

LỜI CẢM ƠN

Em xin bày tỏ lời cảm ơn sâu sắc nhất tới PGS.TS. Đỗ Năng Toàn, thầy đã tận tình hướng dẫn và giúp đỡ em rất nhiều trong quá trình làm tốt nghiệp đề tài tìm hiểu, nghiên cứu đề tài “Kỹ thuật Pointcare trong tự động nhận dạng vân tay” được giao để em có thể hoàn thành tốt đề tài tốt nghiệp của mình.

Em xin chân thành cảm ơn sự dạy bảo của các thầy cô giáo khoa CNTT – Trường ĐHDLHP đã trang bị cho em những kiến thức cơ bản để em có thể hoàn thành tốt đề tài tốt nghiệp.

Tuy có nhiều cố gắng trong quá trình làm đề tài nhưng em không tránh khỏi sai sót. Em rất mong thầy cô giáo chỉ dẫn, đóng góp cho em những ý kiến quý báu để giúp em hoàn thiện hơn đề tài của mình cũng như là để phát triển mở rộng đề tài sau này.

Em xin chân thành cảm ơn!

Hải Phòng ngày tháng năm

Sinh viên

Nguyễn Thị Vân.

MỤC LỤC

MỞ ĐẦU	1
CHƯƠNG I: KHÁI QUÁT VỀ XỬ LÝ ẢNH VÀ BÀI TOÁN NHẬN DẠNG	
VÂN TAY	3
1.1. Khái quát về xử lý ảnh	3
1.1.1. Xử lý ảnh là gì?	3
1.1.2. Các vấn đề cơ bản trong xử lý ảnh.....	3
1.2. Nhận dạng vân tay.....	7
1.2.1. Khoa học về dấu vân tay.	7
1.2.2. Tổng quan về vân tay, một số phương pháp phân loại vân tay.....	9
1.2.2.1. Tổng quan về vân tay.....	9
1.2.2.2. Một số phương pháp phân loại vân tay.	11
1.2.3. Các kiểu vân tay – có 17 kiểu vân tay.	14
1.3. Kết chương.	14
CHƯƠNG II: KỸ THUẬT POINTCARE TRONG TỰ ĐỘNG NHẬN DẠNG	
VÂN TAY	15
2.1. Tiền xử lý ảnh trước khi đưa vào nhận dạng vân tay.....	15
1.1.1. <i>Tính thường hóa.</i>	15
1.1.2. <i>Tính trường định hướng.</i>	17
2.2. Phương pháp trích chọn đặc trưng.....	18
2.2.1. <i>Tổng quan về phương pháp.</i>	18
2.3. Kỹ thuật Pointcare trong nhận dạng vân tay.	19
2.3.1. <i>Cơ sở lý thuyết của kỹ thuật này:</i>	19
2.4. Kết chương.	20
CHƯƠNG III: CHƯƠNG TRÌNH THỬ NGHIỆM.....	21
3.1. Giới thiệu về chương trình.....	21
3.1.1. <i>Công cụ cài đặt.</i>	21
3.1.2. <i>Tiền xử lý ảnh.</i>	21
3.1.3. Phương pháp Pointcare.....	29
3.2. Mô tả chương trình thử nghiệm và các chức năng của chương trình.	34
3.2.1. <i>Giao diện chính của chương trình.</i>	34
3.2.2. <i>Các chức năng chính của chương trình.</i>	35
3.2.3. <i>Dữ liệu phục vụ phân loại vân tay.</i>	41
3.2.4. <i>Nhận xét và đánh giá kết quả chung.</i>	41
3.3. Kết chương.	41
Kết luận	42
Tài liệu tham khảo	43

MỞ ĐẦU

Công nghệ thông tin đã xâm nhập vào hầu hết các lĩnh vực, các mặt của đời sống xã hội với những ứng dụng rộng rãi hỗ trợ cho con người trên nhiều lĩnh vực trong đó có lĩnh vực an toàn an ninh.

Những thành tựu mới của công nghệ thông tin trong những thập kỷ vừa qua đã và đang tạo nên những biến đổi to lớn đối với sự phát triển mọi mặt của đời sống xã hội. Sự phát triển của công nghệ thông tin song song với xu hướng toàn cầu hóa đã hình thành xã hội thông tin mà ở đó con người có thể vượt qua ranh giới về thời gian, không gian, khoảng cách địa lý để xích lại gần nhau, cùng nhau đóng góp và chia sẻ tri thức. Ngày nay, bất cứ một tổ chức nào, một cá nhân nào với bất kỳ một mô hình nào, quy mô nào cũng có những nhu cầu về lưu trữ và bảo mật thông tin.

- ✓ Từ trước tới nay, kỹ thuật lưu trữ thông tin cá nhân và nhận dạng cá nhân chủ yếu vẫn dựa vào một trong hai hoặc cả hai phương pháp là vật sở hữu(thẻ, con dấu, chìa khóa ...) và mã cá nhân (mật khẩu, mã số Pin...). Những phương pháp trên có hạn chế đó là có thể bị thất lạc, bị mất cắp, bị giả mạo, bị quên... Đối với vân tay của con người thì có thể khắc phục được những hạn chế trên.
- ✓ Hiện nay công nghệ này còn rất mới ở Việt Nam và nó đang được rất nhiều người quan tâm. Ngày nay trên toàn thế giới đang hướng tới trong tương lai sẽ thay thế: thẻ tín dụng, chìa khóa ...bằng nhận dạng vân tay.
- ✓ Em nhận thấy rằng hiện tại và trong tương lai thì khoa học về nhận dạng dấu vân tay sẽ ngày càng phát triển và đang rất được quan tâm. Vì thế nó sẽ đóng vai trò rất quan trọng và sẽ được áp dụng trong rất nhiều lĩnh vực của đời sống xã hội: trong ngân hàng, trong an ninh và trong giao dịch mua bán...

- ✓ Vì vậy em đã quyết định chọn đề tài bảo vệ khóa luận của mình là “ Kỹ thuật PointCare trong tự động nhận dạng vân tay”.

2.Cấu trúc của đồ án.

-Để giải quyết bài toán này thì mục tiêu và phạm vi nghiên cứu của em là em sẽ trình bày cơ bản về:

CHƯƠNG I:KHÁI QUÁT VỀ XỬ LÝ ẢNH VÀ BÀI TOÁN NHẬN DẠNG VÂN TAY

CHƯƠNG II: KỸ THUẬT POINTCARE TRONG NHẬN DẠNG VÂN TAY

CHƯƠNG III: CHƯƠNG TRÌNH THỬ NGHIỆM

PHẦN KẾT LUẬN.

TÀI LIỆU THAM KHẢO.

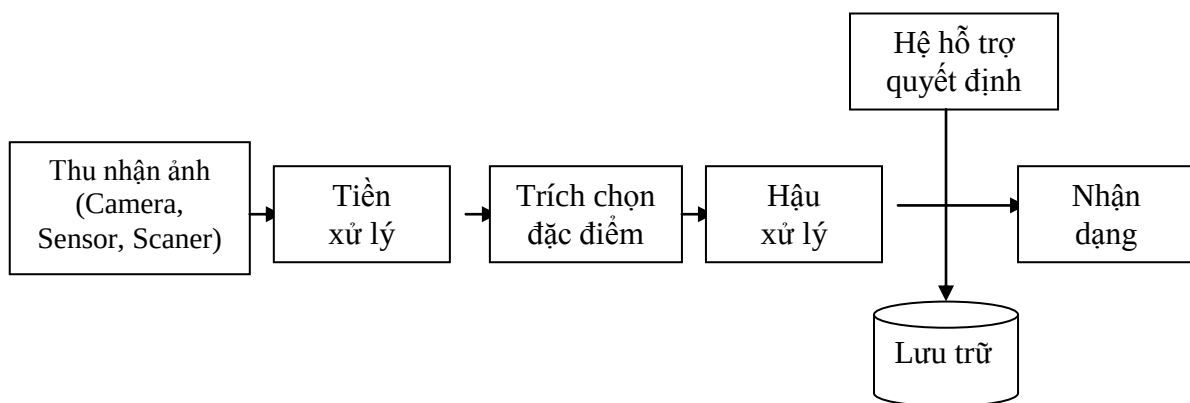
CHƯƠNG I: KHÁI QUÁT VỀ XỬ LÝ ẢNH VÀ BÀI TOÁN NHẬN DẠNG VÂN TAY

1.1. Khái quát về xử lý ảnh .

1.1.1. Xử lý ảnh là gì?

Xử lý ảnh là khoa học thao tác trên hình ảnh. Nó bao trùm một phạm vi rộng lớn các kỹ thuật hiện đang được ứng dụng rất nhiều. Ví dụ các kỹ thuật làm nổi bật hình ảnh, tăng độ sáng của một số đường nét vùng ảnh, phục hồi ảnh.

Quá trình xử lý ảnh được coi là quá trình thao tác ảnh đầu vào nhằm cho kết quả mong muốn.



Hình 1.1. Các giai đoạn chính trong quá trình xử lý ảnh.

1.1.2. Các vấn đề cơ bản trong xử lý ảnh.

▪ Pixel (Picture Element) -Phần tử ảnh.

Ảnh trong thực tế là một ảnh liên tục về không gian và về giá trị độ sáng. Để có thể xử lý ảnh bằng máy tính thì cần phải tiến hành số hóa ảnh. Trong quá trình số hóa, người ta biến đổi tín hiệu liên tục thành tín hiệu rời rạc thông qua quá trình lấy mẫu (rời rạc hóa về không gian) và lượng hóa thành giá trị mà về nguyên tắc mắt thường không phân biệt được hai điểm kề nhau. Trong quá trình này người ta sử dụng khái niệm Pixel- phần tử ảnh.

Vậy mỗi ảnh là một tập hợp các pixel, mỗi pixel gồm một cặp tọa độ(x,y) và giá trị màu. Cặp tọa độ (x,y) tạo nên độ phân giải(resolution).

Như vậy, một ảnh là một tập hợp các điểm ảnh. Khi được số hóa nó thường được biểu diễn dưới dạng mảng hai chiều $I(n,p)$: n dòng, p cột. Ta nói rằng ảnh gồm $n \times p$ Pixel. Ký hiệu $I(x,y)$ để chỉ một Pixel.

- **Gray level:** Mức xám.

Mức xám là kết quả của sự mã hóa tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số- kết quả của quá trình lượng hóa. Cách mã hoá kinh điển thường dùng là 16, 32 hay 64 mức. Mã hoá 256 mức là phổ biến nhất do lý do kĩ thuật. vì $2^8 = 256$ (0,1,2...255), nên với 256 mức, mỗi pixel sẽ được mã hoá bởi 8 bit.

- **Biểu diễn ảnh**

Trong biểu diễn ảnh người ta thường dùng các phần tử đặc trưng của ảnh là pixel. Nhìn chung có thể xem một hàm hai biến chứa các thông tin như biểu diễn một ảnh. Mỗi ảnh được biểu diễn bằng ma trận điểm ảnh cho ta một mô tả logic hay định lượng các tính chất của hàm này. Trong biểu diễn ảnh cần chú ý đến tính trung thực của ảnh hoặc các tiêu chí “thông minh ” để đo chất lượng ảnh hoặc tín hiệu hiệu quả của các kĩ thuật xử lý.

