm có nhiễu và không nhiễu ta chọn dữ liệu ngẫu nhiên, tối thiểu để xây dựng mô hình:

- Tiến hành xây dựng mô hình với dữ liệu đó, sau đó đặt ra một ngưỡng dùng để kiểm chứng mô hình.

- Gọi tập dữ liệu ban đầu trừ đi tập dữ liệu để xây dựng mô hình là tập dữ liệu kiểm chứng. Sau đó, tiến hành kiểm chứng mô hình đã xây dựng bằng tập dữ liệu kiểm chứng. Nếu kết quả thu được từ mô hình vượt quá ngưỡng, thì điểm đó là nhiễu, còn không đó sẽ là ngược lại.

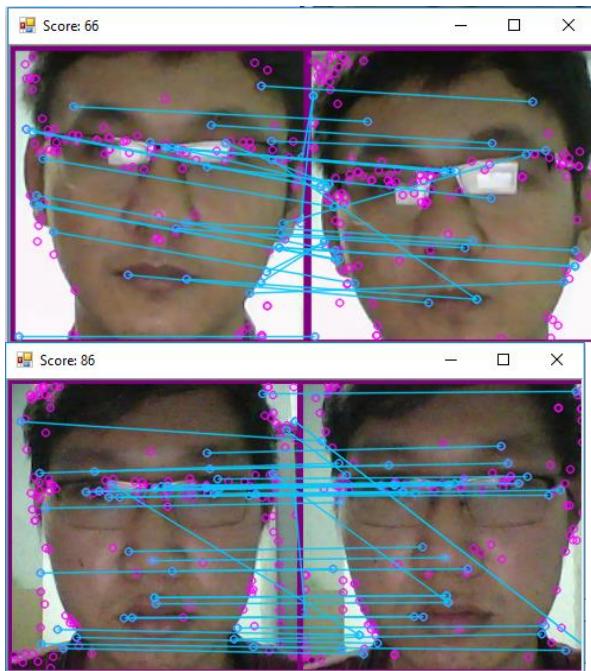
- Quá trình này sẽ được lặp đi lặp lại trong k lần. Tại mỗi vòng lặp giá trị của  $s$  sẽ được tính lại.

- Kết quả là mô hình nào có số dữ liệu không nhiễu nhiều nhất sẽ được chọn là mô hình tốt nhất.

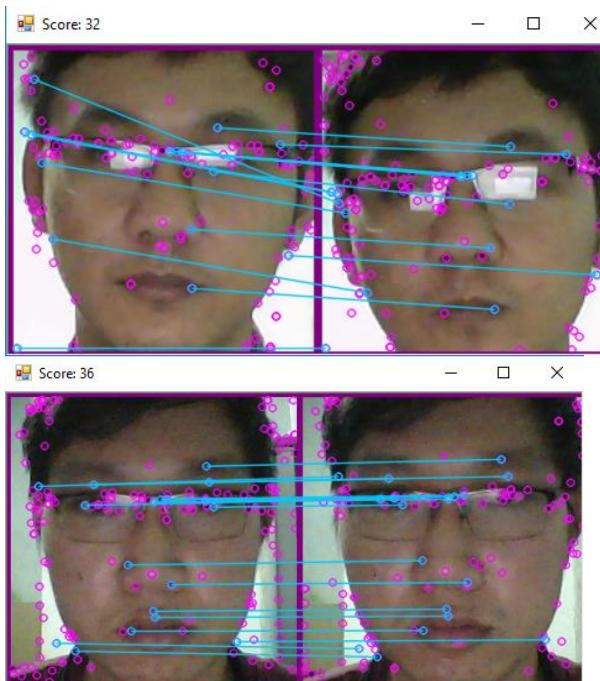
### III. KẾT QUẢ THỰC NGHIỆM

Việc đánh giá kết quả được thực hiện trên máy tính Dell Inspiron 7537 hệ điều hành Windows 10, Ram 6G và bộ xử lý CORE i5-4210U CPU @ 1.7GHz. Sử dụng Visual Studio 2013 và thư viện Emgu CV 3.1.0. Dùng các đặc trưng Haar like và thuật toán Adaboost để phát hiện được khuôn mặt trong camera thông qua tập huấn luyện

“haarcascade\_frontalface\_default.xml” được lấy trong thư viện Emgu CV. Trích đặc trưng SIFT kết hợp SANSAC trên khuôn mặt phát hiện được, ta so khớp khuôn mặt đó lần lượt với từng khuôn mặt trong cơ sở dữ liệu. Nếu số lượng so khớp (score) của khuôn mặt nào trong danh sách là lớn nhất, ta có thể kết luận đó là người trong ảnh. Thiết lập ngưỡng so khớp để tránh tình trạng một khuôn mặt bất kỳ không có trong cơ sở dữ liệu ta vẫn thu được số lượng so khớp giữa 2 ảnh nhưng không chính xác, ta nên cài đặt giá trị score để tránh tình trạng nhận danh nhầm.



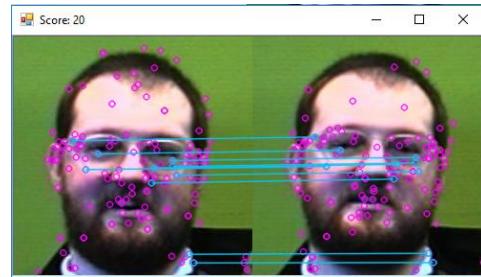
Hình 3.1 So khớp 2 ảnh dùng SIFT



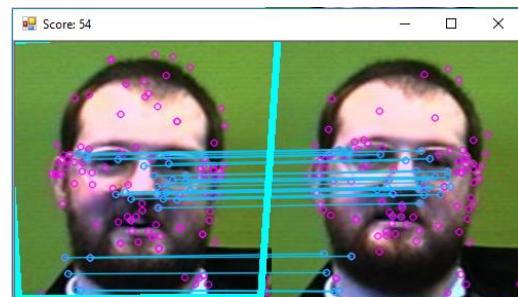
Hình 3.2 So khớp 2 ảnh dùng SIFT kết hợp RANSAC.

