

NHẬN DẠNG CỬ CHỈ CỦA BÀN TAY NGƯỜI THEO THỜI GIAN THỰC

TRẦN NGUYỄN NGỌC

Khoa Công nghệ thông tin, Học viện Kỹ thuật Quân sự; Email: tnn1999@mail.ru

Tóm tắt. Bài báo trình bày một số kết quả nhận dạng cử chỉ của bàn tay người theo thời gian thực sử dụng thông tin thu được từ cảm biến Kinect của hãng Microsoft. Một số kết quả chính của hướng nghiên cứu được trình bày như: kỹ thuật tách vùng bàn tay, nhận dạng tư thế của bàn tay, đề xuất thuật toán hiệu chỉnh kết quả nhận dạng từ chuỗi các tư thế. Kết quả nhận dạng cho độ chính xác khả quan (trên 93%) tạo tiền đề cho các ứng dụng tương tác người máy theo thời gian thực.

Từ khóa. Kinect, HLAC, nhận dạng, cử chỉ bàn tay.

Abstract. This paper presents a novel hand gesture recognition method using Microsoft's Kinect in real-time. Our system includes detecting and recognizing hand gestures via combining shape, local auto-correlation information and multi-class support vector machine (SVM). Our evaluation shows that the system recognizes one-handed gestures with more than 93% accuracy in real-time. The efficiency of system execution is very satisfactory, and we are encouraged to develop a natural human-machine interaction in the near future.

Key words. Kinect, HLAC, hand gesture, recognition.

1. MỞ ĐẦU

Cùng với tiến bộ của công nghệ hiện nay, giao tiếp giữa con người với máy tính đang trở nên tự nhiên, đồng thời gần hơn với giao tiếp giữa con người và con người. Một trong các phương pháp giao tiếp tự nhiên giữa người với người là dựa trên ngôn ngữ cơ thể, cụ thể là cử chỉ của tay. Xu hướng cho phép máy tính nhận dạng cử chỉ bàn tay người theo thời gian thực đã xuất hiện trên một số sản phẩm thương mại như tivi thông minh Samsung, cảm biến Kinect cho máy chơi game Xbox360 của Microsoft, DepthSense của SoftKinect, camera Creative Interactive Gesture của Intel, tivi tích hợp nhận dạng khuôn mặt và bàn tay của hãng Omron, hay thiết bị của Leap Motion.

Theo khảo sát trong nghiên cứu [1], đa phần các nghiên cứu trước đây về tương tác người - máy sử dụng các giải thuật xử lý ảnh đều tập trung khai thác chuyển động của cơ thể hoặc toàn bộ phần tay người để nhận dạng. Các nghiên cứu phân tích chi tiết cấu trúc và cử chỉ của duy nhất bàn tay [2, 3, 9, 12] để triển khai thành các sản phẩm ứng dụng thực tế gần đây đang có xu hướng phát triển mạnh. Vì vậy, bài báo trình bày các kết quả nghiên cứu về nhận dạng cử chỉ của bàn tay người với trọng tâm khai thác đặc điểm chi tiết của bàn tay. Bài báo được bố cục như sau: Mục 2 giới thiệu về phương thức khai thác thông tin từ cảm biến Kinect của hãng Microsoft phục vụ cho bài toán nhận dạng cử chỉ bàn tay. Mục 3 đề xuất giải pháp phát hiện chính xác vùng bàn tay. Mục 4 trình bày thuật toán nhận dạng với các nội dung về

trích chọn đặc trưng, nhận dạng cử chỉ. Mục 5 phân tích và so sánh các kết quả thực nghiệm với các công bố trên thế giới gần đây.

2. LỰA CHỌN THIẾT BỊ THU NHẬN THÔNG TIN BÀN TAY

Với bài toán nhận dạng cử chỉ của bàn tay người, trước đây, các nghiên cứu sử dụng các thiết bị cảm biến khác nhau như camera thông thường, cảm biến hồng ngoại, laser hay găng tay chuyên dụng (xem Hình 1).



