ĐẠI HỌC QUỐC GIA TP.HCM

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

---

**NGUYỄN TRỌNG TRÍ**

# THIẾT KẾ HỆ THỐNG NHẬN DẠNG CỬ CHỈ TAY SỬ DỤNG CẢM BIẾN KINECT

Chuyên ngành: Kỹ Thuật Điện Tử

Mã số: 60 52 70

**TÓM TẮT LUẬN VĂN THẠC SĨ**

TP.HỒ CHÍ MINH, tháng 12 năm 2014

# TÓM TẮT

Nhận dạng cử chỉ tay là trong những cách phổ biến mà con người áp dụng ngày càng rộng rãi trong những sản phẩm công nghệ thông tin cần sự tương tác giữa người với máy tính. Các cảm biến hình ảnh 3D được phát triển gần đây, ví dụ Kinect, ngoài nhận ảnh màu, còn có thể nhận dạng chiều sâu của vật thể đã mở ra nhưng cơ hội mới trong việc phát triển những ứng dụng tương tác giữa con người và máy tính. Đề tài "THIẾT KẾ HỆ THỐNG NHẬN DẠNG CỬ CHỈ TAY SỬ DỤNG CẢM BIẾN KINECT" đưa ra một phương pháp nhận dạng cử chỉ tay mới dựa trên hình ảnh độ sâu thu được từ cảm biến. Đầu tiên, vùng bàn tay được lấy ra bằng cách đặt các ngưỡng trên điểm được theo dấu trên bàn tay sử dụng thư viện NITE 2 cung cấp bởi PrimeSense. Bước thứ hai, tìm véc tơ đặc trưng: đặc trưng đầu tiên đề cập đến là số ngón tay được mở ra, đặc trưng thứ hai mô tả góc giữa đầu ngón tay so với phương ngang của bàn tay với tâm ngón tay là gốc, đặc trưng cuối cùng là hiệu của khoảng cách từ tâm của bàn tay đến các ngón tay và bán kính vòng tròn nội tiếp lớn nhất của bàn tay. Cuối cùng, máy vectơ hỗ trợ (SVM) được áp dụng để nhận ra những cử chỉ khác nhau. Kết quả thực nghiệm đánh giá cho thấy việc nhận dạng chính xác trong thời gian thực.

# MỤC LỤC

CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI ..... 5

1.1 Giới thiệu đề tài ..... 5

1.2 Mục tiêu luận văn và giới hạn đề tài ..... 6

CHƯƠNG II: HỆ THỐNG NHẬN DẠNG CỬ CHỈ TAY ..... 7

2.1 Tổng quan hệ thống đề xuất ..... 7

2.2 Phương pháp trính ảnh bàn tay và tiền xử lý ..... 8

2.2.1 Phát hiện bàn tay ..... 8

2.2.2 Lọc hình thái và làm trơn ảnh ..... 11

2.3.1 Phương pháp trích đặc trưng bàn tay ..... 11

2.3.1 Tìm đường bao bàn tay ..... 11

2.3.2 Tìm phương bàn tay và trọng tâm bàn tay ..... 13

2.3.3 Tìm tâm và bán kính lòng bàn tay ..... 14

2.3.4 Phương pháp tìm các đầu ngón tay ..... 15

2.4 Phân loại cử chỉ ..... 16

CHƯƠNG III: KẾT QUẢ THỰC HIỆN GIẢI THUẬT ..... 21

3.1 Kết quả hiện thực hệ thống ..... 21

3.2 Đánh giá hệ thống nhận dạng cử chỉ tay đề xuất ..... 22

3.3 Ứng dụng dựa trên hệ thống cử chỉ tay ..... 27

CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ..... 29

4.1 Kết luận ..... 29

4.2 Hướng phát triển ..... 29

Tài liệu tham khảo ..... 31

# DANH SÁCH HÌNH ẢNH

*Hình 1-1 Các ứng dụng nhận dạng cử chỉ tay* ..... 5
*Hình 1-2 Ba bước chính để nhận dạng cử chỉ tay* ..... 6
*Hình 2-2 Điểm theo dấu trên bàn tay có được sử dụng thư viện NITE 2* ..... 8
*Hình 2-3 Vùng bàn tay có được khi đặt 2 ngưỡng* $Th1$ *và* $Th2$ ..... 9
*Hình 2-4 Giải thuật cho phương pháp trích ảnh bàn tay* ..... 10
*Hình 2-5 Ảnh vùng bàn tay sau khi khử nhiễu* ..... 11
*Hình 2-6 Ảnh bàn tay với đường bao* ..... 12
*Hình 2-7 Giải thuật tách ảnh bàn tay* ..... 12
*Hình 2-8 Biểu diễn trục bàn tay và góc theo phương ngang* ..... 13
*Hình 2-9 Trọng tâm (điểm màu đỏ), tâm bàn tay (màu tím) và vòng tròn nội tiếp lớn nhất (vòng tròn màu đỏ)* ..... 15
*Hình 2-10 Thuật toán K-curvature* ..... 16
*Hình 2-11 Các đặc trưng của bàn tay được trính ra* ..... 17
*Hình 2-12 Mười một cử chỉ khác nhau được chứa trong cơ sở dữ liệu huấn luyện* ..... 18
*Hình 2-13 Giản đồ biểu diễn véc tơ đặc trưng của 11 cử chỉ* ..... 20
*Hình 3-1 Phần mềm hiện thực giải thuật đề xuất* ..... 21
*Hình 3-2 Hình ảnh độ sâu hiển thị trên phần mềm* ..... 22
*Hình 3-3 Dữ liệu huấn luyện* ..... 23
*Hình 3-4 Độ chính xác của hệ thống qua phương pháp kiểm tra mẫu* ..... 24
*Hình 3-5 Độ chính xác của hệ thống qua phương pháp đánh giá trực quan* ..... 24
*Hình 3-6 Độ chính xác trung bình của hệ thống qua phương pháp đánh giá trực quan* ..... 26
*Hình 3-7 Hệ thống điều khiển trò chơi hoặc ứng dụng trên máy tính sử dụng Kinect* ..... 27
*Hình 3-8 Bảng điều khiển cho trò chơi "búa, kéo, giấy"* ..... 28
*Hình 3-9 Trò chơi "búa, kéo, giấy" sử dụng cử chỉ tay* ..... 28

# CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI

## 1.1 Giới thiệu đề tài

Ngày nay dưới sự phát triển rộng rãi của các ứng dụng công nghệ thông tin vào trong cuộc sống, việc tương tác giữa con người và thiết bị điện tử ngày càng trở nên quan trọng và phổ biến. Con người có thể điều khiển các thiết bị máy tính, điện tử bằng giọng nói hay cử chỉ giống như khi tương tác giữa người với người trong thế giới thực mà không cần thông qua các thiết bị điều khiển khác như bàn phím, chuột hay các nút nhấn. Điều này đem lại cho người sử dụng một sự trải nghiệm mới, thú vị với các sản phẩm này.

Nhận dạng các cử chỉ của tay người là cách tự nhiên khi tương tác người với máy tính và ngày nay được nghiên cứu càng nhiều trong trong các trường đại học trên thế giới và được ứng dụng càng nhiều trong các lĩnh vực khác nhau như thiết bị trò chơi, tivi thông minh, thiết bị y tế, điều khiển robot, tương tác thực ảo,… Ví dụ cho ứng dụng tivi thông mình như ta chỉ cần phẩy tay là có thể chuyển kênh tivi, tăng giảm âm lượng thay vì phải tự tay nhấn nút trên bộ điều khiển.

*Hình 1-1 Các ứng dụng nhận dạng cử chỉ tay*

Đã có nhiều đề tài, ứng dụng nhận dạng cử chỉ tay nhưng nó vẫn là một nghiên cứu đầy thách thức do sự phức tạp hình học của bàn tay và những khó khăn trong việc nhận dạng bàn tay trong điều kiện ánh sáng yếu hay là trong việc phân biệt bàn tay và các đối tượng khác cùng màu sắc nếu chỉ sử dụng camera thông thường. Năm 2010, Micrsoft cho ra đời cảm biến Kinect dành cho máy Xbox 360, bên cạnh phục vụ cho mục đích chơi game, sản phẩm Kinect còn được dùng rộng rãi vào mục đích

nghiên cứu bởi vì ngoài thu nhận hình ảnh màu, nó còn thu nhận được hình ảnh độ sâu dựa vào camera hồng ngoại.

Cử chỉ tay có thể là tĩnh hoặc động. Nhận dạng cử chỉ tĩnh yêu cầu tĩnh phức tạp thấp hơn vì chỉ dựa vào phân tích một khung hình đơn lẻ. Nhận dạng cử chỉ động yêu cầu sự tính toán phức tạp hơn bởi vì cần tính toán trên các khung hình liên tục nhưng thích hợp cho tính toán thời gian thực. Trong luận văn này chúng tôi chỉ tập trung nghiên cứu và thiết kế hệ thống nhận dạng cho cử chỉ tĩnh.

Để nhận dạng cử chỉ tay tĩnh, bước đầu tiên trong các hệ thống nhận dạng là phát hiện ra vị trí bàn bay (segmentation); sau đó tiến hành xử lý hình ảnh, trích chọn đặc trưng (feature extraction) để tính toán các thông tin mô tả bàn tay, từ các dữ liệu quan sát và thực hiện công việc phân loại dựa vào các đặc tính đã được trích chọn để nhận dạng cử chỉ (classification). Đó là ba bước chính của một hệ thống nhận dạng cử chỉ tay.