Việc xử lý ảnh số yêu cầu các ảnh phải được mẫu hoá và lượng tử hoá. Việc lượng tử hoá ảnh là chuyển đổi tín hiệu tương tự sang tín hiệu số của một số ảnh đã lấy mẫu sang một số hữu hạn mức xám. Ảnh được biểu diễn qua 2 mô hình : mô hình Raster và mô hình vector.

- **Mô hình Raster**

- + Đây là cách biểu diễn ảnh thông dụng nhất hiện nay. Ảnh được biểu diễn dưới dạng ma trận chất điểm, tùy theo yêu cầu thực tế mà mỗi điểm ảnh được biểu diễn bằng một bit hay nhiều bit.

- + Ảnh Raster thường thu nhận qua các thiết bị như: Camera, Scanner....

+ Đặc điểm của mô hình này là: thuận lợi cho việc thu nhận, hiển thị, in ấn.

-Mô hình vector

+ Trong mô hình vector người ta sử dụng hướng tới các vector của điểm ảnh lân cận để mã hoá và tái tạo điểm ảnh ban đầu.

+ Ảnh vector thường thu nhận qua các thiết bị như: Sensor (thiết bị đo), Digitalier (thiết bị số hoá) hoặc được chuyển từ Raster sang.

+ Đặc điểm của mô hình này là: tiết kiệm bộ nhớ, dễ dàng cho việc lựa chọn, hiển thị máy in, tìm kiếm, sao chép, di chuyển. Kỹ thuật này đã đáp ứng được các yêu cầu trên và nó ưu việt hơn hẳn so với mô hình Raster.

▪ Khử nhiễu.

Nhiều được xem như là sự dịch chuyển nhanh của tín hiệu trên một khoảng cách ngắn. Xem xét một cách tương đương trong miền tần số: nhiễu tương ứng với các thành phần có tần số cao trong ảnh. Do vậy người ta nghĩ ngay tới việc biến đổi có tính tới khả năng ảnh hưởng tới các phần tử lân cận bằng cách lấy tổ hợp của các điểm lân cận này, hay lọc các thành phần có tần số cao. Tùy theo cách tổ hợp điểm đang xét so với điểm lân cận mà người ta có các kỹ thuật lọc. Có hai kỹ thuật là: lọc trung bình và lọc trung vị.

a. Kỹ thuật lọc trung bình

Trong kỹ thuật này ảnh thu được là trung bình trọng số của các điểm lân cận so với mặt nạ

Giả sử điểm đang xét là $I(x,y)$

H là bộ lọc(mặt nạ) có kích thước $w * w$

Tổng các hệ số trong bộ lọc($\sum H_{ij}$) là k

Khi đó:

$$I_{TB} = \frac{1}{k} \sum_{i \in w} \sum_{j \in w} H_{ij} * I(x+i, y+j)$$

Thuật toán:

- + Cho ảnh I vào cửa sổ lọc H ($w \times w$)
- + Mọi điểm p ta di chuyển cửa sổ lọc sao cho tâm của cửa sổ trùng với điểm mà ta tác động

+ Thực hiện tính I_{TB} theo công thức trên

$$+ \text{Hiệu chỉnh } I_p \quad I_p = \begin{cases} I_{TB} & \text{nếu } |I_p - I_{TB}| \geq \theta \\ I_p & \text{nếu } |I_p - I_{TB}| \leq \theta \end{cases}$$

b. Kỹ thuật lọc trung vị

Kỹ thuật này dựa trên cơ sở kỹ thuật lọc trung bình nhưng hạn chế đi khả năng làm mờ ảnh.

- sử dụng 1 cửa sổ $w \times w$ để di chuyển khắp ảnh, tại từng vị trí

$$I_{p(0)} < I_{p(1)} < I_{p(2)} < \dots < I_{p(w \times w - 1)}$$

- I_{TV} chính là điểm chính giữa của dải.

$$- \text{Hiệu chỉnh } I_p \quad I_p = \begin{cases} I_{TV} & \text{nếu } |I_p - I_{TV}| \geq \theta \\ I_p & \text{nếu } |I_p - I_{TV}| \leq \theta \end{cases}$$

- Giả sử có một dãy gồm n phần tử $x[n]$

- Giá trị trung vị của dãy được định nghĩa như sau:

$$x_{TV} = \begin{cases} x\left[\frac{n}{2}\right] & \text{nếu } n \text{ lẻ} \\ \frac{x\left[\frac{n}{2}\right] + x\left[\frac{n}{2} - 1\right]}{2} & \text{nếu } n \text{ chẵn} \end{cases}$$

Ví dụ: cho dãy

$$x = [1 \quad 5 \quad 6 \quad 2 \quad 3] \Rightarrow x_{TV} = \frac{5}{2}$$

▪ **Nắn chỉnh biến dạng**

Nắn chỉnh biến dạng thực hiện việc biến đổi hình học giữa hai ảnh: ảnh nguồn và ảnh đích. Sự biến đổi hình học định nghĩa mối quan hệ giữa các điểm ảnh nguồn và điểm ảnh đích. Mối quan hệ này có thể được xác định bằng các hàm toán học được áp dụng trên toàn bộ ảnh hoặc chỉ trên một vùng ảnh nào đó. Trong nhiều trường hợp ngoài việc xác định các hàm toán học để nắn chỉnh ảnh, còn phải xác định thêm các đặc trưng sử dụng trong quá trình nắn chỉnh.

▪ **Nén ảnh**

Nén ảnh là quá trình làm giảm lượng thông tin dư thừa trong dữ liệu gốc do đó lượng thông tin thu được sau khi nén thường nhỏ hơn dữ liệu gốc rất nhiều.

Nén ảnh được phân làm hai loại chính:

+ Nén mất mát thông tin: dùng để nén tệp ảnh, nén âm thanh. Ví dụ: RLC, Huffman

+ Nén bảo toàn thông tin: thường sử dụng để mã hoá dữ liệu hoặc nén các tệp chạy. Ví dụ: JPG

1.2. Nhận dạng vân tay.

1.2.1. Khoa học về dấu vân tay.

Người ta biết rằng dấu vân tay của mỗi người là độc nhất. Xác suất hai cá nhân- thậm chí hai anh em sinh đôi cùng trứng có cùng một bộ dấu vân tay là 1/64 tỉ. Ngay cả các ngón trên cùng bàn tay cũng có vân khác nhau. Dấu vân tay của mỗi người là không đổi trong suốt cuộc đời. Người ta có thể phẫu thuật thay da ngón tay, nhưng chỉ sau một thời gian dấu vân tay lại trở nên như ban đầu.

Dấu vân tay được khởi tạo ở thai nhi vào giai đoạn 6 đến 7 tuần tuổi. Đầu tiên lớp đệm được hình thành. Kích cỡ và vị trí lớp đệm sẽ quyết định phần nào hình dạng kiểu vân tay. Nói chung, những lớp đệm có kích thước

nhỏ sẽ tạo nên những vân dạng vòm, những lớp đệm có kích thước lớn hơn sẽ tạo ra những vân hình móc hoặc hình tròn. Nếu lớp đệm bị lệch thì nó sẽ tạo ra vân tay không đối xứng. Dấu vân tay bắt đầu nổi rõ khi thai nhi được 3 tháng tuổi.

Khoa học về dấu vân tay được Francis Galton khởi xướng vào cuối thế kỷ XIX.

Năm 1880 Henry Fault đưa ra lý luận về số lượng vân tay RC (Ridge Count) để đánh giá mức độ phụ thuộc gen vân tay vào di truyền.

Các nhà bác học cho rằng dấu vân tay được hình thành dưới tác động của hệ thống gen di truyền mà thai nhi được thừa hưởng và những tác động của môi trường: sự cung cấp oxi, sự hình thành dây thần kinh...

Năm 1868 nhà bác học Roberts chỉ ra rằng mỗi ngón tay có một môi trường phát triển vi mô khác nhau ngoài ra ngón tay cái và trỏ còn chịu thêm một số tác động của môi trường riêng. Vì vậy vân tay trên mười đầu ngón tay của mỗi cá nhân là khác nhau.

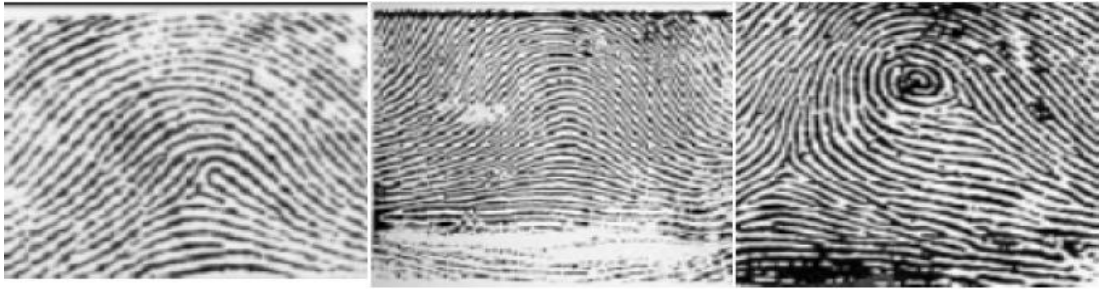
Năm 1968 nhà bác học Holt đã chứng minh được rằng có thể dự đoán tương đối chính xác tổng số lượng vân tay TRC (Total Ridge Count) và mức độ phụ thuộc của chúng vào gen di truyền của mỗi người. Vì vậy có thể coi TRC là biểu hiện phụ trợ của hệ thống gen mà con người được thừa kế.

Việc sử dụng dấu vân tay và vân chân của con người để nhận dạng đã được người Trung Quốc làm từ thế kỷ XIV.

Người ta đã bắt đầu tình cờ sử dụng dấu vân tay vào tháng 7 năm 1858. Ngài William Herschen một quan cai trị người Anh tại Indian do quá bức xúc với tính gian trá đã bắt thương gia bản xứ là Rajyadhar Konai in dấu bàn tay lên mặt sau của tờ hợp đồng.

Vào nửa thế kỷ XIX, Richard Edward Henry của Scotland Yard (cơ quan an ninh của Anh) đã phát triển phương pháp phân loại và nhận dạng vân tay. Phương pháp này được Francis Galton cải tiến vào năm 1892 và được sử dụng rộng rãi làm cơ sở thực nghiệm với độ tin cậy cao. Gần như đồng thời với

hệ thống phân loại vân tay của người Anh, Juan Vucetich đã tạo ra một hệ thống phân loại khác cho các nước đang dùng tiếng Tây Ban Nha. Hiện nay InterPol sử dụng cả hai hệ thống nêu trên. Ở Mỹ có nhiều hệ thống phân loại và xử lý thông tin vân tay. Tuy nhiên có thể phân loại vân tay theo 3 kiểu chính: xoáy tròn, móc và vòm. Ngoài ra mỗi kiểu còn được phân theo độ nghiêng: 0, 45, 90 và 135 độ.



Dấu vân tay được sử dụng trong lĩnh vực hình sự, trong việc xác định nhân thân của cá nhân khi truy cập mạng hoặc mở khóa, một số ngân hàng đã bắt đầu thanh toán thẻ ATM sử dụng máy đọc vân tay.