Chọn giá trị ngưỡng RANSAC phù hợp để tăng số lượng so khớp giữa các ảnh hoặc giảm số lượng so khớp giữa hai khuôn mặt khác nhau. Chương trình cũng đã sử dụng bộ hình ảnh khuôn mặt face94,

face95 của Tiến sĩ Libor Spacek là những tập ảnh không chịu ảnh hưởng bởi các ngoại cảnh xung quanh.



Hình 3.3 với ngưỡng RANSAC 0.5 ta có được 20 điểm so khớp



Hình 3.4 với ngưỡng RANSAC 1.5 ta thu được 54 điểm so khớp

#### IV. KẾT LUẬN

Các ảnh chứa khuôn mặt trong điều kiện ánh sáng bình thường không có tác động bởi ngoại cảnh thì số lượng so khớp của hai ảnh cao trên 50 điểm và các điểm đó đều nằm trên khuôn mặt.

Các ảnh có chứa khuôn mặt và chịu tác động bởi ngoại cảnh thì số lượng so khớp của hai ảnh cũng cao nhưng có nhiều điểm không nằm trên khuôn mặt.

Các ảnh có chứa khuôn mặt trong điều kiện ánh sáng tối thì số lượng so khớp của hai ảnh thấp và dễ bị nhầm lẫn

SIFT + RANSAC là một thuật toán rất mạnh và phức tạp trong các bài toán đối

sánh ảnh. Chúng tôi đã tìm hiểu và cài đặt thuật toán với đầy đủ các bước cơ bản của SIFT , xây dựng chương trình ứng dụng mô phỏng việc đối sánh ảnh tương tự sử dụng SIFT và dùng RANSAC để giám bớt các đối sách không đúng

#### Ưu điểm

Phát hiện được khuôn mặt qua webcam.

Tốc độ phát hiện đối tượng và nhận dạng nhanh.

Tuy nhiên, đề tài cũng còn tồn tại các hạn chế:

Độ chính xác nhận dạng phụ thuộc nhiều vào cường độ ánh sáng

Tập dữ liệu	Tổng số ảnh	Nguồn RANSAC	Tỷ lệ nhận dạng
Face94	3040	0.5	86.84%
		1.5	93.42%
Face95	1440	0.5	58.3%
		1.5	72.2%

Bảng 1 : Tỷ lệ nhận dạng trên các tập dữ liệu

#### DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Châu Ngân Khánh và Đoàn Thanh Nghị. **Nhận Dạng Mặt Người Với Giải Thuật Haar Like Feature – Cascade Of Boosted Classifiers Và Đặc Trung SIFT**. *Tạp chí khoa học trường đại học An Giang Quyển 3 (2)*, trang 15 – 24 năm 2014
- [2] Rainer Lienhart, Alexander Kuranov, Vadim Pisarevsky. **Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection**, MRL Technical Report 2002
- [3] P.Viola, M.Jones, **Rapid Object Detection using a Boosted Cascade of Simple Features**, Computer Vision and Vision and Pattern Recognition. In CVPR 2001, Proceeding of the 2001 IEEE Computer Society Conference on (Volume:1 ), Page(s):I-511 - I-518 vol.1,2001
- [4] David G. Lowe, **Distinctive Image Featuresfrom Scale-Invariant Keypoints**, Computer Science Department, University of British Columbia 2004
- [5] Nguyễn Thị Lan. Luận văn tốt nghiệp “**Truy vấn thông tin dựa trên việc đối sánh ảnh qua các đặc điểm bất biến**”
- [6] Kamarul Hawari Ghazali. **Feature Extraction technique using SIFT keypoints descriptors**. The International Conference on Electrical and Engineering and Informatics Institut technology Bandung, Indonesia, june 17-19, 2007

- [7] Nguyễn Thị Hoàn. *Phương pháp trích chọn đặc trưng ảnh trong thuật toán học máy tìm kiếm ảnh áp dụng vào bài toán tìm kiếm sản phẩm*, Đại học quốc gia Hà Nội. 2010
- [8] Faraj Alhwarin, Chao Wang, Danijela Risti -Durrant, Axel Gräser, *Improved SIFT-Features Matching for Object Recognition, Institute of Automation*, University of Bremen. 2008
- [9] Harris C. and Stephens M. , *A combined corner and edge detector*, Proceedings of the Alvey Vision Conference.1998
- [10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, *SURF: Speeded Up Robust Features*
- [11] Orhan-Sonmez RANSAC, 2006 (“<http://www.math-info.univ-paris5.fr/~lomn/Cours/CV/SeqVideo/Material/RANSAC-tutorial.pdf>”)
- [12] Wikipedia - RANSAC (<http://en.wikipedia.org/wiki/RANSAC>)
- [13] Vinay .A, Avani S Rao, Vinay S Shekhar, Akshay Kumar C, K N Balasubramanya Murthy, S Natarajan *Feature Extractionusing ORB-RANSAC for Face Recognition* , PES University and PES Institute of Technology India 2015

**Tác giả chịu trách nhiệm bài viết:**

Họ tên: Lê Nguyễn Anh Huy

Đơn vị: Trường Đại học Sư Phạm Kỹ Thuật TP.HCM

Điện thoại: 0938 269304

Email: anhhuyspkt@gmail.com