*Hình 1-2 Ba bước chính để nhận dạng cử chỉ tay*

## 1.2 Mục tiêu luận văn và giới hạn đề tài

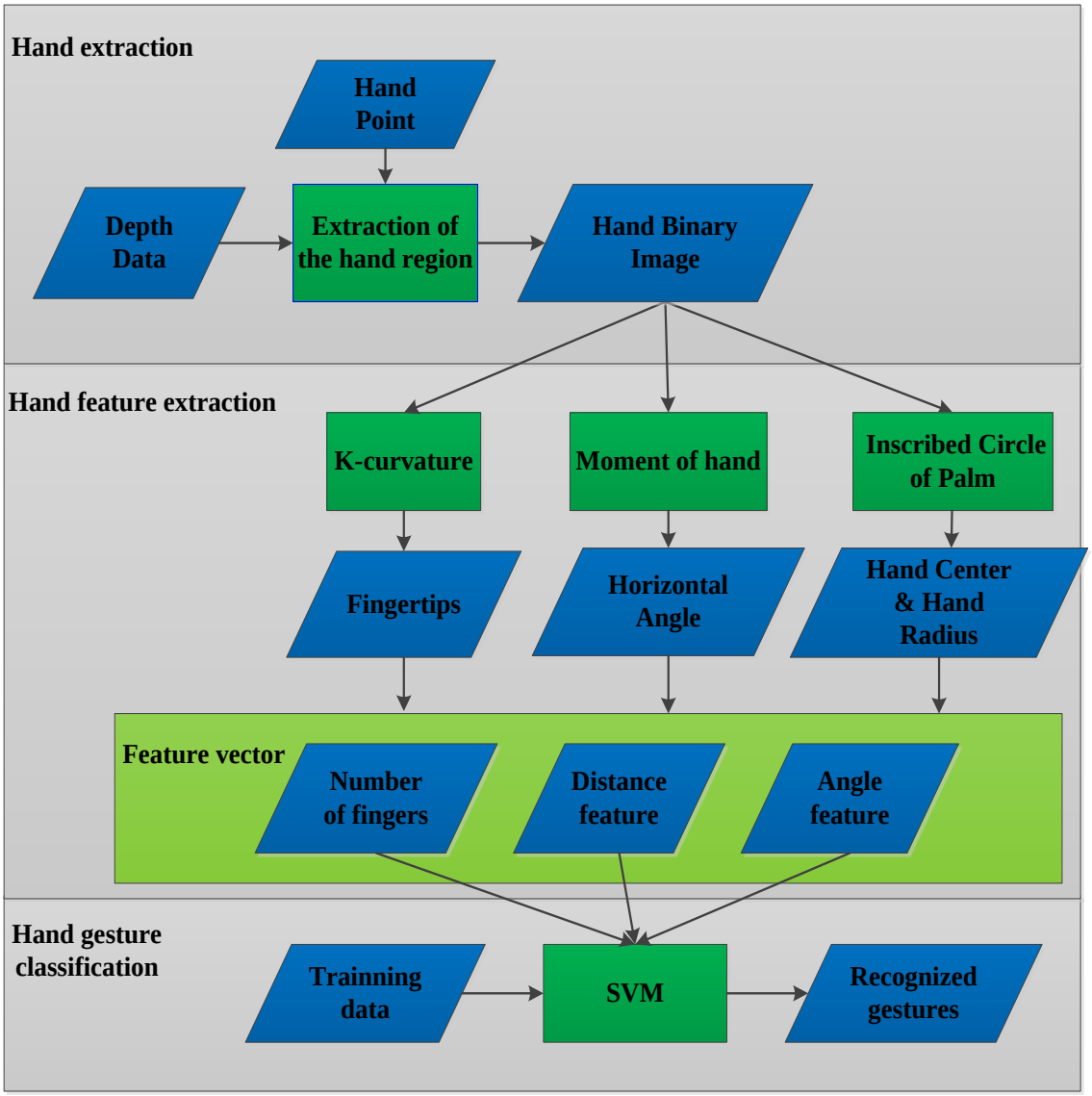
Mục tiêu của luận văn là thiết kế 1 hệ thống nhận dạng cử chỉ tay mà không cần bất kì công cụ hỗ trợ nào, có độ chính xác cao, tương thích với điều kiện trong bóng tối, hiển thị cử chỉ thời gian thực:

- Hiện thực trên máy tính sử dụng các thư viện C#, C++ của Windows, các thư viện hỗ trợ xử lý ảnh OpenCV và thư viện cho Kinect: OpenNi2, Nite2.
- Hệ thống chỉ cho phép nhận dạng với điều kiện không có ánh sáng mặt trời.
- Vùng nhận dạng giới hạn trong phạm vi [50cm – 1m].
- Nhận dạng 11 cử chỉ và góc nghiêng bàn tay không quá 45°.

# CHƯƠNG II: HỆ THỐNG NHẬN DẠNG CỬ CHỈ TAY

## 2.1 Tổng quan hệ thống đề xuất

Hệ thống nhận dạng cử chỉ sử dụng cảm biến Kinect đề xuất cũng bao gồm 3 bước như các hệ thống nhận dạng khác: Tách ngưỡng ảnh bàn tay, trích đặc trưng và phân loại cử chỉ. Sơ đồ khối của hệ thống nhận dạng cử chỉ tay đề xuất được thể hiện ở hình dưới đây.




*Hình 2-1 Sơ đồ khối hệ thống nhận dạng cử chỉ tay đề xuất*

Bước đầu tiên của hệ thống là cần phát hiện được vùng bàn tay và tách ảnh bàn tay ra khỏi hình nền. Dựa vào ảnh độ sâu thu được, vùng bàn tay được lấy ra bằng

cách đặt các ngưỡng độ sâu và ngưỡng khoảng cách tọa độ trên một điểm thuộc bàn tay được theo dấu sử dụng thư viện NITE 2 cung cấp bởi PrimeSense. Bước thứ hai, các đặc trưng của bàn tay được tìm ra bao gồm: vị trí những đầu ngón tay sử dụng thuật toán K-Curvature mà đã được áp dụng trong các bài báo [18], [19], [20]; góc giữa phương ngang của hình và trục bàn tay sử dụng phương pháp phân tích moment [21]; bán kính đường tròn nội tiếp lớn nhất của lòng bàn tay và tâm bàn tay [11], [12], [20]. Từ những đặc trưng của bàn tay được tìm ra, vec tơ đặc trưng gồm 15 phần tử được xây dựng, trong đó giá trị đầu tiên là số đầu ngón tay, 5 giá trị tiếp theo mô tả các góc giữa đầu ngón tay so với phương ngang của bàn tay với tâm bày tay là gốc, 4 giá trị kế tiếp là góc giữa các ngón tay với nhau, và 5 giá trị cuối cùng là những hiệu của khoảng cách từ tâm của bàn tay đến các đầu các ngón tay và bán kính vòng tròn nội tiếp lớn nhất của lòng bàn tay. Cuối cùng, máy vectơ hỗ trợ (SVM) được áp dụng cho véc tơ đặc trưng trên để nhận ra những cử chỉ khác nhau.

## 2.2 Phương pháp trính ảnh bàn tay và tiền xử lý

### 2.2.1 Phát hiện bàn tay

Bước đầu tiên trong nhận dạng cử chỉ tay là tách vùng bàn. Chúng tôi phát hiện vùng ảnh chứa bàn tay ra bằng cách sử dụng thư viện NITE2. Thư viện này cung cấp hàm theo dấu một điểm trên bàn tay sử dụng phương pháp Bayesian Object Localization. Người dùng đưa tay qua lại trước cảm biến, một điểm trên bàn tay sẽ được tìm ra sau một vài giây và sẽ được theo dấu đến khi nào bàn tay không còn hiện diện trước cảm biến.

*Hình 2-2 Điểm theo dấu trên bàn tay có được sử dụng thư viện NITE 2*

Giả sử điểm trên bàn tay sẽ có tọa độ ảnh $P_h(x_h, y_h)$ và độ sâu tại đó là $D_h$. Vùng bàn tay được tách ra bằng cách đặt 1 ngưỡng độ sâu $T_{h1}$ và 1 ngưỡng khoảng cách tọa độ theo 2 phương x, y tình từ $P_h$. Gọi $P_i(x_i, y_i)$ là điểm ảnh trên ảnh độ sâu thu được và $D_i$ là độ sâu tại đó. Chúng tôi định nghĩa vùng bàn tay theo công thức:

$$H = \{\boldsymbol{P_i} \mid ((|x_i - x_h| < T_{h1}) \cap (|y_i - y_h| < T_{h1})) \cap (D_i < D_h + T_{h2})\} \quad (1)$$

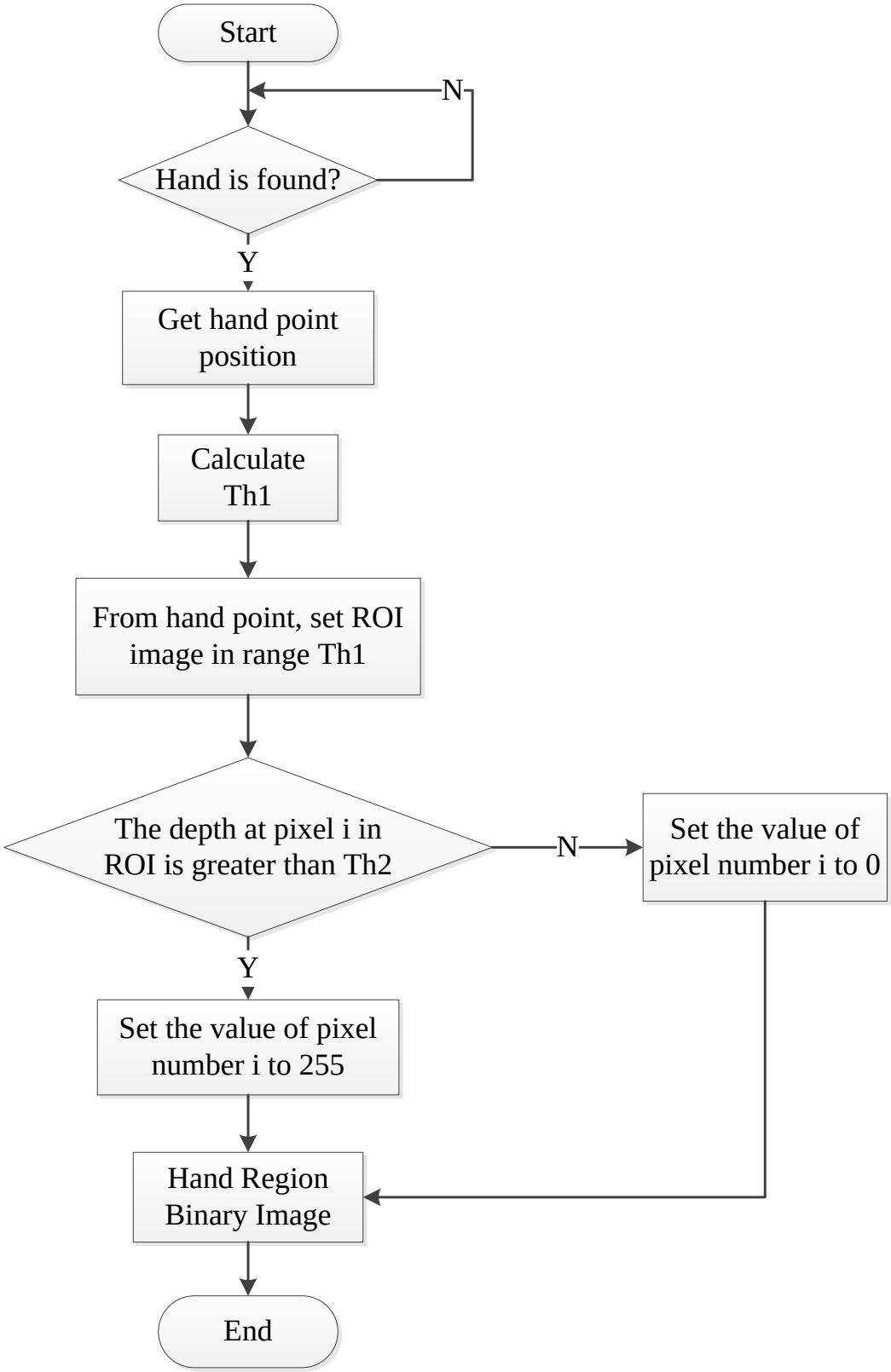
Giá trị của $T_{h1}$ được thiết lập tùy thuộc vào độ sâu của bàn tay để làm sao tách được vùng bàn tay một cách thích hợp. Trong luận văn này, chúng tôi chọn các ngưỡng với $T_{h1} = \frac{64000}{D(X_h)}$, $T_{h2} = 100$ (mm). Vùng ảnh bàn tay H được chuyển qua ảnh nhị phân, nó sẽ bao gồm vùng trắng với diện tích lớn nhất chính là ảnh bàn tay, vùng đen là nền, ngoài ra còn có thể có một số ảnh nhiễu nhỏ.