1.2.2. Tổng quan về vân tay, một số phương pháp phân loại vân tay.

1.2.2.1. Tổng quan về vân tay.

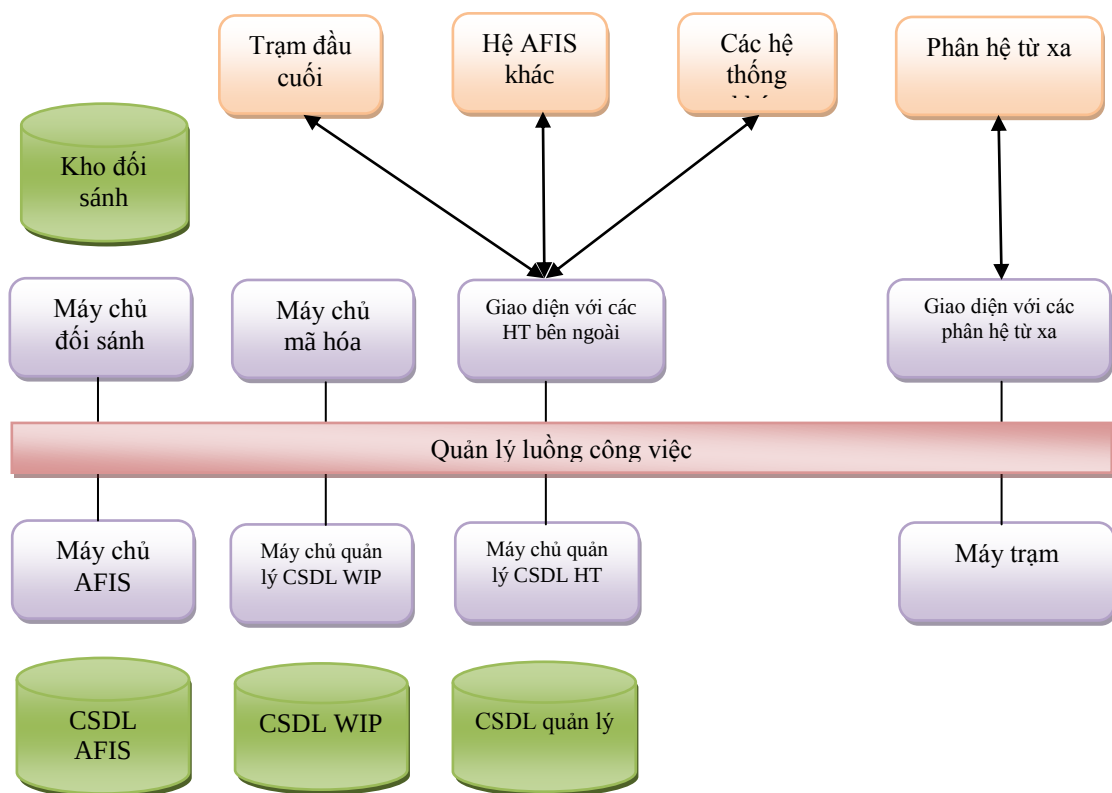
Vân tay là những đường có dạng dòng chảy ở trên ngón tay người. Nó là một tham số sinh học bất biến theo tuổi tác đặc trưng cho mỗi cơ thể. Hiện nay việc sử dụng công nghệ nhận dạng vân tay được tin cậy hơn nhiều so với các phương pháp nhận dạng thông thường khác như chữ ký, mặt người hay giọng nói.

Thông thường việc nhận dạng vân tay được thực hiện một cách thủ công bởi các giám định viên, tuy nhiên việc làm này rất đơn điệu, thiếu khoa học, mất thời gian do cơ sở dữ liệu vân tay thường khá lớn và do đó không thỏa mãn yêu cầu của các ứng dụng mới. Vì vậy hệ AFIS ra đời đã trở thành một nhu cầu cấp thiết. Hiện nay một số nước trên thế giới như Mỹ, Pháp, Nhật đã nghiên cứu thành công hệ AFIS cho công tác hình sự. Mặc dù đã có những tiến bộ đáng kể trong việc thiết kế các hệ AFIS trong hơn 30 năm qua nhưng

do một số yếu tố như thiếu các thuật toán trích chọn đặc điểm đủ tin cậy, khó khăn trong việc xác định một cách định lượng sự giống nhau giữa hai vân tay.

Các hệ AFIS hiện nay vẫn chưa đạt được tính năng mong muốn. Vì vậy nó vẫn đang được nghiên cứu rộng rãi trên thế giới.

Tổng quan những vấn đề cơ bản nhất của hệ AFIS, cấu trúc của một hệ được mô tả như sau:



Sơ đồ kiến trúc của một hệ AFIS (hệ MetaMorpho)

Vân tay thu nhận rất quan trọng trong quá trình xử lý. Khi xây dựng cơ sở dữ liệu vân tay, hệ thống trích ra đặc điểm của ảnh vân tay, đưa vào phân loại vân tay, mã hóa chúng và lưu trữ chúng vào cơ sở dữ liệu để phục vụ cho các giai đoạn xử lý sau này; còn trong quá trình nhận dạng một vân tay cho trước, các đặc điểm trích chọn được dùng cho phân loại và đối sánh. Các đặc điểm của ảnh vân tay có thể được chia làm hai loại chính:

- a. Đặc điểm tổng thể là hướng của các đường vân tay tại các vùng được sử dụng cho quá trình phân loại.
 - b. Đặc điểm cục bộ bao gồm điểm kết thúc và điểm rẽ nhánh, được sử dụng cho quá trình đối sánh. Ngoài ra còn kể đến tâm và tam phân điểm là hai loại đặc điểm được sử dụng trong quá trình phân loại.
- Ý nghĩa của việc phân loại ảnh vân tay.
 - ✓ Phân loại: Các ảnh vân tay được phân loại nhằm tăng tốc độ tìm kiếm trong cơ sở dữ liệu vân tay trong quá trình nhận dạng.
 - ✓ Tìm kiếm: Thông tin về loại của ảnh vân tay được sử dụng để thu hẹp phạm vi tìm kiếm trong cơ sở dữ liệu.
 - ✓ Đối sánh, kiểm tra: Việc đối sánh ảnh vân tay cần nhận dạng chỉ cần được tiến hành trên vân tay (có trong cơ sở dữ liệu) thuộc loại đã được xác định nhờ quá trình phân loại. Đây là giai đoạn quyết định xem hai ảnh vân tay có hoàn toàn giống nhau hay không và đưa ra kết quả nhận dạng, tức là ảnh vân tay cần nhận dạng tương ứng với vân tay của cá thể nào đã được lưu trữ trong cơ sở dữ liệu.

1.2.2.2. Một số phương pháp phân loại vân tay.

Các phương pháp phân loại vân tay đều dựa trên hai đặc điểm chung nhất của mọi vân tay, đó là tâm và tam phân điểm(hay còn được gọi là core và delta). Dựa trên các thông tin về số lượng tam phân điểm và vị trí của chúng là xác định được loại vân tay. Vân tay được phân làm các loại: Left(Quai trái), Right(Quai phải), Arch(Cung), Whorl(Xoắn), và Unkown (không biết).



Fig. 1 Left loop



Fig. 2 Right loop



Fig. 3 Arch



Fig. 4 Whorl

Trích chọn tâm (core) và tam phân điểm (delta) nếu số lượng (core, delta) là (1,1) (2,2) (1,2) Quai hay cung chồi? Xoáy hay quai kép? Quai trái hay quai phải? Cung tròn Quai trái Quai phải Cung chồi Xoáy Quai kép. Sau đây em muốn giới thiệu một số phương pháp phân loại vân tay đã được nghiên cứu và công bố, muốn lưu ý tới phương pháp trích chọn tâm và tam phân điểm được sử dụng:

- Phương pháp phân loại Henry: Đây là phương pháp phân loại cổ điển và phổ biến nhất, được sử dụng chủ yếu khi nhận dạng vân tay một cách thủ công. Các tâm và tam phân điểm được nhận biết bằng mắt thường và vân tay được phân loại dựa trên số lượng đường vân bị cắt bởi đường nối tâm và tam phân điểm.

Các phương pháp phân loại dựa trên các đặc điểm tổng thể. Việc phân loại vân tay trong phần lớn các hệ AFIS hiện nay đều dựa trên các đặc điểm tổng thể. Việc trích chọn tâm và tam phân điểm có thể được

thực hiện trực tiếp trên ảnh vân tay theo phương pháp xử lý ảnh theo từng điểm, nhưng nhược điểm của phương pháp này là tốc độ xử lý chậm. Sau khi tách hướng các vùng sẽ nhận được một ảnh định hướng đặc trưng cho vân tay.

- Phương pháp 2: Mẫu phân bố hướng chuẩn được định nghĩa là một mẫu hai chiều mô tả phân bố của các hướng lần xung quanh một điểm đặc trưng. Bằng nghiên cứu thống kê trên nhiều vân tay, các tác giả đã định nghĩa đặc trưng tâm, tam phân điểm bằng các mẫu phân bố hướng chuẩn. Việc trích chọn tâm và tam phân điểm được qui về việc tìm kiếm trên ảnh định hướng các vector phân bố hướng có dạng giống với mẫu phân bố hướng chuẩn bằng các đối sánh các mẫu phân bố hướng tại các điểm có khả năng là đặc điểm với các mẫu phân bố hướng chuẩn. Mẫu phân bố hướng chuẩn đặc trưng cho tam phân điểm.
- Phương pháp 3: Hướng của các vùng được lượng tử hóa theo 8 hướng trong khoảng từ 0^0 đến 180^0 . Các vùng đặc điểm tâm và tam phân điểm được định vị trên ảnh định hướng bằng cách kiểm tra chỉ số Pointcare Index trên một đường cong nhỏ khép kín xung quanh một điểm.

-> Sau khi nghiên cứu và phân tích ưu nhược điểm của từng phương pháp em thấy phương pháp 3 có nhiều ưu điểm nhất.

Thứ nhất : Đây là cách tiếp cận truyền thống nhất đối với bài toán phân loại vân tay. Chính vì vậy, chúng có thể tiếp nhận và đáp ứng được các yêu cầu, các quy ước của khách hàng để có thể đồng bộ hóa với dữ liệu được phân loại thủ công trước đây.

- Thứ hai : Đây cũng là một phương pháp khá đơn giản nên em có thể cài đặt vào các module.

1.2.3. Các kiểu vân tay – có 17 kiểu vân tay.

- a. Vân xoáy.
- b. Vân xoáy đồng tâm.
- c. Vân xoáy ốc.
- d. Vân xoáy đôi.
- e. Vân xoáy dài.
- f. Vân xoáy vỡ.
- g. Vân xoáy mắt tròn.
- h. Vân móc đôi.
- i. Vân móc.
- j. Vân móc ngược.
- k. Vân móc xuôi.
- l. Vân móc bẹp.
- m. Vân sóng.
- n. Vân sóng thần.
- o. Vân sóng cùn.
- p. Vân móc liên sóng.
- q. Các vân đặc biệt.

1.3. Kết chương.

Chương này đã trình bày khái quát về xử lý ảnh, khoa học về dấu vân tay, một số phương pháp phân loại vân tay và các kiểu vân tay.