Hình 1. Minh họa về cảm biến tổ hợp và găng tay chuyên dụng

Sử dụng găng tay chuyên dụng hoặc cảm biến tổ hợp có thể cho phép xác định đến vị trí của từng ngón tay, tuy nhiên khi sử dụng sản phẩm nhận dạng người dùng không thể sử dụng bàn tay trần tự do để tương tác. Vì thế, các dạng cảm biến loại này được chú trọng phát triển cho các ứng dụng chỉ đòi hỏi độ chính xác cao, nhưng không yêu cầu khả năng tiện ích cho người dùng.

Với các camera thông thường (ví dụ camera tích hợp sẵn với máy tính xách tay) độ chính xác thường bị ảnh hưởng bởi các điều kiện khách quan như ánh sáng, ảnh nền phía sau bàn tay, tốc độ chuyển động của bàn tay..., vì thế, chỉ giới hạn trong phạm vi các ứng dụng yêu cầu số lượng cử chỉ có hạn và các tư thế đơn giản (chủ yếu là các cử chỉ của cả bàn tay và các chuyển động đơn giản lên xuống, sang trái, sang phải...).



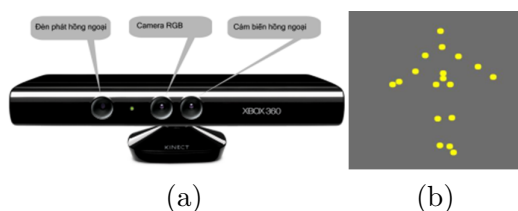
Hình 2. Smart Tivi, Camera Creative và Leap motion Controller

Với một số loại camera hồng ngoại khác mới xuất hiện trên thị trường công nghệ trong nửa cuối năm 2012 như camera Creative của Intel hay cảm biến Leap motion chuyên dụng cho nhận dạng cử chỉ của bàn tay trần với độ chính xác cao thì lại yêu cầu khoảng cách tương tác khá nhỏ (khoảng 150mm đến 900mm tới thiết bị)(xem Hình 2).

Từ những phân tích đã nêu, với mục tiêu hướng tới các ứng dụng có tính tiện ích và đảm bảo khả năng tương tác tự do của bàn tay người, ta sử dụng cảm biến Kinect để thu nhận tín hiệu với nhiều thông tin hơn camera thông thường nhưng hạn chế hơn so với găng tay chuyên dụng và cảm biến tổ hợp laser+ hồng ngoại, mặt khác, cho phép khoảng cách tương tác lên đến 1500mm – 2000mm để giải quyết bài toán nhận dạng cử chỉ của bàn tay người. Trong trường hợp này, các thuật toán xử lý ảnh và nhận dạng đề xuất sẽ phải tập trung giải quyết bổ sung phần thông tin chi tiết cấu trúc của đối tượng, từ đó hướng tới mục tiêu nhận dạng với độ tin cậy gần tương đương với các thiết bị chuyên dụng gắn kèm với tay người.

Cảm biến Kinect thực chất là một hệ thống các camera thường và hồng ngoại, ngoài ra được tích hợp sẵn một số thuật toán nhận dạng do hãng Microsoft phát triển cho phép sử

dùng 3 luồng thông tin cơ bản về đối tượng là người đứng trước camera gồm: ảnh màu thông thường (RGB image); ảnh thông tin chiều sâu (Depth image) cho phép ước lượng khoảng cách từ các vùng trên ảnh đối tượng tới thiết bị Kinect; khung xương mô hình bám theo 20 khớp cơ bản trên cơ thể người (Skeleton Tracking).



Hình 3. Cảm biến kinect (a) và khung xương mô hình (b)

Khi sử dụng luồng thông tin Skeleton Tracking có thể cho phép xác định gần đúng vị trí tâm bàn tay, tuy nhiên việc khoanh vùng chính xác để biết được cấu trúc bàn tay thì cần bổ sung thêm các thao tác xử lý ảnh phù hợp. Do vậy, trong nội dung tiếp theo bài báo đề xuất giải thuật nhận dạng cử chỉ của bàn tay sử dụng các luồng thông tin về Depth image và Skeleton Tracking của cảm biến Kinect.