*Hình 2-3 Vùng bàn tay có được khi đặt 2 ngưỡng $\boldsymbol{T_{h1}}$ và $\boldsymbol{T_{h2}}$*

Tổng số điểm ảnh trên vùng bàn tay sẽ thay đổi khi khoảng cách giữa bàn tay đến Camera thay đổi, chúng tôi đặt ra một hệ số tỉ lê là $Scl$, nó sẽ áp dụng cho các thuật toán ở các bước xử lý sau.

$$Scl = \frac{5000}{D_h} \quad (2)$$

Giải thuật tách vùng bàn tay:




*Hình 2-4 Giải thuật cho phương pháp trích ảnh bàn tay*

### 2.2.2 Lọc hình thái và làm trơn ảnh

Trước khi triển khai các giải thuật trích đặc trưng cho ảnh vùng bàn tay được lấy ra ở trên, nhiễu trên vùng ảnh bàn tay cần được loại bỏ. Để loại bỏ nhiễu một cách hiệu quả, chúng tôi áp dụng lọc hình thái với phép đóng ảnh (giãn ảnh sau đó co ảnh) với phần tử cấu trúc hình elíp chiều rộng và cao đều bằng 4, gốc là tâm elíp. Tiếp theo chúng tôi sử dụng một bộ lọc trung vị với kích thước 7x7 để làm trơn biên ảnh.

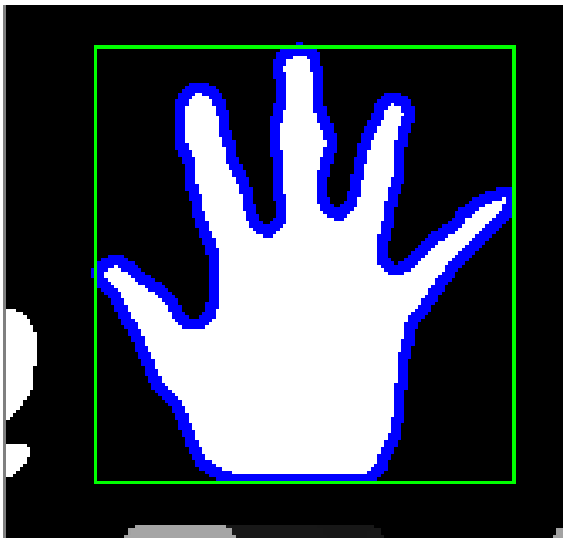
*Hình 2-5 Ảnh vùng bàn tay sau khi khử nhiễu*

### 2.3.1 Phương pháp trích đặc trưng bàn tay

### 2.3.1 Tìm đường bao bàn tay

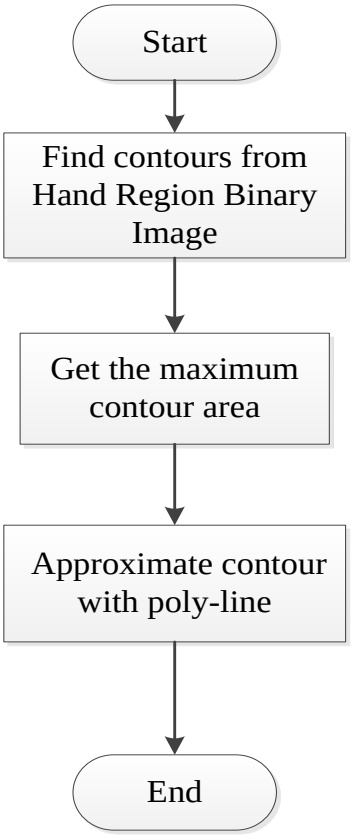
Vùng bàn tay sau khi được tách ra có thể còn bao gồm các ảnh nhỏ (nhiễu), chúng tôi sử dụng thuật toán phân tích cấu trúc hình thái của tác giả SATOSHI SUZUKI và KEIICHI ABE [13] được tích hợp sẵn trong bộ thư viện mã nguồn mở OPENCV để tìm đường bao các ảnh. Ảnh có diện tích đường bao lớn nhất là chính là ảnh bàn tay. Ngõ ra của bước này là danh sách các điểm thuộc đường bao. Chúng

tôi áp dụng thêm một phép xấp xỉ đa giác để giảm số lượng điểm ảnh, nhằm mục đích giảm khối lượng tính toán cho các bước trích đặc trưng sau này mà có sử dụng đường bao.



*Hình 2-6 Ảnh bàn tay với đường bao*

Giải thuật thực hiện:




*Hình 2-7 Giải thuật tách ảnh bàn tay*

### 2.3.2 Tìm phương bàn tay và trọng tâm bàn tay

Trong luận văn này chúng tôi áp dụng tính chất mô ment của một hình [21], để tìm ra góc giữa phương ngang của bàn tay đối với trục nằm ngang của hình và trọng tâm của bàn tay.

Nhắc lại công thức tính mô ment của 1 ảnh xám:

$$m(p,q) = \sum_{i=1}^{n} I(x,y) x^p y^q \quad (3)$$

Trong đó $I(x,y)$ là giá trị tại tọa độ $I(x,y)$, n là tổng số điểm ảnh trên hình, $m(p,q)$ là mô ment bậc $(p,q)$. Trọng tâm của bàn tay sẽ được tính thông qua $m(1,0)$, $m(0,1)$ và $m(0,0)$:

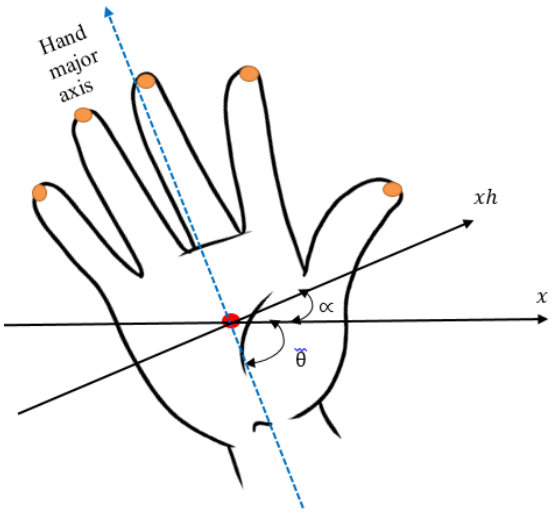
$$x_C = \frac{m(1,0)}{m(0,0)}, \quad y_C = \frac{m(0,1)}{m(0,0)} \quad (4)$$

Góc giữa trục đứng bàn tay và phương ngang của hình sẽ được tính theo công thức:

$$\theta = \frac{1}{2}(\arctan(\frac{2m(1,1)}{m(2,0)-m(0,2)})) \quad (5)$$

Từ đây góc giữa phương ngang bàn tay và phương ngang của hình sẽ được xác định:

$$\alpha = 90° - \theta \quad (6)$$




*Hình 2-8 Biểu diễn trục bàn tay và góc theo phương ngang*

Góc phương ngang $\alpha$ sẽ được sử dụng để tính toán những đặc trưng về góc để khi bàn tay thay đổi hướng thì đặc trưng về góc thay đổi không nhiều.

### 2.3.3 Tìm tâm và bán kính lòng bàn tay

Tâm và bán kính lòng bàn tay được tìm ra nhằm mục đính chính để tính toán cho cái đại lượng đặc trưng khoảng cách. Vị trí trọng tâm ở mục **2.3.2** sẽ bị thay đổi khi chuyển đổi cử chỉ tay do mật độ điểm ảnh thay đổi khiến cho các đại lượng mô ment thay đổi. Vì vậy cần có một phương pháp khác để tìm ra tâm lòng bàn tay với vị trí ổn định. Trong luận văn này chúng tôi sử dụng phương pháp tìm tâm bàn tay mà đã được áp dụng ở các bài báo [11], [12], [20]; đó là phương pháp tìm vòng tròn nội tiếp lớn nhất của bàn tay. Với mỗi điểm ảnh $\boldsymbol{P_i}$ trên bày tay, khoảng cách nhỏ nhất $\boldsymbol{D_{imin}}$ của nó đó đến đường bao sẽ được tính toán. Điểm mà có giá trị khoảng cách lớn nhất trong tập khoảng cách nhỏ nhất đó chính là tâm lòng bàn tay $\boldsymbol{P_h}$ và khoảng cách đó chính là bán kính **R**. Vòng tròn nội tiếp lớn nhất đại diện cho vùng lòng bàn tay (Palm Region). Khối lượng tính toán giữa toàn điểm ảnh trên bàn tay và đường bao là rất lớn. Ví dụ, nếu chúng ta có N điểm ảnh trên bàn tay, M là số điểm ảnh trên đường bao, thì số vòng lặp sẽ là N x M + N. Chúng tôi giảm khối lượng tính toán bằng cách chỉ xem xét những điểm ảnh lân cận trọng tâm bàn tay.