CHƯƠNG II: KỸ THUẬT POINTCARE TRONG TỰ ĐỘNG NHẬN DẠNG VÂN TAY

2.1. Tiền xử lý ảnh trước khi đưa vào nhận dạng vân tay.

Các ảnh vân tay của chúng ta hầu hết là các ảnh được quét từ các chỉ bản giấy. Các chỉ bản này theo thời gian có thể không chuẩn(kích thước, màu sắc). Mặt khác chất lượng ảnh rất khác nhau. Vì vậy cần có một bước chuẩn hóa về một mức chuẩn tốt nhất để có thể khai thác các thông tin trên ảnh. Bước này gọi là thường hóa (Normalization).

Hầu hết các phương pháp phân loại vân tay đều sử dụng cách mô tả hướng của các đường vân tay, người ta thường gọi là Trường hướng “Orientation Field”. Để tính một hướng cục bộ người ta tính hướng biến thiên lớn nhất của một 17×17 . Bước này gọi là tính định hướng (Direction).

Vậy ở giai đoạn tiền xử lý ảnh có hai bước là: Thường hóa và Tính trường định hướng.

Tính thường hóa.

Phương pháp này do Hong, Wan và Jain đề xuất :

Với mỗi điểm ảnh:

$$N(i, j) = \begin{cases} M_0 + \sqrt{\frac{V_0(I(i, j) - M_i)^2}{V_i}} & \text{if } I(i, j) > M \\ M_0 - \sqrt{\frac{V_0(I(i, j) - M_i)^2}{V_i}} & \text{otherwise} \end{cases}$$

where

$$M(I) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i, j),$$

and

$$V(I) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - M(I))^2$$

Trong đó μ_0 và σ_0 là các giá trị mức xám và độ lệch chuẩn.

Như vậy phép thường hóa thực chất là chuyển kỳ vọng (giá trị trung bình) và phương sai (độ lệch chuẩn) của tập hợp các điểm trong ảnh vân tay về một kỳ vọng và phương sai chuẩn.

Trong bài toán này ta chọn $\mu_0=100$, $\sigma_0=100$.



Đây là ảnh vân tay trước khi thực hiện chức năng thường hóa.



Đây là ảnh vân tay sau khi thực hiện chức năng thường hóa.

Phép thường hóa có tác dụng làm giảm lem của những vùng có độ lem không lớn làm cho ảnh rõ hơn.

Tính trường định hướng.

Các bước để tính trường định hướng.

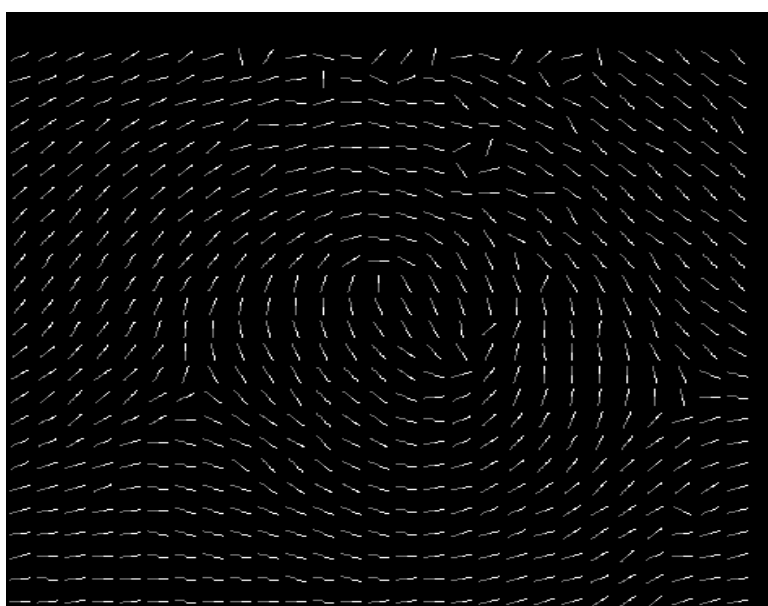
- Dùng phép lọc Sobel để tính tốc độ biến thiên tại từng điểm theo hướng x và y. Từ đó tính được vecto Gradient của từng điểm, nó thể hiện hướng có tốc độ biến thiên lớn nhất, đó chính là pháp tuyến của đường vân tay tại điểm đang xét.

$$\alpha(x, y) = \arctan\left(\frac{G_y}{G_x}\right).$$

- Tuy nhiên, để đạt được độ chính xác tốt mà độ phức tạp tính toán chấp nhận được thì phải xét đến hướng chủ đạo cho từng vùng. Theo các nghiên cứu thì người ta chọn vùng 17* 17 làm chuẩn.

$$\tilde{\theta}(x, y) = \frac{1}{2} \arctan\left(\frac{\sum_{(i,j) \in W_2} \sin 2\alpha(i, j)}{\sum_{(i,j) \in W_2} \cos 2\alpha(i, j)}\right).$$

Đây là ví dụ về tính trường định hướng.



2.2. Phương pháp trích chọn đặc trưng.

Đây là phương pháp trích chọn điểm Core và điểm Delta dựa trên các Singular Point

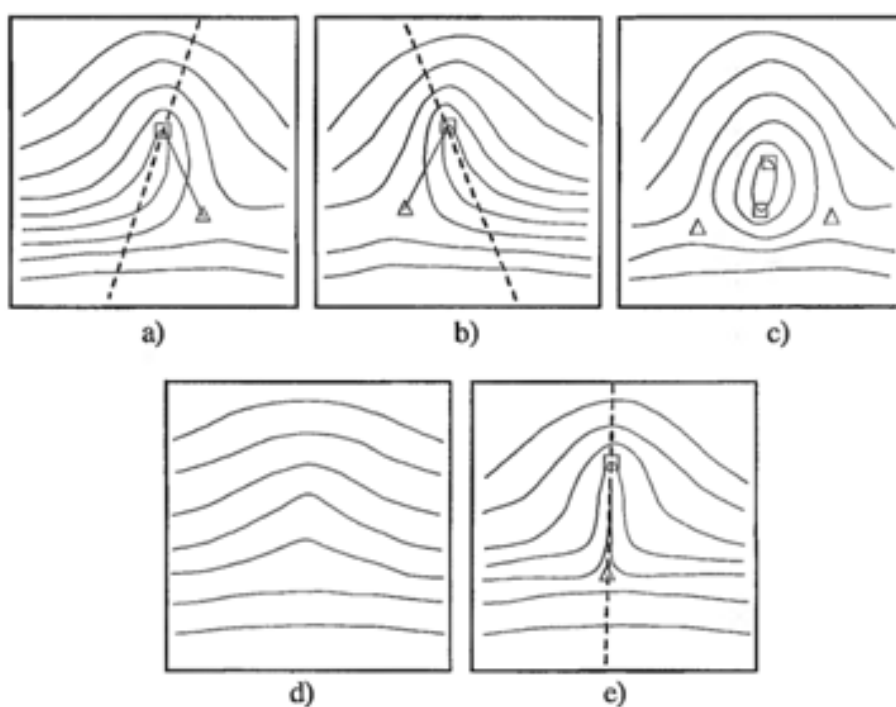
2.2.1. Tổng quan về phương pháp.

Phương pháp này dựa trên việc xác định vị trí của các Singular Point để phân loại vân tay.

Có 2 loại Singular Point là Core và Delta.

- Core là tâm của vòng xoáy vân tay.
- Delta là tam phân điểm, là nơi giao nhau của 3 hướng vân, có thể coi nó là điểm xa nhất của các vòng vân chính.

Nhờ đặc điểm này mà ta có thể phân loại vân tay:



Đây là 4 loại vân tay chính: Left, Right, Whorl, và Arch.

Như vậy để có thể áp dụng được phương pháp này ta phải tìm được Core và Delta. Trong báo cáo này em sẽ sử dụng phương pháp Pointcare để áp dụng cho bài toán này.

2.3. Kỹ thuật Pointcare trong nhận dạng vân tay.

Phương pháp này dựa trên việc tính độ vòng của một nhóm các điểm trong 1 vùng 3*3.

Ở đây ta chia thành hai nhóm vòng lên và vòng xuống để tính Pointcare Index của từng nhóm.

Tính tổng cộng 8 góc xung quanh điểm (x,y):

$$Poincare(x, y) = \frac{1}{2\pi} \sum_{k=0}^{N_p-1} \Delta(k),$$

Trong đó:

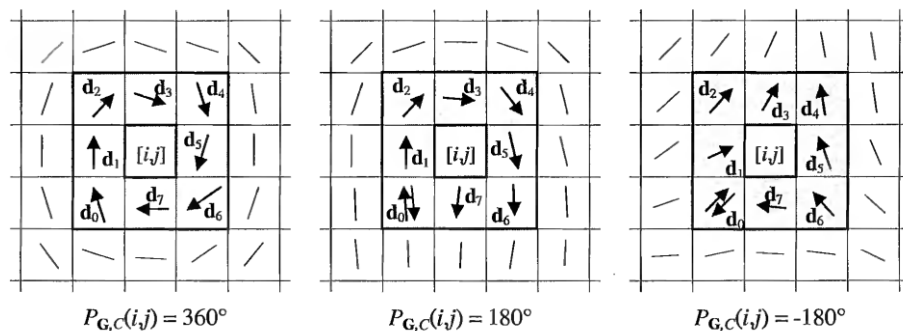
$$\Delta(k) = \begin{cases} \delta(k), & \text{if } |\delta(k)| < \frac{\pi}{2} \\ \pi + \delta(k), & \text{if } \delta(k) \leq -\frac{\pi}{2} \\ \pi - \delta(k), & \text{otherwise} \end{cases}$$

$$\delta(k) = \theta(x_{(k+1) \bmod N_p}, y_{(k+1) \bmod N_p}) - \theta(x_k, y_k)$$

$\theta(x_k, y_k)$ là góc chủ đạo tại vùng (x,y) .

2.3.1. Cơ sở lý thuyết của kỹ thuật này:

Khi tính được các Pointcare Index ở trên ta có thể xác định vùng đó có phải là Core và Delta không.



Nếu $\text{Pointcare}(x,y)=180$: vùng đang xét là Core (Tâm).

Nếu $\text{Pointcare}(x,y)=-180$: vùng đang xét là Delta (Tam phân điểm).

Trường hợp khác : không phải là Singular Point.

Ngoài ra nếu $\text{Pointcare}(x,y)=360$: vùng đó là Whorl.

Dựa trên đánh giá về vị trí và số lượng của Core và Delta ta có thể đưa ra kết luận phân loại vân tay.

2.4. Kết chương.

Ở chương này em đã trình bày giai đoạn tiền xử lý ảnh, phương pháp trích chọn đặc trưng và về kỹ thuật PointCare trong nhận dạng vân tay.

CHƯƠNG III: CHƯƠNG TRÌNH THỬ NGHIỆM

3.1. Giới thiệu về chương trình

3.1.1. Công cụ cài đặt.

- ✓ Sơ nét về Visual C++6.0

Visual C++6.0 nằm trong bộ Microsoft Visual Studio 6.0. Đây là một môi trường lập trình đa năng dành cho ngôn ngữ C/C++ và vì là một môi trường lập trình trên hệ điều hành Windows nên Visual C++ cho phép lập trình viên thực hiện rất nhiều công việc, hỗ trợ lập trình viên việc coding, thiết kế giao diện... Trong Visual C++6.0 chúng ta có thể tạo được: các ứng dụng trên Windows, Active hay thư viện liên kết động DLL... Visual C++6.0 có nhiều công cụ giúp việc thiết kế giao diện cho chương trình, kiểm lỗi và sửa lỗi. Thư viện MFC (Microsoft Foundation Class) là thư viện tập hợp các lớp hướng đối tượng đóng gói các hàm API của Windows và các kiểu dữ liệu đặc biệt. Bên cạnh sử dụng thư viện MFC trong Visual C++6.0, chúng ta có thể sử dụng các hàm C chuẩn.