3. THUẬT TOÁN PHÁT HIỆN VÙNG BÀN TAY

Các thuật toán phát hiện vùng bàn tay trước đây dựa trên thông tin hình ảnh thường sử dụng một số kỹ thuật xử lý ảnh như: phân vùng màu da dựa trên các ngưỡng màu sắc [4]; phát hiện dựa trên đặc trưng chênh lệch mức xám Haar-like [5]; khảo sát phân bố mức xám HOG (Histogram of Oriented Gradients) [6]. Tuy nhiên, như đã nêu trong Mục 2, cảm biến Kinect cho phép xác định vị trí gần đúng tâm bàn tay, vì vậy để phát hiện vùng bàn tay, trong nghiên cứu này đề xuất tiến hành phân vùng ảnh chiều sâu Depth image xung quanh tâm bàn tay để xác định vùng đối tượng theo các bước sau:

Bước 1. Xác định thô vùng bàn tay, sử dụng đánh giá về tương quan khoảng cách

$$\text{hand } I_{x,y} = (\text{dist}(x, y, x_0, y_0) < \rho) \wedge (|\text{depth}I_{x,y} - \text{depth}I_{x_0,y_0}| < \theta), \quad (1)$$

trong đó, x_0, y_0 là tọa độ tâm bàn tay được trả về bởi luồng thông tin từ cảm biến Kinect;

x, y là tọa độ một điểm bất kỳ thuộc vùng bàn tay;

$\text{hand } I_{x,y}$ xác định trạng thái điểm ảnh có tọa độ (x, y) trên ảnh có thuộc vùng bàn tay hay không;

$$\text{dist}(x, y, x_0, y_0) = \sqrt{(x - x_0)^2 + (y - y_0)^2};$$

$\text{depth}I_{x,y}$ là giá trị tại điểm ảnh có tọa độ (x, y) trên ảnh Depth image, cho phép xác định khoảng cách tới cảm biến;

ρ, θ là các ngưỡng xác định theo cấu hình thiết bị cảm biến và cấu trúc sinh học của khung xương (thu được từ luồng Skeleton Tracking).

Cụ thể trong nghiên cứu này lựa chọn $\rho = 0.8h_r$; $\theta = 150$ (dành cho thiết bị Kinect Xbox360), giá trị h_r đặc trưng cho bán kính vùng bàn tay được xác định thông qua các giá trị về độ rộng của vai người d_{shoulder} và giá trị chiều sâu tại tâm bàn tay $\text{depth}I_{x_0,y_0}$ trên ảnh Depth image và điểm giữa hai vai $\text{depth}I_{\text{shoulder_center}}$ mà luồng Skeleton Tracking cung cấp

như sau

$$h_r = \frac{\text{depth}I_{\text{sholder_center}} \cdot d_{\text{sholder}}}{\text{depth}I_{x_0,y_0}}$$

Bước 2. Thực hiện tiền xử lý ảnh vùng bàn tay thông qua một bộ lọc trung vị, tiếp đó phát hiện đường bao (coutour) của vùng bàn tay, loại bỏ các vùng có diện tích nhỏ và định dạng lại vùng bàn tay. Trong bước này, nghiên cứu đề xuất sử dụng các thao tác phát hiện đường bao và vùng liên kết có sẵn trong bộ thư viện mã nguồn mở do hãng Intel khởi xướng phát triển OpenCV [14].