Giải thuật tìm tâm bàn tay và bán kính lòng bàn tay được mô tả theo những công thức như sau:

$$G = \{P_i \mid ( d(P_i,\ P_c ) < T_{h3} \} \qquad (7)$$

G là tập hợp tất cả những điểm cần xét để tìm tâm và bán kính bàn tay. $d(P_i,\ P_c)$ là khoảng cách Euclid từ điểm trên bàn tay đến điểm trọng tâm của bàn tay $P_c$ cái mà được tính ở công thức (4) mục **3.3.2.** $T_{h3}$ là ngưỡng khoảng cách với $T_{h3} = 4 * Scl$ .

$$D_{imax} = Min(d(P_i, P_b(j)) \mid (P_i \in G \cap P_b(j) \in C)) \qquad (8)$$

Trong đó **C** là tập hợp tất cả các điểm thuộc đường bao, $P_b(j)$ thứ $j$ trên đường bao **C** đó. Bán kính lòng bàn tay R sẽ là:

$$R = \mathrm{Max}(D_{imax} \mid P_i \in G ) \qquad (9)$$

*Hình 2-9 Trọng tâm (điểm màu đỏ), tâm bàn tay (màu tím) và vòng tròn nội tiếp lớn nhất (vòng tròn màu đỏ)*

Kết quả của giải thuật cho thấy tâm lòng bàn tay ổn định khi thay đổi các cử chỉ khác nhau.

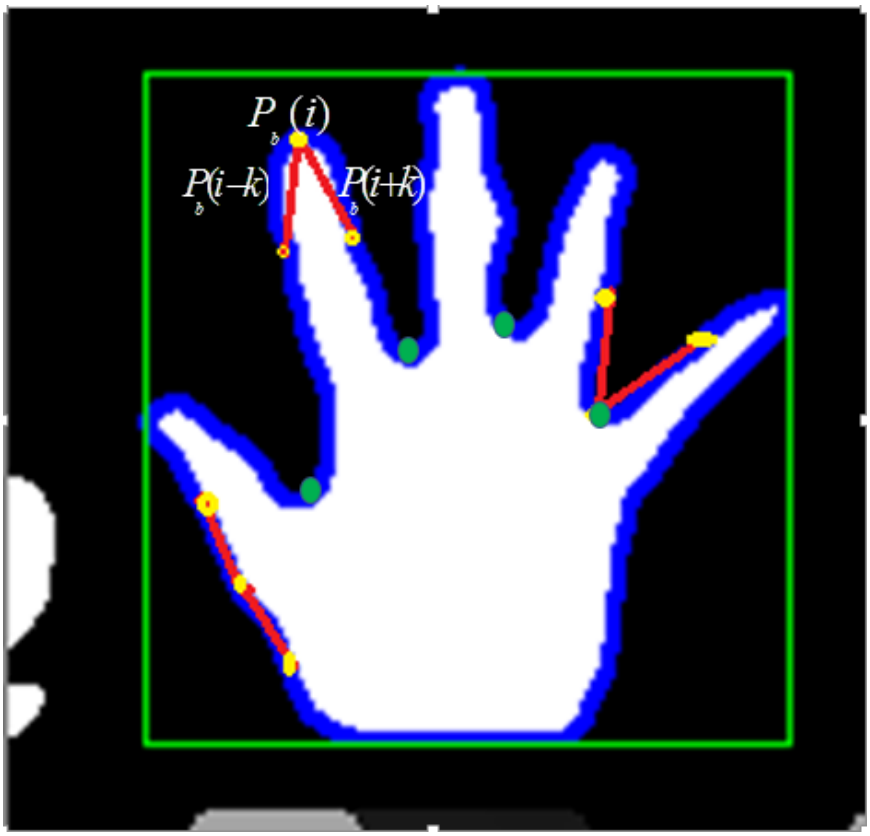
### 2.3.4 Phương pháp tìm các đầu ngón tay

Chúng tôi cải tiến phương pháp k-curvature để tìm ra các đầu ngón tay mà các bài báo trước đó ([18], [19], [20]) đã áp dụng. K-curvature là phương pháp tính góc giữa 2 vec tơ trên đường bao. Tại mỗi điểm $P_b(i)$ trên tập điểm mô tả đường bao, góc giữa 2 véc tơ $P_b(i-k)$ và $P_b(i+k)$ theo hướng ngược nhau sẽ được tìm ra. Những điểm đầu ngón tay và kẽ tay sẽ có góc k-curvature nhỏ thua một ngưỡng được chỉ ra. Thay vì chọn các hệ số k và góc ngưỡng là cố định (Bài báo số [18] chọn k = 10, góc ngưỡng là 30°; còn báo số [20] chọn K bằng 20 và góc ngưỡng là 60°), trong luận văn này chúng tôi chọn chúng tùy thuộc vào sự thay đổi của độ sâu bàn tay đối với cảm biến mà được đại diện bởi hệ số $Scl$ được chỉ ra ở công thức số (2) mục **3.3.1**:

Hệ số k-curvature : $k = 2 * Scl$ (10)

Góc ngưỡng k- curvature: $\theta_k = \frac{75}{Scl} + 60°$ (11)

Hai công thức trên được chúng tôi rút ra bằng thực nghiệm khi thực hiện thuật toán trong khoảng cách từ 50cm – 1. Hình dưới đây miêu tả thuật toán k-curvature:




*Hình 2-10 Thuật toán K-curvature*

Những điểm ở kẽ tay (màu xanh lá) có khoảng cách nhỏ thua 1.2 lần bán kình vòng tròn nội tiếp tìm ra ở mục 3.3 sẽ bị loại bỏ. Những đầu ngón tay sử dụng thuật toán k-curvature được tìm ra cho kết quả một cách chính xác trong phạm vi 50 cm – 1 m.

## 2.4 Phân loại cử chỉ

Từ những đặc trưng về hình ảnh bàn tay tìm được ở trên, chúng tôi định nghĩa 1 vec tơ đặc trưng gồm 15 phần tử như sau:

**Vf = {N, α[i], β[j], *D*[i]}** (12)

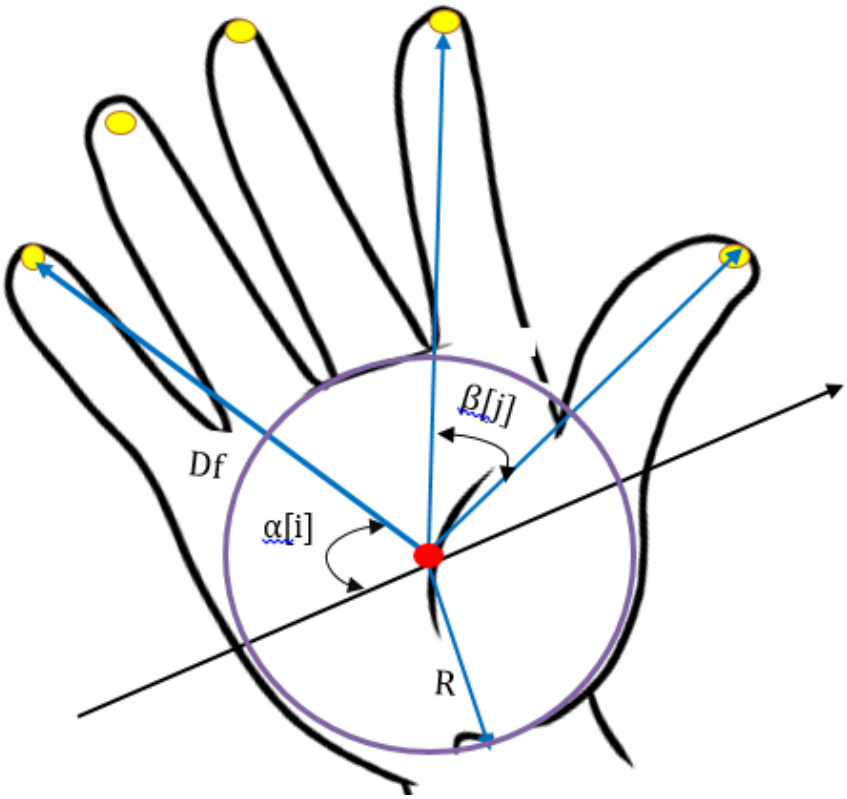
Trong đó:

**N**: là số các ngón tay được mở ra.

**α[i]**: là góc từ tâm bàn tay cho tới các đầu ngón tay so với trục ngang của bàn tay, i = 0…4

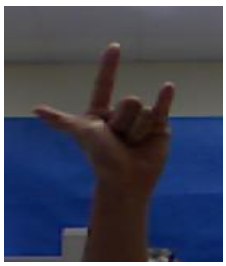
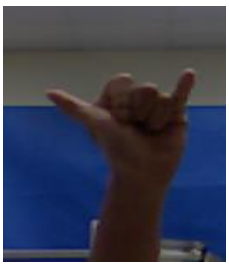
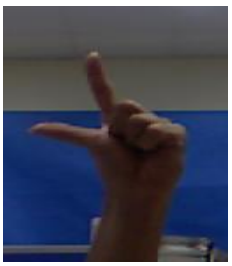
**β[j]**: là góc giữa các ngón tay liên tiếp với nhau, j = 0…3

***D*[i]**: là hiệu của khoảng cách từ tâm bàn tay đến cách đầu ngón tay thứ i ( *Df* [i]) và bán kính đường tròn nội tiếp lớn nhất (R). Nó cũng được tỉ lệ với độ sâu: $D[i] = \frac{(Df[i] - R)}{Scl}$; i = 0..4 (13)



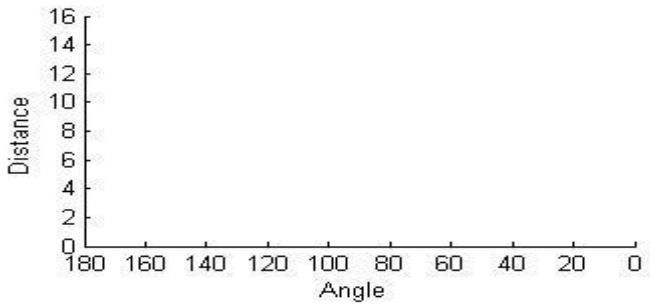
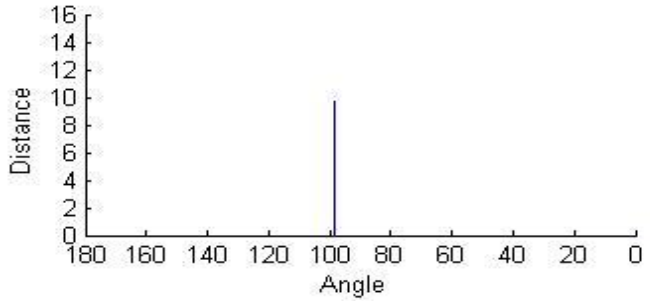