- ✓ Lý do sử dụng công cụ Visual C++6.0.

Chương trình được viết bằng ngôn ngữ Visual C++ là vì tốc độ xử lý của Visual C++ nhanh, nhẹ. Ngoài ra Visual C++ còn hỗ trợ nhiều hàm chuẩn, nhiều thư viện giúp cho việc lập trình trở nên dễ dàng hơn. Tuy nhiên vì thời gian thực hiện đồ án không nhiều nên đồ án này chỉ tập trung vào việc cài đặt để minh họa hoạt động, cũng như kết quả ban đầu của kỹ thuật này.

3.1.2. Tiền xử lý ảnh.

Các ảnh vân tay của chúng ta hầu hết là các ảnh được quét từ các chỉ bản giấy. Các chỉ bản này theo thời gian có thể sẽ không chuẩn (kích thước, màu sắc) mặt khác chất lượng ảnh sẽ rất khác nhau. Vì vậy cần phải có một bước chuẩn hóa về một mức chuẩn tốt nhất để có thể khai thác các thông tin trên ảnh. Bước này gọi là bước thường hóa (Normalization). Ở trong giai

đoạn này thì nó bao gồm hai chức năng con đó là thường hóa và trường định hướng.

_Trước tiên em sẽ mô tả về chức năng thường hóa (Normalization). Chức năng này sử dụng một hàm chính đó là hàm Threshold(). Hàm này sẽ cho ta biết được giá trị mức xám trung bình của từng ảnh. Từ giá trị mức xám trung bình đó ta sẽ phân ngưỡng cho ảnh. Tại mỗi điểm ảnh:

- ❖ Nếu giá trị mức xám mà lớn hơn giá trị mức xám trung bình thì gán cho điểm đó có giá trị mức xám là 255.
- ❖ Nếu giá trị mức xám mà nhỏ hơn giá trị mức xám trung bình thì gán cho điểm đó có giá trị mức xám là 0.

```
void CTestAlgorithmDlg::Threshold(){
    m_pImageThreshold=(unsigned char *)malloc(m_lSizeOfBitmap);
    pStartWindows=7;
    qStartWindows=7;
    pEndWindows=m_iWidthCom-7;
    qEndWindows=m_iHeightCom-7;
    int m_iAverage=0;
    //Tính giá trị trung bình của mức xám
    for(unsigned long run1=0;run1<m_lSizeOfBitmap;run1++)
        m_iAverage+=*(m_pImage+run1);
    m_iAverage/=m_lSizeOfBitmap;
    int ThresholdValue=m_iAverage;
    //Phân ngưỡng
    for (unsigned long run2=0;run2<m_lSizeOfBitmap;run2++)
    {
        if (*(m_pImage+run2)<ThresholdValue)
            *(m_pImageThreshold+run2)=0;
        else
            *(m_pImageThreshold+run2)=255;
    }
}
```



```
//Đưa ra màn hình giá trị trung bình của mức xám
```

```
m_sResult.Format("%d",ThresholdValue);
```

```
UpdateData(FALSE);
```

```
}
```

Sau khi thực hiện xong hàm này, màn hình của chương trình sẽ hiển thị cho ta biết được giá trị mức xám trung bình của ảnh cần phân loại là bao nhiêu đồng thời sẽ hiển thị cho ta thấy được ảnh vân tay sau khi đã được phân ngưỡng.



Ví dụ ảnh sau khi đã được phân ngưỡng.

_ Việc tiếp theo là em sẽ mô tả về chức năng trường định hướng. Để thực hiện chức năng này thì đầu tiên em sử dụng phép lọc Sobel để tính tốc độ biến thiên tại từng điểm theo hướng x,y. Từ đó ta tính được vecto Gradient của từng điểm, nó thể hiện hướng có tốc độ biến thiên lớn nhất, đó chính là pháp tuyến của đường vân tay tại điểm đang xét.

Việc biểu diễn các ảnh vân tay thực chất là sử dụng một ma trận để biểu diễn nó nên bản chất của phép lọc Sobel sẽ được thực hiện như sau: ta chia ảnh vân tay ra thành các vùng có kích thước 3x3 (đó chính là ma trận con 3x3) sau đó ta nhân chập từng vùng (từng ma trận con) theo:

- Chiều dọc với ma trận $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$

- Chiều ngang với ma trận $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

```
void CTestAlgorithmDlg::Sobel(){

// SIZEOFMASK=3

int * Mask = (int *)malloc(SIZEOFMASK*SIZEOFMASK*sizeof(int));

m_piSobelx =(int *)malloc(m_lSizeOfBitmap*sizeof(int));

m_piSobely =(int *)malloc(m_lSizeOfBitmap*sizeof(int));

int temp=0;

int a= SIZEOFMASK/2;

//Sobelx
*(Mask+0)=1;
*(Mask+1)=2;
*(Mask+2)=1;

*(Mask+3)=0;
*(Mask+4)=0;
```

```

*(Mask+5)=0;

*(Mask+6)=-1;
*(Mask+7)=-2;
*(Mask+8)=-1;
for (unsigned int y1=a;y1<m_IImageHeight-a;y1++)

    for (unsigned int x1=a;x1<m_IImageWidth-a;x1++) {
        temp=0;

        for(int n1=-a;n1<=a;n1++)
            for (int m1=-a;m1<=a;m1++){
temp+=
GetMaskValue(Mask,SIZEOFMASK,m1,n1)*GetBufferValue(m_pcImageThreshold,x1+m1,y1+n1);
            }

        SetBufferValue(m_piSobelx,x1,y1,temp);
    }
//Sobely
*(Mask+0)=-1;
*(Mask+1)=0;
*(Mask+2)=1;
*(Mask+3)=-2;
*(Mask+4)=0;
*(Mask+5)=2;
*(Mask+6)=-1;
*(Mask+7)=0;
*(Mask+8)=1;
for (unsigned int y2=a;y2<m_IImageHeight-a;y2++)

    for (unsigned int x2=a;x2<m_IImageWidth-a;x2++) {
        temp=0;

        for(int n2=-a;n2<=a;n2++)

```

```

                                for (int m2=-a;m2<=a;m2++){
temp+=
GetMaskValue(Mask,SIZEOFMASK,m2,n2)*GetBufferValue(m_pcImageThreshold,x2+m2,y2+n2);
                                }

                                SetBufferValue(m_piSobely,x2,y2,temp);

                                }

                                free(Mask);

                                m_sResult="Sobelx,Sobely are created.";

                                UpdateData(FALSE);
}

```

Sau khi thực hiện phép lọc Sobel xong thì ta xác định hướng của các điểm bằng hàm Direction().

```

void CTestAlgorithmDlg::Direction()
{

                                int a=SIZEOFCOMPONENT/2;

                                int p,q,x,y;
                                int dem=0;

                                m_pdAtan= (double *)malloc(m_lSizeOfBitmap*sizeof(double));
                                memset(m_pdAtan,0,m_lSizeOfBitmap*sizeof(double));

                                m_pdOrientation= (double *)malloc(m_iSizeOfComponent*sizeof(double));
                                memset(m_pdOrientation,0,m_iSizeOfComponent*sizeof(double));

```

```

        m_pdirDirection = (Directional
*)malloc(m_iSizeOfComponent*sizeof(Directional));

double temp,temp1,temp2;
Directional dtemp;
for (unsigned long run=0;run<m_lSizeOfBitmap;run++)
{
    if ((* (m_piSobelx+run)==0)&&(* (m_piSobely+run)==0))
        *(m_pdAtan+run)=OVERPOINT;
    else
        *(m_pdAtan+run) = atan2(* (m_piSobely+run),* (m_piSobelx+run));
}
for (int run1=0;run1<m_iSizeOfComponent;run1++)
{
    p=run1%m_iWidthCom;
    q=run1/m_iWidthCom;
    x= a+ SIZEOFCOMPONENT*p;
    y= a+SIZEOFCOMPONENT*q;
    temp1=0;
    temp2=0;
    int dem=0;
    for(int n=-a;n<=a;n++)

        for (int m=-a;m<=a;m++)
        {
            if (GetBufferValueAll(m_pdAtan,x+n,y+m)!=OVERPOINT)
            {
                dem++;

                temp1+=sin(2*GetBufferValueAll(m_pdAtan,x+n,y+m));

                temp2+=cos(2*GetBufferValueAll(m_pdAtan,x+n,y+m));
            }
        }
}

```

```

if(dem==0)
{
    temp=OVERPOINT;
    dtemp.co=OVERPOINT;
    dtemp.si=OVERPOINT;
}
else
{
    temp=0.5*atan2(temp1,temp2);
    dtemp.co=R*cos(temp);
    dtemp.si=R*sin(temp);
    temp=temp*180/PI;

}

*(m_pdOrientation+run1)=temp;
*(m_pdirDirection+run1)=dtemp;

}

```

```

m_pcDirection= (unsigned char *)malloc(m_lSizeOfBitmap*sizeof(unsigned
char));

```

```

for (unsigned long run2=0;run2<m_lSizeOfBitmap;run2++)
{
    *(m_pcDirection+run2)=0;
}

```

```

char Deltax=0,Deltay=0;

```

```

for (int run3=0;run3<m_iSizeOfComponent;run3++)
{
    dtemp=*(m_pdirDirection+run3);
    double dAngle=*(m_pdOrientation+run3);

```

```

p=run3%m_iWidthCom;
q=run3/m_iWidthCom;
x= a+ SIZEOFCOMPONENT*p;
y= a+SIZEOFCOMPONENT*q;
if ((dtemp.si!=OVERPOINT)&&(dtemp.co!=OVERPOINT)){
    dem++;
    for (int r=0;r<=R;r++){
        Deltax=Round(r*dtemp.co/R);
        Deltay=Round(r*dtemp.si/R);
        SetBufferValue(m_pcDirection,x+Deltax,y+Deltay,255);
        SetBufferValue(m_pcDirection,x-Deltax,y-Deltay,255/2);
    }
}
}
}
}

```

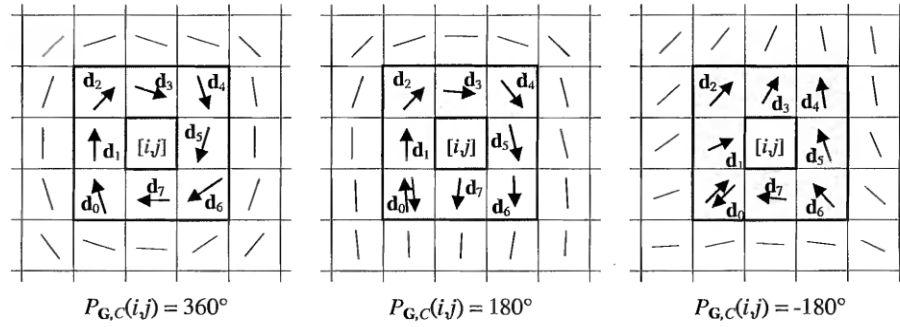
3.1.3. Phương pháp Pointcare

Phương pháp Pointcare là một trong những phương pháp trích chọn đặc trưng, phương pháp này dựa trên việc tính độ vòng của một nhóm các điểm. Từ đó trích chọn được hai điểm đó là Core và Delta.