Hình 4. Xử lý phát hiện vùng bàn tay

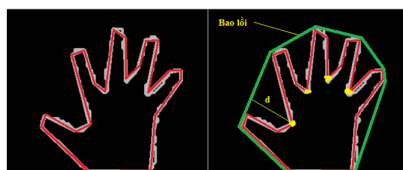
4. THUẬT TOÁN NHẬN DẠNG CỬ CHỈ

Để tiến hành nhận cử chỉ của bàn tay, trước hết cần tiến hành trích chọn đặc trưng vùng thu được ở phần phát hiện đối tượng. Các nghiên cứu trước đây cho thấy, có thể sử dụng các kỹ thuật phân tích hình dạng đối tượng như: sử dụng các dạng moment [10], sử dụng bộ lọc Gabor và trích chọn PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis) [11] hoặc sử dụng đơn thuần các kỹ thuật trích chọn đặc trưng dựa trên phân bố mức xám [12]... tuy nhiên, qua khảo sát cho thấy nếu chỉ sử dụng các đặc trưng về đường bao thì cơ bản chỉ nhận được một số cử chỉ có hình dạng tương đối tương minh. Khi kích thước bàn tay thu được nhỏ, các kẽ tay sẽ khó phân biệt, lúc này cần bổ sung thêm một số đặc trưng dạng thống kê. Vì vậy, trong nghiên cứu này sử dụng kỹ thuật xấp xỉ đa giác cho vùng đường bao và tìm bao lồi nhỏ nhất chứa vùng bàn tay, tiếp đó trích chọn các đặc trưng theo đa giác, sau đó sử dụng một dạng đặc trưng thống kê dựa trên mức độ tự tương quan về cường độ sáng và có khả năng bất biến với một số phép biến đổi hình học như xoay, tịnh tiến, co giãn.

Trước hết, để phân biệt các ngón tay, ta tiến hành phân tích độ sâu các kẽ tay, đồng thời loại bỏ các kẽ không phù hợp về độ sâu (xem Hình 5). Trong đó, các thao tác về xấp xỉ đa giác và tìm bao lồi được nhóm tác giả sử dụng thuật toán Douglas–Peucker tích hợp sẵn trong bộ thư viện OpenCV.

Vùng bàn tay sau khi được chuẩn hóa kích thước thì các đặc trưng hình dạng cho bàn tay được xác định gồm các thành phần sau: f_1 số lượng kẽ tay; f_2 độ sâu trung bình của các kẽ tay; f_3 diện tích vùng bàn tay, f_4 giá trị của tâm bàn tay; f_5 khoảng cách trung bình giữa hai kẽ tay liên tiếp; f_6 khoảng cách xa nhất giữa hai kẽ tay; f_7 số đỉnh bao lồi; f_8 độ dài cạnh lớn nhất của bao lồi, f_9 độ dài cạnh nhỏ nhất của bao lồi; f_{10} tỷ lệ giữa chiều rộng và chiều cao của hình chữ nhật chứa vùng bàn tay.

Tiếp đó, để cải thiện chất lượng nhận dạng, đặc biệt để có thể nhận dạng tốt khi bàn tay xuất hiện ở các hướng xoay khác nhau, ta bổ sung thêm đặc trưng mức độ tự tương quan HLAC (Higher Order Local Autocorrelation), theo các nghiên cứu [7, 8] HLAC cũng có khả năng bất biến với các phép co giãn và xoay đối tượng, hơn nữa còn cho phép có thể mở rộng hơn nữa số lượng cử chỉ có thể nhận dạng.



Hình 5. Trích chọn đặc trưng hình dạng bàn tay thông qua các kẽ tay

Công thức xác định độ tự tương quan HLAC bậc n được xác định như sau [7]

$$F^n(a_1, a_2, \dots, a_n) = \int I(r)I(r + a_1)\dots I(r + a_n)dr, \quad (2)$$

trong đó,

n là bậc tự tương quan;

r là tọa độ của điểm ảnh;

a là độ dịch chuyển theo các hướng;

$I(\cdot)$ là cường độ sáng tại các điểm ảnh.

Lưu ý rằng, thay vì tính toán trực tiếp dựa trên cường độ sáng tại các điểm ảnh ta tiến hành tính toán dựa trên ảnh bàn tay sau khi đã phát hiện biên. Khi đó quá trình tính tích cường độ sáng chuyển thành phép AND logic dựa trên trạng thái điểm ảnh có phải là biên hay không. Để khối lượng tính toán không quá lớn sẽ chỉ tính đến độ tự tương quan bậc 3, với cửa sổ 3×3 di trên toàn bộ ảnh.