*Hình 2-11 Các đặc trưng của bàn tay được trính ra*

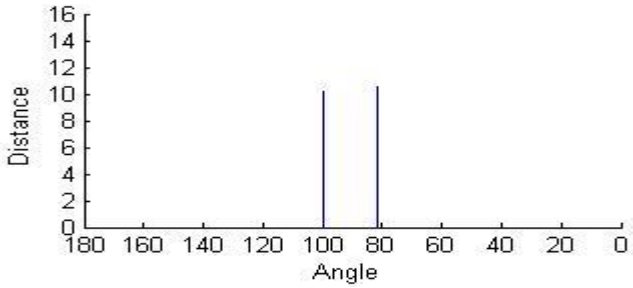
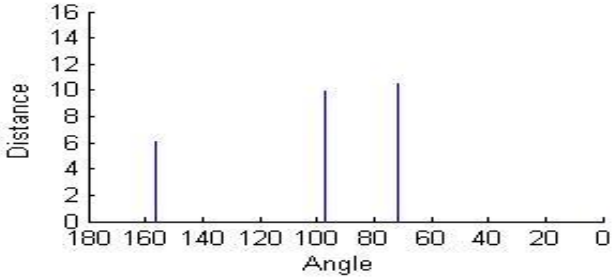
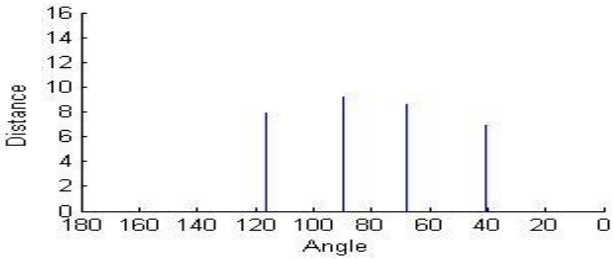
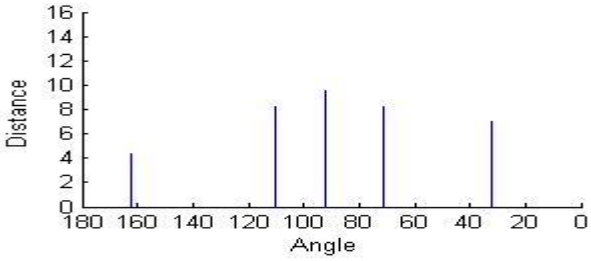
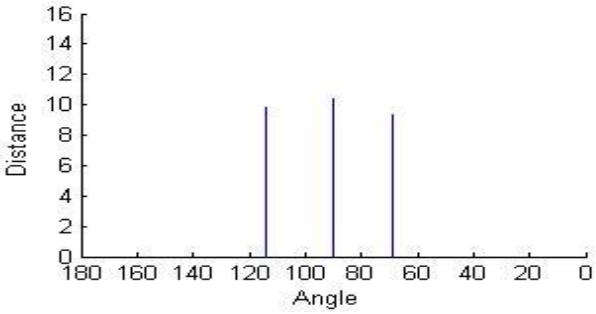
Mỗi cử chỉ tay sẽ tương ứng với một vec tơ đặc trưng Vf. Sau khi xây dựng 1 dữ liệu huấn luyện từ các mẫu, chúng tôi áp dụng một máy hỗ trợ vec tơ (SVM) dựa trên phương pháp one-vs-one (mục **2.2.3**) để phân loại các cử chỉ khác nhau. Hệ thống nhận dạng cử chỉ tay mà chúng tôi đề xuất cho phép nhận dạng 11 cử chỉ khác nhau dưới đây:

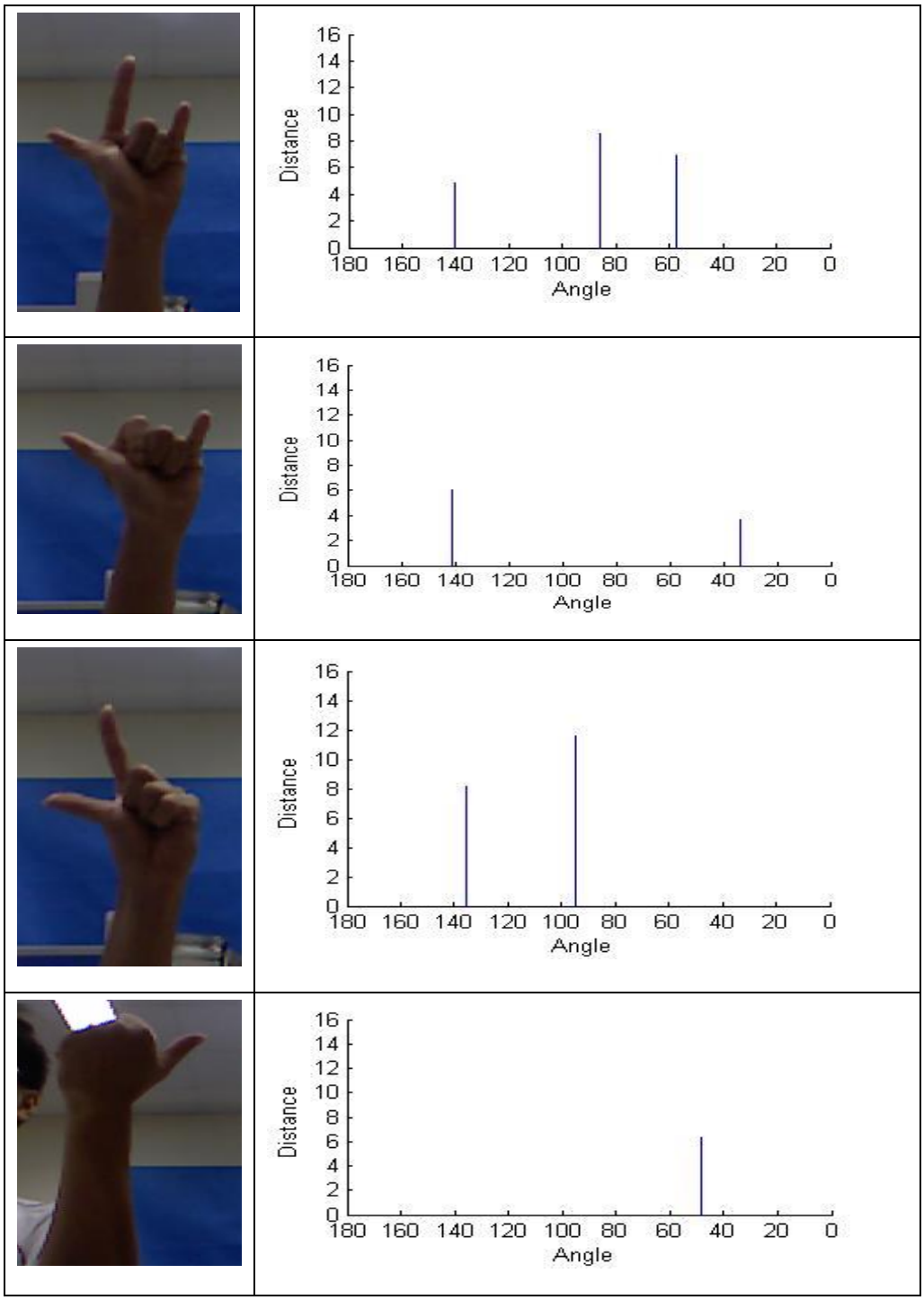
| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| **Gesture 0** | **Gesture 1** | **Gesture 2** | **Gesture 3** | **Gesture 4** | **Gesture 5** |
| |  |  |  | | |
| **Gesture 6** | **Gesture 7** | **Gesture 8** | **Gesture 9** | **Gesture 10** | |

*Hình 2-12 Mười một cử chỉ khác nhau được chứa trong cơ sở dữ liệu huấn luyện*

Ví dụ cho véc tơ đặc trưng với một mẫu tương ứng với cử chỉ**:**

| **Gesture** | **Vector Feature** |
|---|---|
| |   |
| |   |

| | |
|---|---|
| |  Distance<br>Angle |
| |  Distance<br>Angle |
| |  Distance<br>Angle |
| |  Distance<br>Angle |
| |  Distance<br>Angle |