- Core là tâm của vòng xoáy vân tay.

- Delta là tam phân điểm, là nơi giao nhau của 3 hướng vân, có thể coi nó là điểm xa nhất của các vòng vân chính.

Ở mục 3.1.2 em đã mô tả chức năng tiền xử lý ảnh. Sau khi chương trình thực hiện xong chức năng tiền xử lý ảnh thì ta thu được một ảnh vân tay bao gồm các hướng vân của các điểm. Dựa vào độ vòng của một nhóm các điểm trong một vùng 3×3 ta xác định được điểm Core và Delta.



Dưới đây là đoạn code mô tả kỹ thuật trích chọn điểm Core và Delta:

```

for (int run2=0;run2<dem;run2++){
    m_dPIndexCore=0;
    m_dPIndexDelta=0;
    temp=0;
    int fault;

    if (TRUE){
        int p=PointArray[run2]/256;
        int q=PointArray[run2]%256;

        if ((p<4)||((p>m_iWidthCom-4))||((q<4))||((q>m_iHeightCom))) continue;
        int x= a+ SIZEOFCOMPONENT*p;
        int y= a+SIZEOFCOMPONENT*q;

        if (PointLeft[run2]) {
            //Tim Core
            fault=0;

            for (int i1=0;i1<=5;i1++) {
                temp= GetMaskValue(m_pdOrientation,i1,p,q)-
                GetMaskValue(m_pdOrientation,i1+1,p,q);
            }
        }
    }
}

```



```

        if ((temp<CANTREN)&&(temp>-CANDUOI))
        m_dPIndexCore+=temp;
        else fault++;
    }

    if
    ((m_dPIndexCore>CHAPNHANDUOI)&&(m_dPIndexCore<CHAPNHANTREN
    )&& (fault<=1)&&(p>5)&&(p<m_iWidthCom-5)&&(q<m_iHeightCom-7)){
        CoreLeft[SumCore1]=PointLeft[run2];
        Core[SumCore1++]=p*256+q;
    }

    //Tim Delta
    fault=0;
    temp =GetMaskValue(m_pdOrientation,4,p,q)-GetMaskValue(m_pdOrientation,5,p,q);

    if ((temp<=CANDUOI)&&(temp>=-CANTREN)) m_dPIndexDelta+=temp;
    else fault++;
    temp =GetMaskValue(m_pdOrientation,5,p,q)-
    GetMaskValue(m_pdOrientation,6,p,q);

    if ((temp<=CANDUOI)&&(temp>=-CANTREN)) m_dPIndexDelta+=temp;
    else fault++;
    temp =GetMaskValue(m_pdOrientation,6,p,q)-
    GetMaskValue(m_pdOrientation,7,p,q);

    if ((temp<=CANDUOI)&&(temp>=-CANTREN)) m_dPIndexDelta+=temp;
    else fault++;
    temp =GetMaskValue(m_pdOrientation,7,p,q)-
    GetMaskValue(m_pdOrientation,0,p,q);

```

```

        if ((temp<=CANDUOI)&&(temp>=-CANTREN)) m_dPIndexDelta+=temp;
        else fault++;
        temp =GetMaskValue(m_pdOrientation,0,p,q)-
GetMaskValue(m_pdOrientation,1,p,q);

        if ((temp<=CANDUOI)&&(temp>=-CANTREN)) m_dPIndexDelta+=temp;
        else fault++;
        temp =GetMaskValue(m_pdOrientation,1,p,q)-
GetMaskValue(m_pdOrientation,2,p,q);
        if ((temp<=CANDUOI)&&(temp>=-CANTREN)) m_dPIndexDelta+=temp;
        else fault++;

if ((m_dPIndexDelta>-CHAPNHANTREN)&&(m_dPIndexDelta<-CHAPNHANDUOI)&&(fault<=1)){
    Delta[SumDelta++]=p*256+q;

    if (Core[SumCore1-1]==p*256+q) {
        Core[SumCore1-1]=0;
        SumCore1--;
    }
}
else{
//Tim Core
temp= GetMaskValue(m_pdOrientation,4,p,q)-GetMaskValue(m_pdOrientation,5,p,q);
    if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
    else fault++;
    temp= GetMaskValue(m_pdOrientation,5,p,q)-
GetMaskValue(m_pdOrientation,6,p,q);

    if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
    else fault++;
    temp= GetMaskValue(m_pdOrientation,6,p,q)-
GetMaskValue(m_pdOrientation,7,p,q);

```

```

if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
else fault++;
temp= GetMaskValue(m_pdOrientation,7,p,q)-
GetMaskValue(m_pdOrientation,0,p,q);

```

```

if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
else fault++;
temp= GetMaskValue(m_pdOrientation,0,p,q)-
GetMaskValue(m_pdOrientation,1,p,q);
if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
else fault++;
temp= GetMaskValue(m_pdOrientation,1,p,q)-
GetMaskValue(m_pdOrientation,2,p,q);

```

```

if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
else fault++;
    if (GetMaskValue(m_pdOrientation,3,p,q)>0) {
        temp= GetMaskValue(m_pdOrientation,2,p,q)-(GetMaskValue(m_pdOrientation,3,p,q)-
180);

```

```

if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
else fault++;
    }
    else{
        temp= GetMaskValue(m_pdOrientation,2,p,q)-GetMaskValue(m_pdOrientation,3,p,q);

if ((temp<CANTREN)&&(temp>-CANDUOI)) m_dPIndexCore+=temp;
else fault++;
    }

```

```

        if
        ((m_dPIndexCore>CHAPNHANDUOI)&&(m_dPIndexCore<CHAPNHANTREN))
        {
            CoreLeft[SumCore1]=PointLeft[run2];
            Core[SumCore1++]=p*256+q;
        }
    }
}

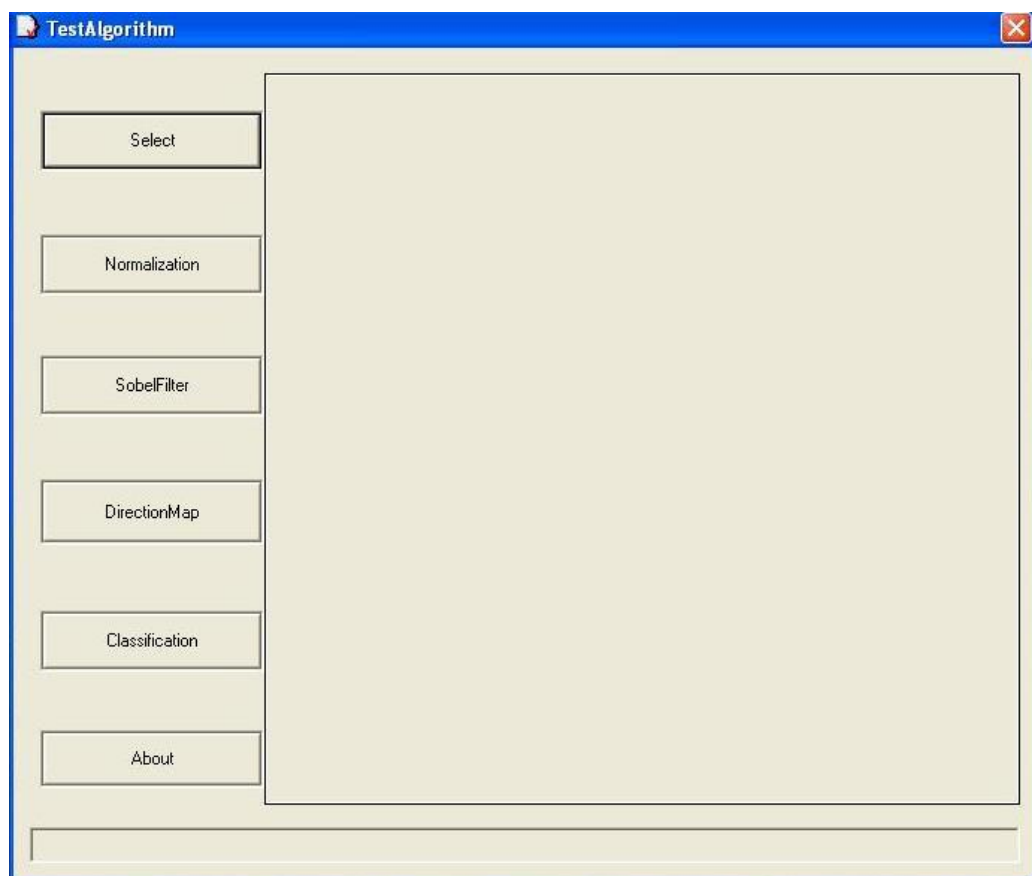
```

Vậy mục này em đã trình bày được việc thiết kế chương trình và cài đặt demo cho chương trình nhận dạng vân tay sử dụng kỹ thuật Pointcare.

Mục tiếp theo em sẽ trình bày về giao diện và cách sử dụng chương trình.

3.2. Mô tả chương trình thử nghiệm và các chức năng của chương trình.

3.2.1. Giao diện chính của chương trình.



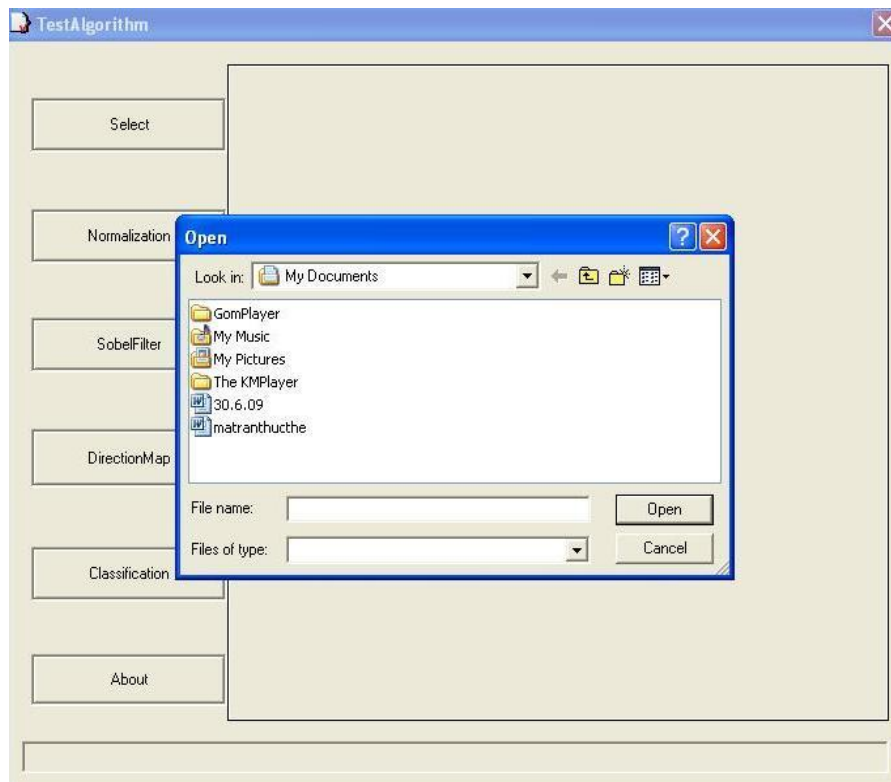
Giao diện của chương trình gồm:

- ✓ Các nút button: Select, Normalization, Sobelilter, DirectionMap, Classification, About.
- ✓ Bên phải các nút button là một Edit box để hiển thị ảnh trong quá trình thực hiện các chức năng của chương trình
- ✓ Edit box ở phần cuối của màn hình hiển thị dùng để hiển thị các đoạn text.