Để huấn luyện cho nhận dạng tư thế tĩnh của bàn tay có thể sử dụng công cụ SVM (Support Vector Machine) với hàm hạt nhân cơ bản Gaussian RBF (kết quả nhận dạng chi tiết được trình bày ở Mục 5). Kết thúc giai đoạn này mỗi tư thế của bàn tay có thể xác định bởi hàm phân loại từ 45 đặc trưng

$$h^{SVM} = G(f_1, f_2, \dots, f_{10}, f_{11}, \dots, f_{45}). \quad (3)$$

Do trong thực tế tương tác, thiết bị Kinect có thể cung cấp thông tin hình ảnh với tốc độ cao (khoảng 30 hình/giây) nên để tăng độ tin cậy của kết quả nhận dạng, cần thực hiện việc so sánh trên nhiều khung hình liên tiếp, tiếp đó sử dụng các đánh giá xác suất để xác định cử chỉ tay phù hợp nhất.

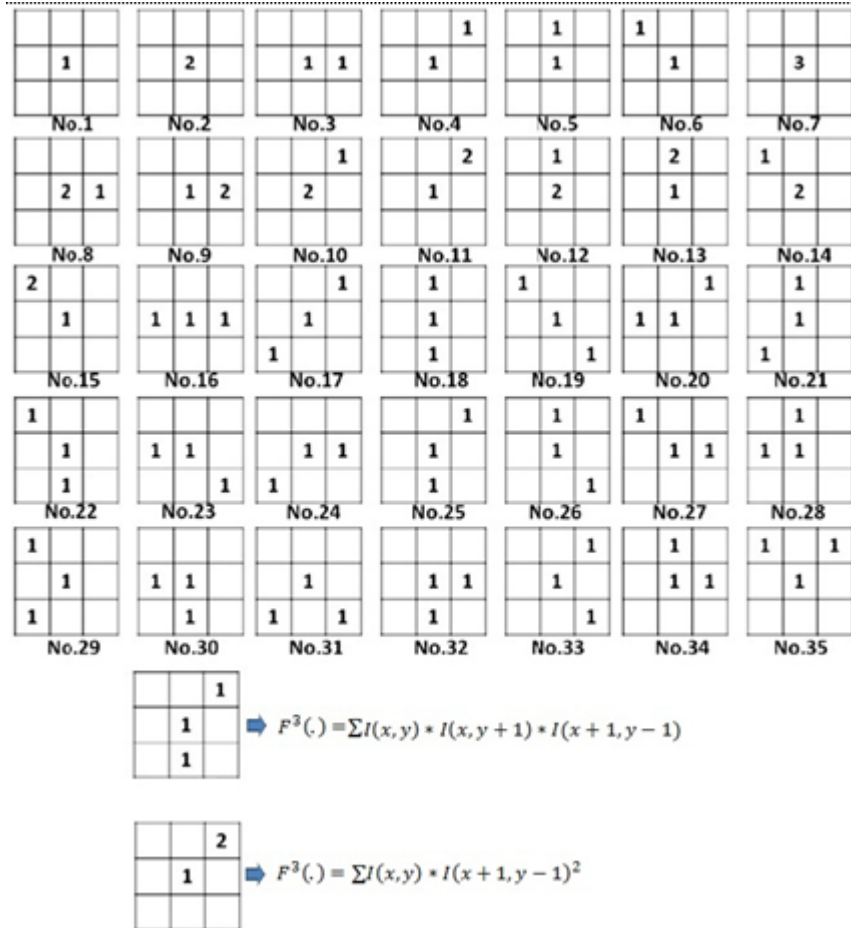
Xét tập hợp N khung hình liên tiếp, giả sử khi áp dụng kỹ thuật học máy (3) sẽ cho ra N kết quả so sánh: (K_1, K_2, \dots, K_N) ; trong đó K_i là kết quả thu được từ khung hình thứ i , là một trong các cử chỉ tay trong tập hợp các cử chỉ tay của bộ hình ảnh đã huấn luyện. Khi đó, nếu bộ hình ảnh huấn luyện có M cử chỉ tay, gồm $C = (C_1, C_2, \dots, C_M)$, thì $K_i \in C$.