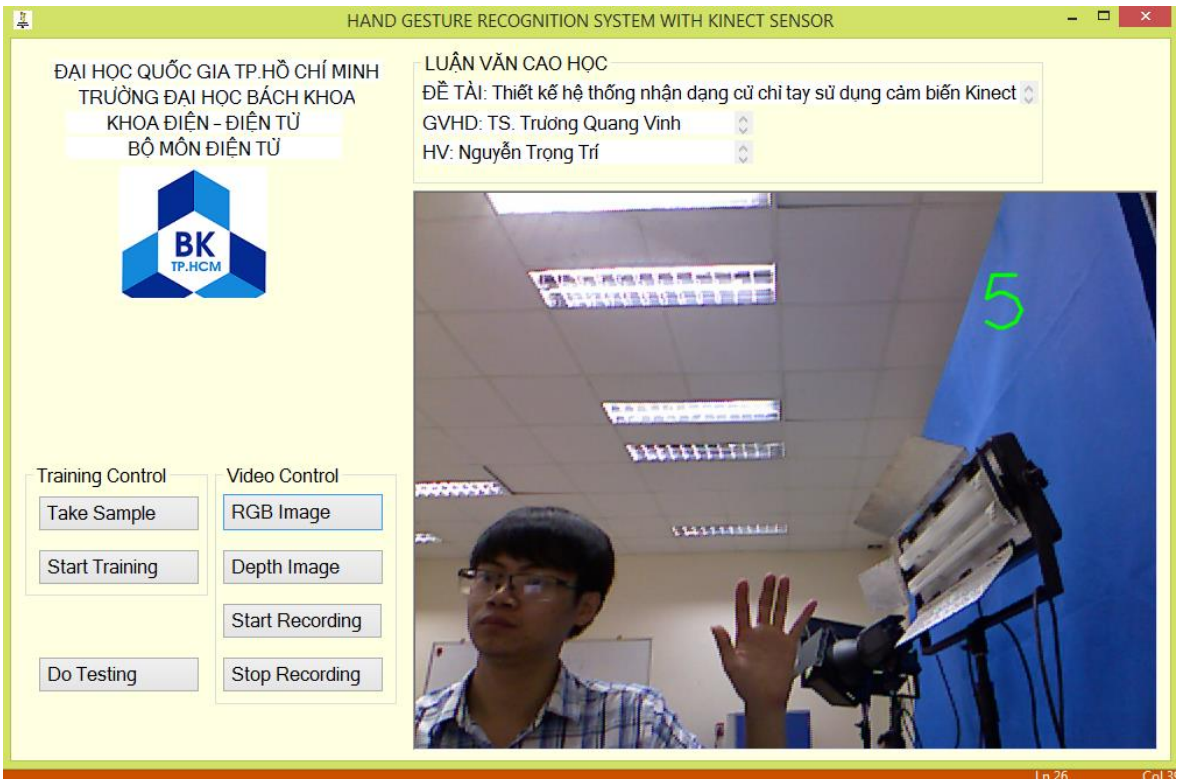



*Hình 2-13 Giản đồ biểu diễn véc tơ đặc trưng của 11 cử chỉ*

# CHƯƠNG III: KẾT QUẢ THỰC HIỆN GIẢI THUẬT

## 3.1 Kết quả hiện thực hệ thống

Hệ thống nhận dạng cử chỉ tay của chúng tôi được hiện thực trên máy tính Intel core I5-560M, RAM 4G với hệ điều hành Windows 8.1. Giải thuật được phát triển trên môi trường Visual Studio 2012 với ngôn ngữ C++ cho phần giải thuật và C# cho phần giao diện. Chúng tôi sử dụng thư viện OPENNI2 cho phần kết nối cảm biến và máy tính, thư viện NITE2 cho phần theo dấu bàn tay và thư viện OPENCV cho một số thuật toán xử lý ảnh và phần nhận dạng sử dụng SVM. Kết quả phần mềm hiện thực giải thuật thể hiện ở hình dưới đây.




*Hình 3-1 Phần mềm hiện thực giải thuật đề xuất*

Các nút điều khiển trên phần mềm:

| **Phím điều khiển** | **Mô tả chức năng** |
|---|---|
| RGB Image | Cho phép hiển thị ảnh màu |

| | |
|---|---|
| Depth Image | Cho phép hiển thị ảnh màu |
| Take Sample | Lấy mẫu ảnh độ sâu và vị trí bàn tay |
| Start Recording | Bắt đầu ghi video |
| Stop Recording | Dừng ghi video |
| Start Training | Cho phép huấn luyện mẫu |
| Do Testing | Cho phép thực hiện kiểm tra từ các mẫu thu được |

*Hình 3-2 Hình ảnh độ sâu hiển thị trên phần mềm*

## 3.2 Đánh giá hệ thống nhận dạng cử chỉ tay đề xuất

Để đánh giá kết quả chính xác của hệ thống nhận dạng cử chỉ tay của chúng tôi đề xuất, chúng tôi mời 5 học viên trong IC-Lab trường đại học trong độ tuổi 20-30 thực hiện các 11 cử chỉ ở trên trong phạm vi 50cm-1m, mỗi cử chỉ thực hiện bởi từng người chúng tôi lấy mẫu ít nhất 10. Tổng cộng chúng tôi có 620 mẫu để huấn luyện cho SVM.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 96.436 | 0.000 | 0.000 | 0.000 | 0.000 | 8.316 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1 |
| 1 | 101.000 | 0.000 | 0.000 | 0.000 | 0.000 | 8.584 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1 |
| 1 | 97.106 | 0.000 | 0.000 | 0.000 | 0.000 | 9.929 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1 |
| 2 | 103.248 | 83.194 | 0.000 | 0.000 | 0.000 | 8.414 | 7.601 | 0.000 | 0.000 | 0.000 | 19.147 | 0.000 | 0.000 | 0.000 | 2 |
| 2 | 102.260 | 82.533 | 0.000 | 0.000 | 0.000 | 9.545 | 9.280 | 0.000 | 0.000 | 0.000 | 18.289 | 0.000 | 0.000 | 0.000 | 2 |
| 2 | 96.627 | 76.045 | 0.000 | 0.000 | 0.000 | 10.124 | 10.844 | 0.000 | 0.000 | 0.000 | 17.133 | 0.000 | 0.000 | 0.000 | 2 |
| 3 | 174.196 | 116.730 | 88.041 | 0.000 | 0.000 | 6.742 | 10.903 | 8.435 | 0.000 | 0.000 | 57.527 | 29.577 | 0.000 | 0.000 | 3 |
| 3 | 144.985 | 94.975 | 67.959 | 0.000 | 0.000 | 6.675 | 8.582 | 8.935 | 0.000 | 0.000 | 44.979 | 25.645 | 0.000 | 0.000 | 3 |
| 3 | 151.344 | 95.003 | 73.979 | 0.000 | 0.000 | 5.759 | 9.137 | 6.639 | 0.000 | 0.000 | 52.534 | 19.357 | 0.000 | 0.000 | 3 |
| 3 | 160.870 | 97.919 | 68.458 | 0.000 | 0.000 | 6.313 | 10.604 | 9.983 | 0.000 | 0.000 | 51.849 | 25.221 | 0.000 | 0.000 | 3 |
| 4 | 118.387 | 94.014 | 73.212 | 37.849 | 0.000 | 7.510 | 8.500 | 6.992 | 6.189 | 0.000 | 25.006 | 21.499 | 35.097 | 0.000 | 4 |
| 4 | 124.534 | 103.182 | 75.303 | 48.571 | 0.000 | 6.895 | 6.393 | 7.682 | 7.246 | 0.000 | 21.352 | 27.879 | 26.733 | 0.000 | 4 |
| 4 | 123.773 | 94.496 | 72.646 | 44.310 | 0.000 | 6.961 | 8.634 | 8.797 | 6.720 | 0.000 | 27.761 | 20.806 | 26.794 | 0.000 | 4 |
| 5 | 183.467 | 123.174 | 99.525 | 76.332 | 35.520 | 5.389 | 8.547 | 9.468 | 8.951 | 7.803 | 50.659 | 19.646 | 19.428 | 35.507 | 5 |
| 5 | 161.133 | 119.652 | 91.731 | 73.383 | 36.639 | 4.005 | 8.228 | 10.432 | 9.225 | 7.711 | 33.814 | 23.629 | 15.730 | 31.201 | 5 |
| 5 | 190.103 | 135.264 | 103.027 | 79.891 | 36.084 | 5.312 | 7.207 | 9.369 | 9.080 | 7.114 | 45.913 | 25.953 | 19.211 | 37.322 | 5 |
| 3 | 106.163 | 82.095 | 59.808 | 0.000 | 0.000 | 10.340 | 10.845 | 9.154 | 0.000 | 0.000 | 19.964 | 18.122 | 0.000 | 0.000 | 6 |
| 3 | 104.254 | 82.693 | 61.310 | 0.000 | 0.000 | 9.818 | 10.121 | 8.818 | 0.000 | 0.000 | 18.058 | 17.645 | 0.000 | 0.000 | 6 |
| 3 | 114.094 | 87.294 | 63.195 | 0.000 | 0.000 | 9.189 | 9.637 | 8.729 | 0.000 | 0.000 | 22.233 | 20.022 | 0.000 | 0.000 | 6 |
| 3 | 141.058 | 87.581 | 42.461 | 0.000 | 0.000 | 5.515 | 9.091 | 4.015 | 0.000 | 0.000 | 49.887 | 41.604 | 0.000 | 0.000 | 7 |
| 3 | 136.799 | 88.202 | 37.406 | 0.000 | 0.000 | 5.466 | 8.646 | 3.702 | 0.000 | 0.000 | 47.771 | 50.307 | 0.000 | 0.000 | 7 |
| 3 | 137.447 | 91.727 | 39.682 | 0.000 | 0.000 | 4.951 | 8.470 | 4.238 | 0.000 | 0.000 | 44.700 | 51.173 | 0.000 | 0.000 | 7 |
| 2 | 165.185 | 64.593 | 0.000 | 0.000 | 0.000 | 6.212 | 4.221 | 0.000 | 0.000 | 0.000 | 114.067 | 0.000 | 0.000 | 0.000 | 8 |
| 2 | 140.896 | 46.248 | 0.000 | 0.000 | 0.000 | 6.361 | 4.633 | 0.000 | 0.000 | 0.000 | 113.820 | 0.000 | 0.000 | 0.000 | 8 |
| 2 | 146.117 | 37.662 | 0.000 | 0.000 | 0.000 | 5.721 | 3.794 | 0.000 | 0.000 | 0.000 | 122.284 | 0.000 | 0.000 | 0.000 | 8 |
| 2 | 165.969 | 83.502 | 0.000 | 0.000 | 0.000 | 6.694 | 9.883 | 0.000 | 0.000 | 0.000 | 85.918 | 0.000 | 0.000 | 0.000 | 9 |
| 2 | 170.623 | 83.439 | 0.000 | 0.000 | 0.000 | 6.417 | 10.265 | 0.000 | 0.000 | 0.000 | 89.807 | 0.000 | 0.000 | 0.000 | 9 |
| 2 | 154.746 | 75.496 | 0.000 | 0.000 | 0.000 | 7.130 | 10.026 | 0.000 | 0.000 | 0.000 | 82.128 | 0.000 | 0.000 | 0.000 | 9 |
| 1 | 59.336 | 0.000 | 0.000 | 0.000 | 0.000 | 6.185 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 10 |
| 1 | 62.306 | 0.000 | 0.000 | 0.000 | 0.000 | 5.746 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 10 |
| 1 | 51.717 | 0.000 | 0.000 | 0.000 | 0.000 | 5.516 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 10 |

*Hình 3-3 Dữ liệu huấn luyện*
*(Cột A-O là các đặc trưng, cột P là cử chỉ tương ứng)*

Sau đó chúng tôi thực hiện 2 phương pháp để đánh giá cho hệ thống: phương pháp kiểm tra mẫu và phương pháp trực quan.