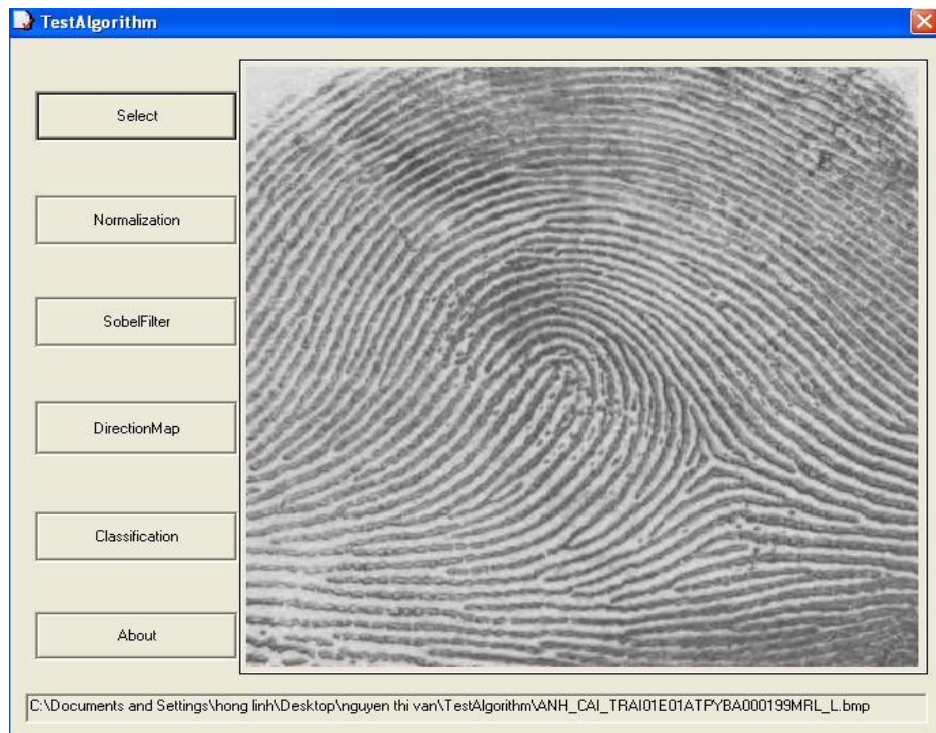
3.2.2. Các chức năng chính của chương trình.

- ✓ Chức năng lựa chọn ảnh đưa vào phân loại.

Kích chuột trái vào nút button *Select* sẽ có giao diện như sau:



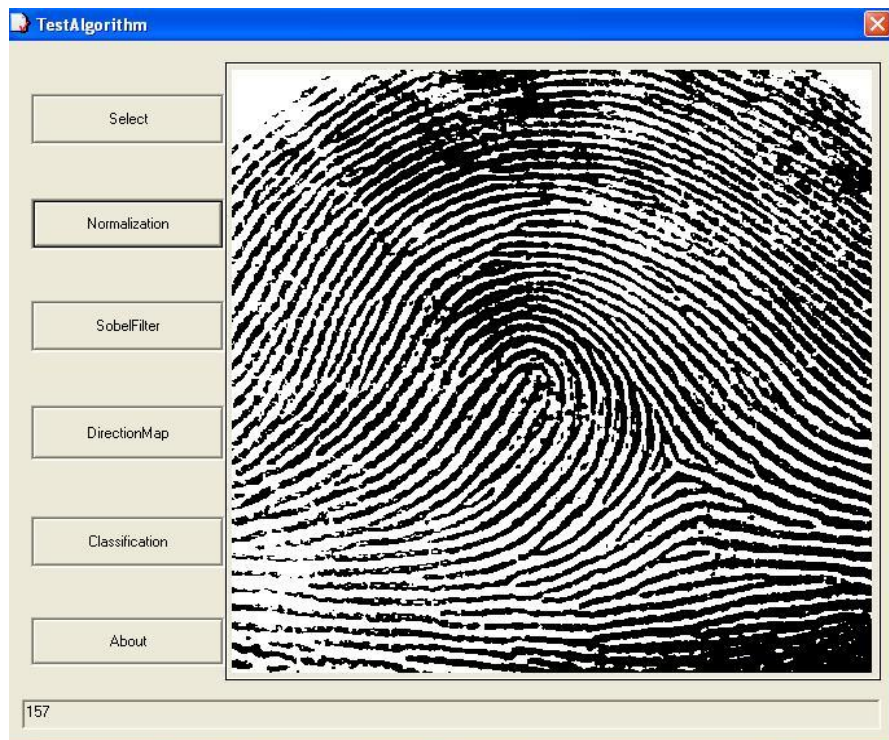
Chương trình sẽ mở ra một cửa sổ để chúng ta lựa chọn ảnh vân tay sẽ đưa vào phân loại. Sau khi chọn ảnh vân tay định đưa vào phân loại ta kích chuột vào button OPEN sau đó sẽ xuất hiện một cửa sổ. Màn hình sẽ hiện lên ảnh vân tay mà chúng ta vừa lựa chọn, đồng thời Editbox ở bên dưới sẽ hiển thị cho ta thấy được đường dẫn tới ảnh vừa chọn.



Sau khi lựa chọn ảnh đưa vào phân loại xong, tiếp đến sẽ là chức năng thường hóa.

✓ Chức năng thường hóa

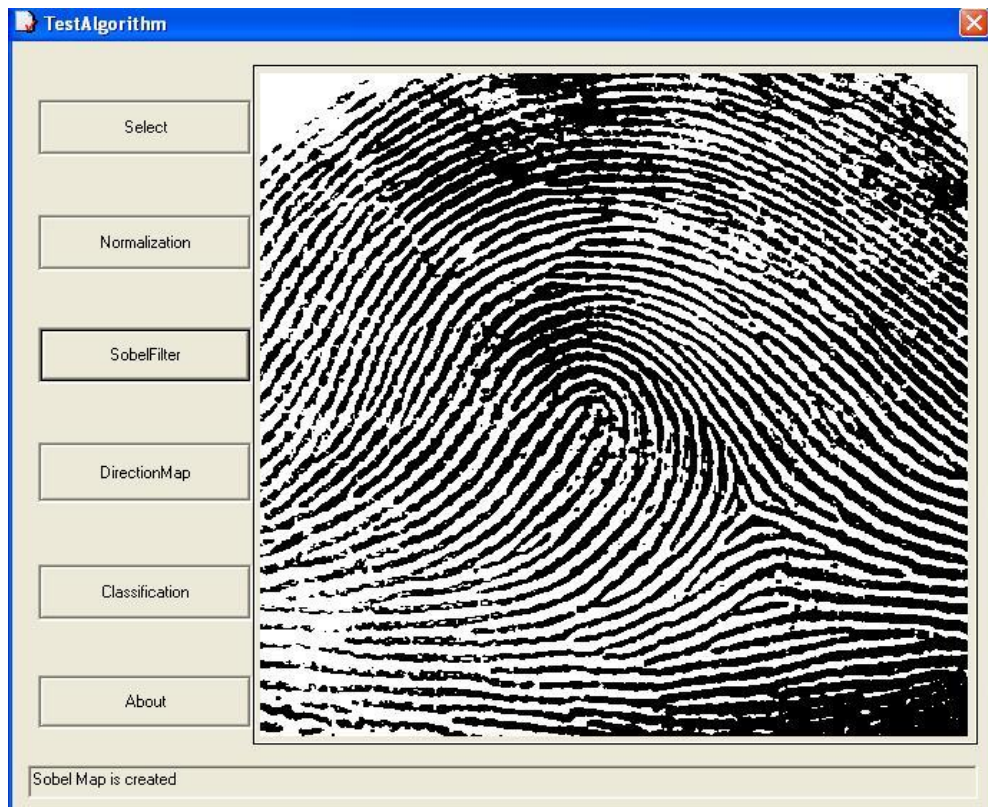
Kích chuột trái vào button *Normalization*, chương trình sẽ hiển thị cho ta mức xám trung bình của ảnh ở Editbox ở phía bên dưới màn hình giao diện. Đồng thời chương trình sẽ hiển thị cho ta giá trị mức xám của ảnh sau khi xử lý chức năng *Normalization*. Khi đó giao diện của chương trình sẽ như sau:



Sau khi thực hiện xong chức năng thường hóa ta sẽ đến một bước tương đối quan trọng đó là xác định trường định hướng của ảnh vân tay. Đó là chức năng lọc Sobel

✓ Chức năng lọc Sobel.

Để thực hiện chức năng lọc Sobel, ta kích chuột trái vào button *SobelFilter*. Sau khi thực hiện chức năng lọc Sobel, giao diện của chương trình không có gì thay đổi nhiều. Chương trình vẫn hiển thị cho ta ảnh vân tay sau khi thực hiện chức năng thường hóa. Ngoài ra ở hộp Edit box phía bên dưới sẽ thông báo cho chúng ta là chức năng lọc Sobel đã thành công.

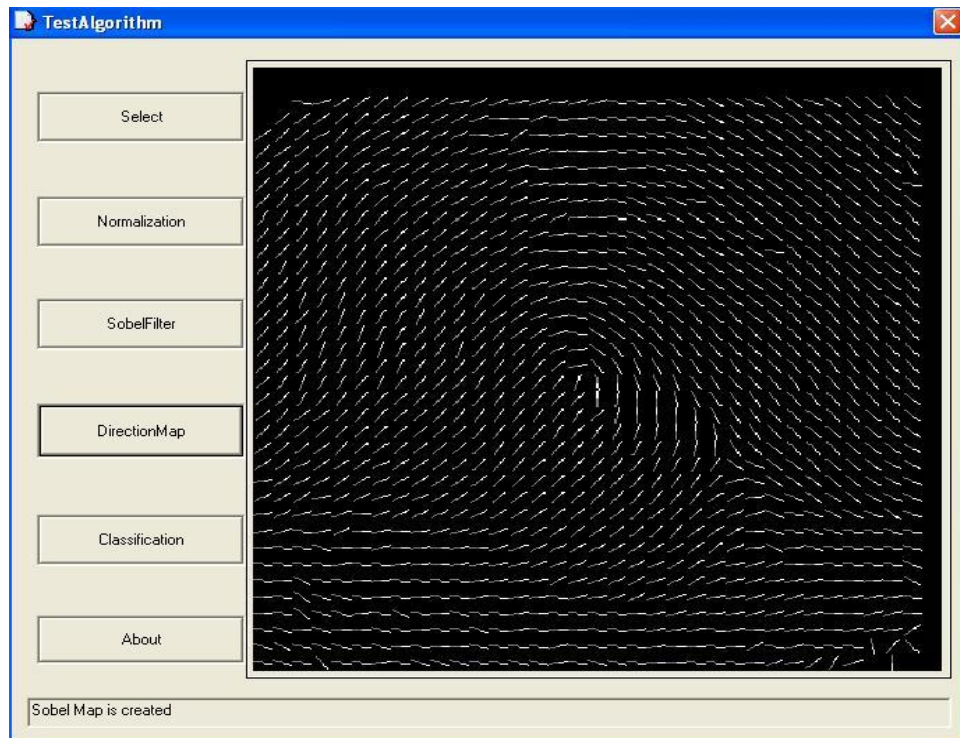


✓ Chức năng trường định hướng.