Việc đề xuất chọn cử chỉ C_j thuộc C là phù hợp nhất, trong số tập hợp mọi cử chỉ đã biết, là cử chỉ thỏa mãn tổng $S_j = P(C_j, K_1) + P(C_j, K_2) + \dots + P(C_j, K_N)$ có giá trị lớn nhất, tức là

$$C_j = \operatorname{argmax}_{C_i \in C} \sum_{t=1}^N P(C_i, K_t), \quad (4)$$

với $P(C_j, K_i)$ là xác suất kết quả so sánh 01 khung hình cho ra kết luận K_i khi người dùng thực hiện cử chỉ C_j , và tất cả $M \times M$ giá trị gần đúng của $P(C_j, K_i)$ đã được xác định qua thực nghiệm và lưu trữ sẵn. Bảng các giá trị (xem Bảng 3) này được xây dựng thông qua

chính quá trình đánh giá mức độ “nhầm lẫn” khi nhận dạng một cử chỉ của bàn tay, tức là thông qua việc sử dụng khái niệm ma trận nhập nhằng (Confusion Matrix [13]) mà các nghiên cứu về nhận dạng trước đây thường chỉ dùng để khảo sát tính hiệu quả của thuật toán.



Hình 6. Cách trích chọn đặc trưng HLAC theo bậc 3

5. KẾT QUẢ THỰC NGHIỆM VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

Quá trình thực nghiệm được tiến hành theo các giai đoạn như sau:







Giai đoạn I. Chuẩn bị bộ dữ liệu huấn luyện các cử chỉ của bàn tay theo các tư thế tự nhiên (SH01) là các ảnh vùng bàn tay đã được cắt riêng (xem Bảng 1); Tiến hành đánh giá kết quả nhận dạng 6 cử chỉ này theo các tiêu chí: đặc trưng ROC (Receiver Operating Characteristic), chỉ số AUC (Area Under Curve), khảo sát khả năng nhận dạng đúng của thuật toán nhận dạng sử dụng giải pháp học máy (3).

Giai đoạn II. Sử dụng kết quả thu được ở giai đoạn I, tiến hành làm thực nghiệm trước thiết bị Kinect thực và xây dựng bảng xác suất nhầm lẫn.

Giai đoạn III. Sử dụng kết quả hiệu chỉnh ở giai đoạn II với việc sử dụng $N = 8$ khung hình liên tiếp để đánh giá lại kết quả nhận dạng thực tế với cùng điều kiện thực nghiệm để tạo tập

mẫu SH01. Kết quả nhận dạng của giai đoạn I được phản ánh trong Bảng 2, ngoài ra, kết quả

Bảng 1. Thông tin về dữ liệu huấn luyện SH01

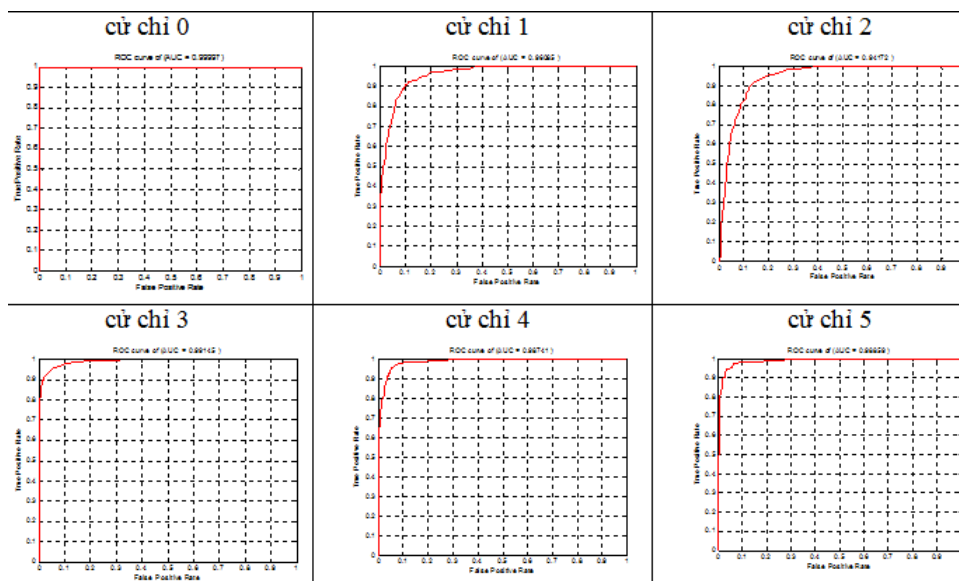
STT	Dạng ảnh mô tả cử chỉ	Định danh cử chỉ	Số lượng ảnh trong tập huấn luyện	Số lượng ảnh trong tập kiểm thử
1		0	500	200
2		1	500	200
3		2	500	200
4		3	500	200
5		4	500	200
6		5	500	200

phân tích dựa trên đồ thị ROC (xem Hình 7) và chỉ số AUC trung bình đạt trên 0.94.