**Phương pháp kiểm tra mẫu:**

Phương pháp thứ nhất, chúng tôi cũng lấy mẫu trên từng người 9 người khác nhau, mỗi người thực hiện 11 cử chỉ khác nhau, mỗi cử chỉ cũng lặp lại khoảng trên 10 lần, tổng cộng có 1926 mẫu ảnh độ sâu với độ sâu và tọa độ điểm trên bàn tay tương ứng. Kết quả cho việc đánh giá phương pháp pháp thứ nhất phản ảnh bởi bảng Confusion Matrix dưới đây:

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **G0** | 112 | 3 | | | | | | | | | |
| **G1** | 1 | 105 | 4 | | | | | | | | 8 |
| **G2** | | 2 | 144 | | | | | | 2 | 9 | |
| **G3** | | | | 221 | | | | 15 | | | |
| **G4** | | | | | 120 | 5 | 2 | | | | |
| **G5** | | | | | 5 | 139 | 1 | | | | |
| **G6** | | | | 2 | | | 185 | 2 | | | |
| **G7** | | | | 12 | 1 | | 2 | 230 | | 1 | |
| **G8** | | | 1 | | 2 | | | | 210 | 7 | 1 |
| **G9** | | | | | | | | 1 | 14 | 235 | |
| **G10** | | | 1 | | | | | | | | 121 |
| **Accuracy** | **99,12%** | **95,5%** | **96%** | **94%** | **93,8%** | **97%** | **97,4%** | **93%** | **93%** | **93,3%** | **93,08%** |

*Hình 3-4 Độ chính xác của hệ thống qua phương pháp kiểm tra mẫu*

**Phương pháp trực quan:**

Trong phương pháp đánh giá thứ 2, chúng tôi mời 10 người thực hiện các cử chỉ và lưu lại các video họ thực hiện. Ở phương pháp này chúng tôi thực hiện kết hợp các khung hình liên tiếp với nhau để cho ra kết quả cử chỉ. Nếu 3 khung hình liên tiếp cho ra cùng 1 cử chỉ thì cử chỉ mới được hiển thị, còn không thì vẫn giữ cử chỉ trước đó. Độ chính xác kiểm nghiệm đối với phương pháp thứ 2 đối với từng người sử dụng:

| Người dùng | Kết quả |
|---|---|
| | |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **G0** | 1 | | | | | | | | | | 0.1 |
| **G1** | | 1 | | | | | | | | | |
| **G2** | | | 1 | | | | 0.1 | | | | |
| **G3** | | | | 1 | | | | | | | |
| **G4** | | | | | 1 | | | | | | |
| **G5** | | | | | | 1 | | | | | |
| **G6** | | | | | | | 0.9 | | | | |
| **G7** | | | | | | | | 1 | | | |
| **G8** | | | | | | | | | 1 | | |
| **G9** | | | | | | | | | | 1 | |
| **G10** | | | | | | | | | | | 0.9 |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **G0** | 1 | | | | | | | | | | |
| **G1** | | 1 | | | | | | | | | |
| **G2** | | | 1 | | | | | | | | |
| **G3** | | | | 1 | | | | | | | |
| **G4** | | | | | 1 | | | | | | |
| **G5** | | | | | | 1 | | | | | |
| **G6** | | | | | | | 1 | | | | |
| **G7** | | | | | | | | 1 | | | |
| **G8** | | | | | | | | | 1 | | |
| **G9** | | | | | | | | | | 1 | |
| **G10** | | | | | | | | | | | 1 |

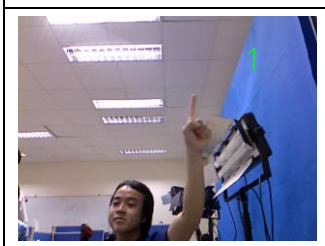
| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | | | | | | | | |
| G3 | | | | 1 | | | | | | | |
| G4 | | | | | 0.9 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | | | 1 | | | | |
| G7 | | | | | | | | 1 | | | |
| G8 | | | | | 0.1 | | | | 1 | | |
| G9 | | | | | | | | | | 1 | |
| G10 | | | | | | | | | | | 1 |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | | | | | | | | |
| G3 | | | | 1 | | | | | | | |
| G4 | | | | | 1 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | | | 1 | | | | |
| G7 | | | | | | | | 1 | | | |
| G8 | | | | | | | | | 1 | | |
| G9 | | | | | | | | | | 1 | |
| G10 | | | | | | | | | | | 1 |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | 0.1 | | | | | | | |
| G3 | | | | 0.9 | | | 0.2 | 0.2 | | | |
| G4 | | | | | 0.8 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | 0.1 | | 0.7 | | | | |
| G7 | | | | | | | 0.1 | 0.7 | | | |
| G8 | | | | | | | | | 0.9 | | |
| G9 | | | | | 0.1 | | | 0.1 | 0.1 | 1 | |
| G10 | | | | | | | | | | | 1 |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | | | | | | | | |
| G3 | | | | 1 | | | | | | | |
| G4 | | | | | 1 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | | | 1 | | | | |
| G7 | | | | | | | | 1 | | | |
| G8 | | | | | | | | | 1 | | |
| G9 | | | | | | | | | | 1 | |
| G10 | | | | | | | | | | | 1 |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | | | | | | | | |
| G3 | | | | 1 | | | | | | | |
| G4 | | | | | 1 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | | | 1 | | | | |
| G7 | | | | | | | | 1 | | | |
| G8 | | | | | | | | | 1 | | |
| G9 | | | | | | | | | | 1 | |
| G10 | | | | | | | | | | | 1 |



| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | | | | | | | | |
| G3 | | | | 1 | | | | | | | |
| G4 | | | | | 1 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | | | 1 | | | | |
| G7 | | | | | | | | 1 | | | |
| G8 | | | | | | | | | 1 | | |
| G9 | | | | | | | | | | 1 | |
| G10 | | | | | | | | | | | 1 |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | | | | 0.1 | | | | |
| G3 | | | | 1 | | | | | | | |
| G4 | | | | | 1 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | | | 1 | | | | |
| G7 | | | | | | | | 1 | | | |
| G8 | | | | | | | | | 1 | 0.2 | |
| G9 | | | | | | | | | | 0.8 | |
| G10 | | | | | | | | | | | 1 |

| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G0 | 1 | | | | | | | | | | |
| G1 | | 1 | | | | | | | | | |
| G2 | | | 1 | | | | | | | | |
| G3 | | | | 1 | | | | | | | |
| G4 | | | | | 1 | | | | | | |
| G5 | | | | | | 1 | | | | | |
| G6 | | | | | | | 1 | | | | |
| G7 | | | | | | | | 1 | | | |
| G8 | | | | | | | | | 1 | | |
| G9 | | | | | | | | | | 1 | 0.3 |
| G10 | | | | | | | | | | | 0.7 |

*Hình 3-5 Độ chính xác của hệ thống qua phương pháp đánh giá trực quan của từng người sử dụng*

Trung bình kết quả đánh giá ở phương pháp thứ 2:

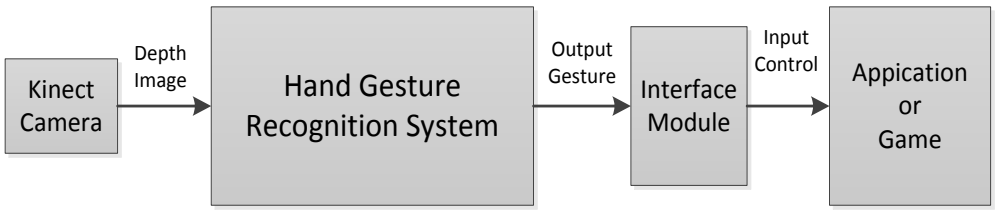
| | G0 | G1 | G2 | G3 | G4 | G5 | G6 | G7 | G8 | G9 | G10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **G0** | 1 | | | | | | | | | | 0.01 |
| **G1** | | 1 | | | | | | | | | |
| **G2** | | | 1 | 0.01 | | | 0.02 | | | | |
| **G3** | | | | 0.99 | | | 0.02 | 0.02 | | | |
| **G4** | | | | | 0.97 | | | | | | |
| **G5** | | | | | | 1 | | | | | |
| **G6** | | | | | 0.01 | | 0.95 | | | | |
| **G7** | | | | | | | | 0.97 | | | |
| **G8** | | | | | 0.01 | | 0.01 | | 0.99 | 0.02 | |
| **G9** | | | | | 0.01 | | | 0.01 | 0.01 | 0.98 | 0.03 |
| **G10** | | | | | | | | | | | 0.96 |

*Hình 3-6 Độ chính xác trung bình của hệ thống qua phương pháp đánh giá trực quan*

Hệ thống cho phép nhận dạng cử chỉ tay với độ chính xác trên 95% trong thời gian thực quan sát.

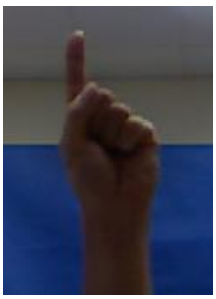
## 3.3 Ứng dụng dựa trên hệ thống cử chỉ tay

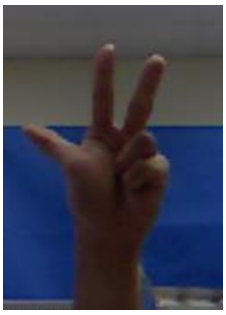
Dựa vào những cử chỉ có được qua hệ thống nhận dạng, chúng tôi sử dụng những cử chỉ đó làm điều khiển ngõ vào cho các ứng dụng hoặc trò chơi trên máy tính. Vì giới hạn về thời gian thực hiện, chúng tôi chỉ hiện thực 1 ứng dụng đơn giản búa, kéo, giấy.