Sau khi thực hiện chức năng lọc Sobel, ta có thể đi đến tính trường định hướng cho ảnh vân tay mà chúng ta cần phân loại

Kích chuột phải vào button *DirectionMap* để thực hiện chức năng này. Sau khi thực hiện chức năng tính trường định hướng, chương trình sẽ hiển thị cho ta một ảnh vân tay nhưng không phải là ảnh vân tay sau khi được phân ngưỡng mà là ảnh vân tay sau khi đã tính trường định hướng.

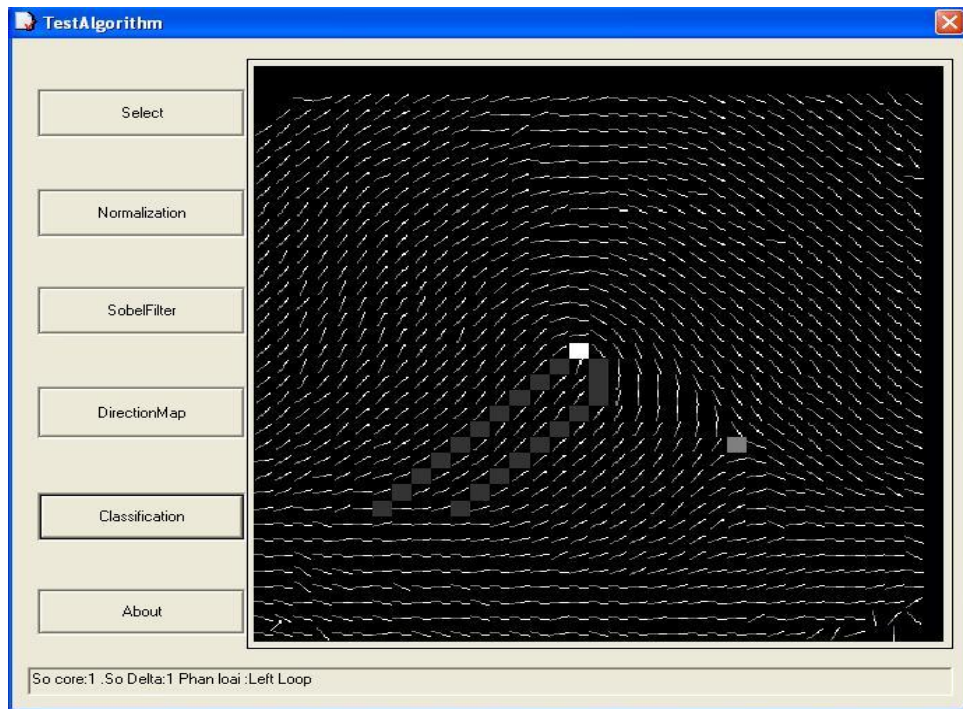
Sau đây là ảnh vân tay sau khi thực hiện chức năng trường định hướng.



✓ Chức năng phân loại vân tay.

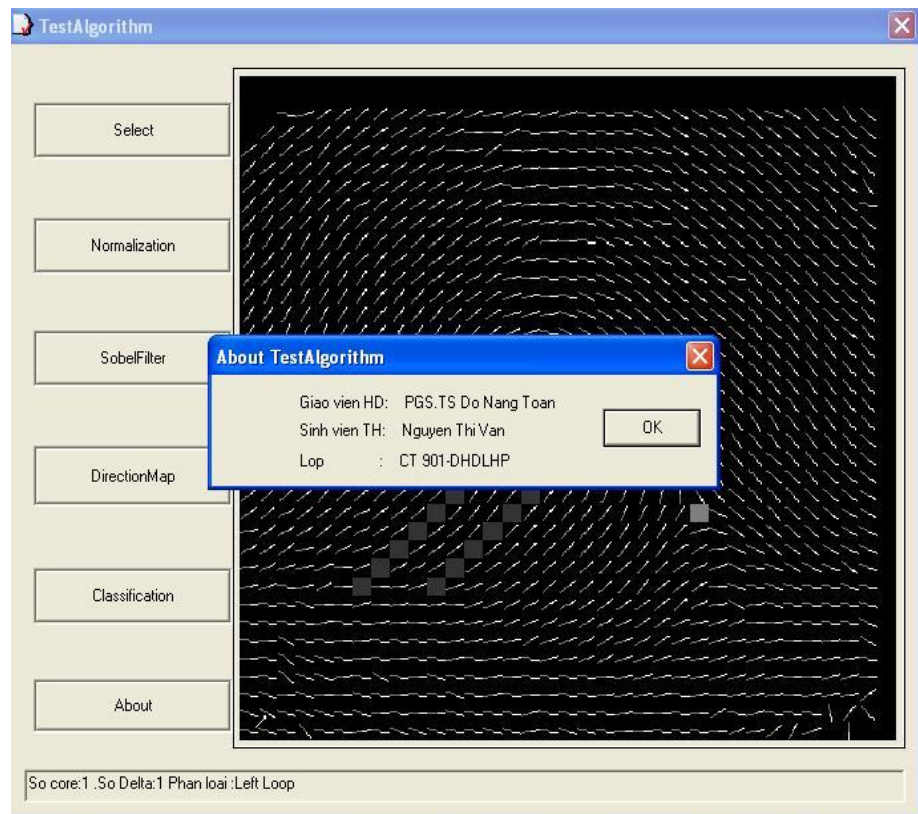
Sau khi thực hiện chức năng tiền xử lý ảnh thì ta đi đến chức năng cuối cùng đó là chức năng phân loại vân tay. Chức năng này sẽ thực hiện phương pháp PointCare như đã trình bày ở trên.

Để thực hiện chức năng phân loại vân tay ta kích chuột trái vào nút button *Classification* của chương trình. Sau khi thực hiện chức năng phân loại vân tay, chương trình sẽ hiển thị cho ta kết quả ở Edit box bên dưới của chương trình. Chương trình sẽ cho ta biết được số điểm Core và Delta và ảnh vân tay ta đưa vào phân loại thuộc loại nào. Ngoài ra chương trình còn hiển thị cho chúng ta ảnh vân tay sau khi vẽ thêm đường giả vân đồng thời sẽ tô sáng các điểm Core và Delta. Giao diện của chương trình sẽ như sau:



✓ Thông tin về tác giả.

Trong phần giao diện của chương trình em thiết kế thêm nút About. Chức năng của nút button *About* này là khi kích chuột vào nút About thì giao diện của chương trình sẽ hiển thị cho ta một Dialog box giới thiệu thông tin về sinh viên thực hiện và giáo viên hướng dẫn.



3.2.3. Dữ liệu phục vụ phân loại vân tay.

Dữ liệu phục vụ cho chương trình thử nghiệm nhận dạng vân tay bao gồm 200 ảnh vân tay với định dạng file ảnh là bitmap, các loại có kích thước từ 450x 450 đến 500x 500. Các ảnh vân tay bao gồm đủ tất cả các loại như: quai trái, quai phải, cung, xoáy ốc, ảnh mờ, ảnh nhiễu...

3.2.4. Nhận xét và đánh giá kết quả chung.

- Một số ưu điểm của chương trình.
 - _ Chương trình đã thể hiện cho chúng ta thấy được tầm quan trọng của khâu tiền xử lý ảnh đối với việc phân loại vân tay.
 - _ Chương trình đã thể hiện cho chúng ta thấy được các ưu nhược điểm của phương pháp Pointcare trong bài toán nhận dạng vân tay.
- Một số hạn chế của chương trình.
 - _ Qua kết quả chạy thử nghiệm chương trình ta thấy đôi lúc chương trình nhận dạng vẫn cho kết quả sai – xác định sai các kiểu vân.

3.3. Kết chương.

Đây là chương cuối cùng của luận án này. Nội dung chủ yếu của chương là mô tả một số chức năng chính, một số hàm chính sử dụng trong chương trình và đồng thời mô tả chương trình, cách chạy của chương trình thử nghiệm. Trong chương này đã nêu bật được phương pháp chính cũng như cách thiết phương pháp này trong chương trình. Đồng thời chương này cũng đã nêu nên được nhận xét đánh giá chung của chương trình cũng như là ưu nhược điểm của nó.

Kết luận

1. Nhắc lại sự cần thiết của đề án.

Như vậy luận văn đã trình bày một số khái niệm về khoa học dấu vân tay, kỹ thuật tiền xử lý ảnh và đặc biệt là kỹ thuật PointCare trong nhận dạng vân tay. Chương trình thử nghiệm cũng đạt được mục tiêu, yêu cầu đề ra.

2. Các việc đã làm được.

Trong quá trình thực hiện đề tài này em đã làm được:

- Nghiên cứu và hiểu được cơ bản về lĩnh vực khoa học dấu vân tay.
- Tìm hiểu được một số phương pháp tiền xử lý ảnh trước khi nhận dạng vân tay.
- Đọc các tài liệu và các bài báo trên các trang web để hiểu rõ hơn về kỹ thuật PointCare trong bài toán nhận dạng vân tay.
- Nghiên cứu được các khâu căn bản, bản chất của phương pháp PointCare để áp dụng vào cài đặt giải quyết bài toán nhận dạng vân tay.
- Cài đặt chương trình thực hiện giải quyết bài toán nhận dạng vân tay tự động áp dụng phương pháp PointCare.

3. Các việc có thể làm trong tương lai.

- Trong thời gian tới em sẽ:

_Tham khảo ý kiến góp ý của giáo viên hướng dẫn và các thầy cô giáo trong khoa để đưa ra những ý tưởng làm tăng độ chính xác cho chương trình.

_Em sẽ cố gắng tìm hiểu, nghiên cứu để có thể tìm các kỹ thuật khác phù hợp để có thể kết hợp với kỹ thuật PointCare trong bài toán nhận dạng vân tay để khắc phục những nhược điểm mà kỹ thuật Pointcare vẫn chưa đạt được.

Do luận văn được thực hiện trong thời gian tương đối ngắn nên vẫn còn một số hạn chế chưa thể đáp ứng được hoàn toàn yêu cầu của một module hoàn chỉnh và không tránh khỏi được những hạn chế và sai sót. Em rất mong được sự nhận xét, góp ý trao đổi của thầy cô, bạn bè để luận văn được hoàn thiện hơn, đáp ứng được yêu cầu của thực tiễn.

Tài liệu tham khảo

<http://www.cs.bris.ac.uk/home/burghard>

<http://tams-www.informatik.uni-hamburg.de>

<http://www.opengl.org/resources/code/samples/sig99/advanced99/notes/node258.html>

<http://www.tuvanvala.com>

<http://bias.csr.unibo.it/maltoni/handbook>