Bảng 2. Kết quả nhận dạng cử chỉ của bàn tay trên tập SH01

Cử chỉ thực tế	Cử chỉ đoán nhận						Tỷ lệ nhận dạng đúng
	0	1	2	3	4	5	
0	191	9	0	0	0	0	95,50%
1	0	199	1	0	0	0	99,50%
2	0	0	199	1	0	0	99,50%
3	0	0	13	175	0	0	93,09%
4	0	0	0	4	194	2	97,00%
5	0	0	3	7	47	143	71,50%

Ngoài ra kết quả phân tích dựa trên đồ thị ROC (Receiver Operating Characteristic) và chỉ số AUC (Area Under Curve) trung bình trên 0.94.



Hình 7. Đồ thị ROC và chỉ số AUC với từng kết quả nhận dạng

Kết quả giai đoạn II trong Bảng 3 phản ánh kết quả nhận dạng đúng cho mỗi cử chỉ khi thao tác nhanh trực tiếp trước thiết bị khi giải pháp học máy (3) đạt xấp xỉ 86%.

Kết quả nhận dạng sau khi sử dụng hiệu chỉnh xác suất (4) dựa trên Bảng 3 của giai đoạn III cho thấy độ chính xác được cải thiện đạt trung bình 93% khi tương tác trực tiếp với thiết bị, đánh giá dựa trên 50 lần nhận dạng liên tiếp mỗi cử chỉ (xem Bảng 4). Bên cạnh đó, kết quả so sánh cho thấy độ chính xác khi tương tác trực tiếp theo thời gian thực của thuật toán

Bảng 3. Xác suất nhầm lẫn $P(C_i, K_t)$ giữa các cử chỉ

Cử chỉ thực tế	Cử chỉ đoán nhận						Số mẫu thực nghiệm
	0	1	2	3	4	5	
0	0.9311	0.0654	0.0032	0.0004	0.0000	0.0000	2800
1	0.1021	0.8857	0.0100	0.0021	0.0000	0.0000	2800
2	0.0279	0.0732	0.8546	0.0443	0.0000	0.0000	2800
3	0.0946	0.0021	0.0707	0.8107	0.0218	0.0000	2800
4	0.0068	0.0143	0.0046	0.1307	0.8350	0.0086	2800
5	0.0018	0.0000	0.0000	0.0168	0.1125	0.8689	2800

đã đề xuất tương đương với kết quả nhận dạng trên tập ảnh tĩnh được chuẩn bị trong phòng thí nghiệm mà nghiên cứu [10] thử nghiệm.

Bảng 4. Kết quả nhận dạng khi tương tác trực tiếp

Cử chỉ thực tế	Cử chỉ đoán nhận						Số lượt khoanh đúng vùng bàn tay	Tỷ lệ phát hiện đúng vùng bàn tay	Tỷ lệ nhận dạng đúng theo tương tác thời gian thực	Tỷ lệ nhận dạng đúng với ảnh tĩnh công bố trong [10]
	0	1	2	3	4	5				
0	48	1	1	0	0	0	49	98%	96%	93,3%
1	1	47	2	0	0	0	49	98%	94%	90%
2	3	3	44	0	0	0	48	96%	88%	90%
3	0	0	1	44	4	1	49	98%	88%	93,3%
4	0	0	0	2	46	2	50	100%	92%	96,7%
5	0	0	1	1	1	47	49	98%	94%	96,7%

Ngoài ra, khi so sánh về độ chính xác của giai đoạn phát hiện vùng bàn tay thì việc sử dụng đầy đủ các luồng thông tin của Kinect mà bài báo đề xuất cho phép đạt kết quả vượt trội (98%) so với chỉ sử dụng thông tin về chiều sâu (90,2%) như trong [12].