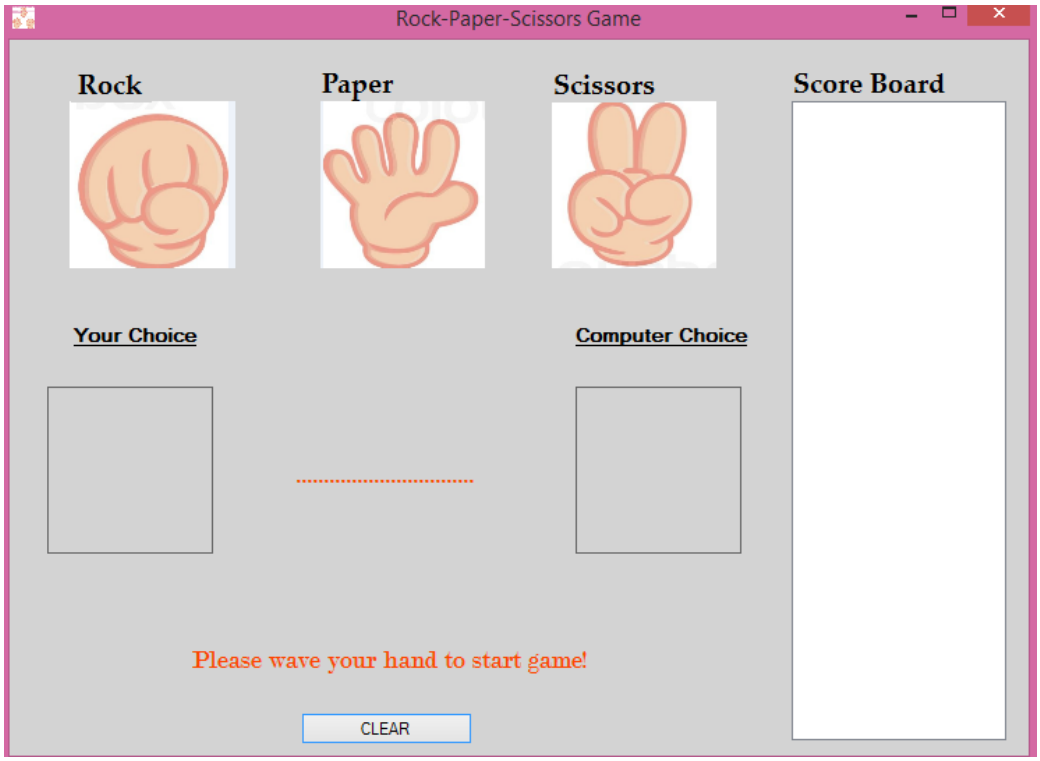

*Hình 3-7 Hệ thống điều khiển trò chơi hoặc ứng dụng trên máy tính sử dụng Kinect*

Trong luận văn này chúng tôi thử nghiệm hệ thống nhận dạng cử chỉ tay trên trò chơi "búa, kéo, giấy", trò chơi mà cũng được thử nghiệm ở bài báo số [22], với các cử chỉ điều khiển như hình 4-8. Sau khi người dùng tùy chọn cử chỉ của mình, máy tính sẽ chọn cử chỉ của nó một cách ngẫu nhiên, và kết quả sẽ được tính toán và được hiển thỉ trên bảng điểm.

| **Cử chỉ điều khiển** | **Mô tả chức năng** | **Cử chỉ điều khiển** | **Mô tả chức năng** |
|---|---|---|---|
| | Bắt đầu trò chơi | | Búa |
|  | Bắt đầu trò chơi mới (tiếp theo) | | Kéo |

| | | | |
|---|---|---|---|
|  | Xóa kết quả | | Giấy |

*Hình 3-8 Bảng điều khiển cho trò chơi "búa, kéo, giấy"*




*Hình 3-9 Trò chơi "búa, kéo, giấy" sử dụng cử chỉ tay*

Trò chơi được chúng tôi kiểm tra trên 5 người, kết quả thông qua quan sát việc trải nghiệm của người dùng, dường như không có sự sai sót về các cử chỉ nào. Người dùng trải nghiệm ứng dụng một cách thoải mái và tự nhiên.

# CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1 Kết luận

Trong luận văn này chúng tôi đã phát triển một phương pháp nhận dạng cử chỉ tay hiệu quả, chính xác trong thời gian thực sử dụng thông tin độ sâu từ cảm biến Kinect. Người dùng có thể tương tác với ứng dụng và trò chơi trên máy tính bằng cách hiện thực cử chỉ tay thay vì sử dụng chuột hay bàn phím. Điều này mang tới một trải nghiệm thú vị cho người sử dụng.

Hệ thống chúng tôi đề xuất được phân lớp qua ba bước chính: trích ảnh, tìm vec tơ đặc trưng và nhận dạng sử dụng SVM. Bước đầu tiên, từ ảnh độ sâu thu được từ Kinect, chúng tôi tách vùng bàn tay một cách dễ dàng sử dụng thư viện OPEN2/NITE2. Sau đó các đặc trưng như phương bàn tay, bán kính, tâm lòng bàn tay và các đỉnh ngón tay được tìm ra. Dựa vào những đặc trưng đó chúng tôi xây dựng một vec tơ đặc trưng bao gồm đặc trưng về khoảng cách và góc biểu diễn cho một cử chỉ. Cuối cùng chúng tôi áp dụng SVM cho việc nhận dạng các cử chỉ khác nhau

Phương pháp của chúng tôi đề xuất tương đối đơn giản nhưng vẫn mang tới chính xác cao trong việc nhận dạng, độ chính xác của việc nhận dạng đạt trên 98% trong thời gian thực quan sát trong một số giới hạn về khoảng cách và cử chỉ. Hệ thống chúng tôi có thể hoạt động trong bóng tối, không phụ thuộc vào môi trường ánh sáng và nhận dạng một cách đáng tin cậy.

## 4.2 Hướng phát triển

Độ chính xác có thể nâng cao nếu cải thiện tốt hơn trong các quá trình tiền xử lý, trích chọn đặc trưng và bổ sung bộ dữ liệu huấn luyện đa dạng hơn. Áp dụng thuật toán huấn luyện khác như Neural-Network (NN) hoặc Hidden-Markov Model (HMM) có thể cho kết quả tốt hơn SVM. Hệ thống có thể mở rộng tập cử chỉ bằng cách lấy quy định các cử chỉ khác với tập 11 cử chỉ trên, sau đó lấy mẫu và huấn luyện.

Về việc pháp triển ứng dụng, hiện tại ứng dụng trên máy tính cần lập trình chung với hệ thống nhận dạng cử chỉ tay mới có thể áp dụng cử chỉ tay được. Vì thế

để mở rộng cho hệ thống nhận dạng cử chỉ có thể tương tác với các ứng dụng bên thứ ba cần sử dụng bộ thư viện TUIO để làm cầu nối cho các ứng dụng độc lập trên Windows có thể tương tác với nhau. Khi sử dụng bộ thư viện này, bất kì ứng dụng nào cũng có thể sử dụng cử chỉ tay để điều khiển.

Thư viện OPENNI/OPENCV hỗ trợ cho việc phát triển trên ARM-LINUX, nên việc phát triển hệ thống trên nền ARM hoàn toàn có thể hiện thực được. Với việc triển khai trên ARM, các ứng dụng cho cử chỉ tay sẽ linh động hơn, có thể mở rộng các để tài như điều khiển nhà thông minh, thang máy…

## Tài liệu tham khảo

[1] Mokhtar M. Hasan and Pramoud K. Misra, "Brightness Factor Matching for Gesture Recognition System using Scaled Normalization," *International Journal of Computer Science & Information Technology (IJCSIT),* vol. 3, no.2, 2011.

[2] Mokhtar M. Hasan and Pramoud K. Misra, "Robust Gesture Recognition Using Gaussian Distribution for Features Fitting," *International Journal of Machine Learning and Computing,* vol. 2, No. 3, June 2012.

[3] Parul Chaudhary, Hardeep Singh Ryait, "Neural Network Based Static Sign Gesture Recognition System," *International Journal of Innovative Research in Computer and Communication Engineering,* February 2014 .

[4] Luigi Lamberti, Francesco Camastra, "Real-Time Hand Gesture Recognition using a Color Glove," *Springer Proceedings of the 16th international conference on Image analysis and processing:,* 2011.

[5] Robert Y. Wang and Jovan Popovi´c, "Real-Time Hand-Tracking with a Color Glove," *ACM Transactions on Graphics 28(3),* Article 63, 2009.

[6] Lale Akarun *et al.*, "Real Time Hand Pose Estimation using Depth Sensors," *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on,* Nov. 2011.

[7] Z. Ren, J. Yuan, and Z. Zhang, "Robust Hand Gesture Recognition Based on FingerEarth Mover's Distance with a Commodity Depth Camera," *In Proc. of ACM Conference on Multimedia,* vol. ACM, pp. 1093 - 109, 2011.

[8] Fabio Domini, Mauro Donadeo, Giulio Marin, "Hand Gesture Recognition with Depth Data," *ARTEMIS '13 Proceedings of the 4th ACM/IEEE international workshop on Analysis and retrieval of tracked events and motion in imagery stream,* pp. 9-16, 2013.

[9] Abidatul Izzah and Nanik Suciati, "Translation of Sign Language using Generic Fourier Descriptor and Nearest Neighbour," *International Journal on Cybernetics & Informatics ( IJCI),* vol. 3, No.1, February 2014.

[10] Zhou Ren, Jingjing Meng, Junsong Yuan, "Depth Camera Based Hand Gesture Recognition and its Applications in Human-Computer-Interaction," *Information, Communications and Signal Processing (ICICS) 2011 8th International Conference on,* pp. 1 - 5, Dec. 2011.

[11] Yan Wen, Chuanyan Hu, Guanghui Yu, Changbo Wang, "A Robust Method of Detecting Hand Gestures Using Depth Sensors," *Haptic Audio Visual Environments and Games (HAVE), 2012 IEEE International Workshop on,* 2012.

[12] Li, Yi., "Hand gesture recognition using Kinect," *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on,* 2012.

[13] Suzuki, S. and Abe, K., , "Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46," 1985.

[14] K. Stergios Poularakis, "Finger Detection and Hand Posture Recognition Based on Depth Information," *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP),* 2014.

[15] Jagdish L. Raheja *et al.*, "Tracking of Fingertips and Centers of Palm Using KINECT," *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on,* 2011.

[16] Arun Kulshreshth, Chris Zorn, Joseph J. LaViola Jr., "Real-time Markerless Kinect based Finger Tracking and Hand Gesture Recognition for HCI," *IEEE Symposium on 3D User Interfaces 2013,* 2013.

[17] W.-c. Chen, "Real-Time Palm Tracking and Hand Gesture Estimation," *Master Thesis,* 2011.

[18] Seung Il Han *et al.*, "Vision Based hand tracking for Interaction," *http://kowon.dongseo.ac.kr,* 2008.

[19] Thiago R. Trigo and Sergio Roberto M. Pellegrino, "An Analysis of Features for Hand-Gesture Classification," *IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing,* 2010 .

[20] Hui-Shyong Yeo *et al.*, "Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware," *Springer Science+Business Media New York 2013,* 2013.

[21] J. Kilian, "Simple Image Analysis By Moments," March 15, 2001.

[22] Zhou Ren *et al.*, "Robust Hand Gesture Recognition with Kinect Sensor," *In Proc. of ACM MM,* 2011.