Kết quả thống kê trên máy tính xách tay với bộ vi xử lý Intel core i3 CPU 1.7Hz cho thấy: thời gian nhận dạng trung bình cho mỗi cử chỉ xấp xỉ 85mili giây, tương đương 12hình/giây cho phép đáp ứng các yêu cầu xử lý theo thời gian thực.

Như vậy, bài báo đã trình bày các kết quả nghiên cứu về quá trình phát hiện và nhận dạng cử chỉ của bàn tay người theo thời gian thực với một số đóng góp chính liên quan đến cách thức trích chọn đặc trưng, giải pháp cải thiện độ chính xác của kỹ thuật học máy trong bài toán nhận dạng cử chỉ bàn tay. Trong các nghiên cứu tiếp theo, chúng tôi sẽ tiến hành xây dựng mô hình nhận dạng cử chỉ khi kết hợp thêm các yếu tố về quỹ đạo chuyển động và gia tốc, vị trí tương đối giữa hai bàn tay, cũng như giải pháp tích hợp kết quả nhận dạng với điều khiển máy tính thông qua cơ chế giả lập các thiết bị ngoại vi.

TÀI LIỆU THAM KHẢO

- [1] Song, Yale, David Demirdjian, and Randall Davis, Continuous body and hand gesture recognition for natural human-computer interaction, *ACM Transactions on Interactive Intelligent Systems (TiiS)* **2** (1) (2012) 1–28.
- [2] Rautaray, Siddharth S., and AnupamAgrawal, Vision based hand gesture recognition for human computer interaction: a survey, *Artificial Intelligence Review* **6** (11) (2012) 1–54.
- [3] J. Suarez, R. R. Murphy, Hand gesture recognition with depth images: A review, *RO-MAN, 2012 IEEE*, Paris 9-13, Sept. 2012 (411–417).

- [4] V. Vezhnevets, V. Sazonov, A. Andreeva, A survey on pixel-based skin color detection techniques, *Proc. Graphicon, Vol. 3*, Moscow, Russia, 2003 (85–92).
- [5] Mathias Klsch, Matthew Turk, Analysis of rotational robustness of hand detection with a viola-jones detector, *International Conference on Pattern Recognition (Cambridge, UK)* **3** (2004) 107–110.
- [6] E. J. Ong, R. Bowden, A boosted classifier tree for hand shape detection, Automatic Face and Gesture Recognition, *Proceedings Sixth IEEE International Conference*, Seoul, Korea, 2004 (889–894).
- [7] T. Kurita, S. Hayamizu, Gesture recognition using HLAC features of PARCOR images and HMM based recognizer, In Automatic Face and Gesture Recognition, 1998, *Proceedings Third IEEE International Conference*, Nara, Japan, 1998 (422–427).
- [8] Tetsu Matsukawa, Takio Kurita, Image representation for generic object recognition using higher-order local autocorrelation features on posterior probability images, *Pattern Recognition* **45** (2) (2012) 707–719.
- [9] Kao, Chang-Yi, and Chin-ShyurngFahn, A human-machine interaction technique: hand gesture recognition based on hidden Markov models with trajectory of hand motion, *Procedia Engineering* **15** (2011) 3739–3743.
- [10] Yun, Liu, Zhang Lifeng, and Zhang Shujun, A hand gesture recognition method based on multi-feature fusion and template matching, *Procedia Engineering* **29** (2012) 1678–1684.
- [11] Gupta, Shikha, JafreezalJaafar, and Wan Fatimah Wan Ahmad, Static hand gesture recognition using local gabor filter, *Procedia Engineering* **41** (2012) 827–832.
- [12] H. Li, L. Yang, X. Wu, Static hand gesture recognition based on hog with kinect, *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference, Vol. 1*, Nanchang, China, 2012 (271–273).
- [13] Stehman, Stephen V, Selecting and interpreting measures of thematic classification accuracy, *Remote Sensing of Environment* **62** (1) (1997) 77–89.
- [14] Bradski, Gary, and Adrian Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, Incorporated, 2008.

Ngày nhận bài 14 - 3 - 2013

Nhận lại sau sửa ngày 01 - 8 - 2013