BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

UNG NHO DÃI

NGHIÊN CỨU TRÍCH CHỌN ĐẶC TÍNH TRONG NHẬN DẠNG HÀNH ĐỘNG NGƯỜI TRONG KHÔNG GIAN 3D

LUẬN VĂN THẠC SĨ KỸ THUẬT

ĐÀ NẴNG - Năm 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC ĐÀ NẴNG

UNG NHO DÃI

NGHIÊN CỨU TRÍCH CHỌN ĐẶC TÍNH TRONG NHẬN DẠNG HÀNH ĐỘNG NGƯỜI TRONG KHÔNG GIAN 3D

Chuyên ngành: KHOA HỌC MÁY TÍNH
Mã số: 60.48.01

LUẬN VĂN THẠC SĨ KỸ THUẬT

Người hướng dẫn khoa học: TS. PHẠM MINH TUẤN

ĐÀ NẴNG - Năm 2015

LỜI CẢM ƠN

Tôi chân thành cảm ơn TS. Phạm Minh Tuấn đã tận tình hướng dẫn, chỉ bảo trong quá trình hoàn thành luận văn này.

Tôi cũng xin gửi lời cảm ơn đến các quý thầy cô giảng dạy trong chương trình cao học chuyên ngành Khoa học máy tính khóa K.27, những người đã truyền đạt cho tôi những kiến thức hữu ích làm cơ sở cho tôi thực hiện luận văn này.

LỜI CAM ĐOAN

Tôi xin cam đoan :

a. *Những nội dung trong luận văn này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của thầy giáo TS. Phạm Minh Tuấn.*

b. *Mọi tham khảo dùng trong luận văn đều được trích dẫn rõ ràng và trung thực tên tác giả, tên công trình, thời gian, địa điểm công bố.*

c. *Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.*

Tác giả

UNG NHO DÃI

MỤC LỤC

DANH MỤC CÁC CHỮ VIẾT TẮT

2D	Two Dimensional
3D	Three Dimensional
AMC	Acclaim Motion Capture
ASF	Acclaim Skeleton File
CMU	Carnegie Mellon University
DOF	Degrees of Freedom
EM	Expectation Maximization
FCM	Fuzzy C-means
FE	Feature Extraction
FS	Feature Selection
GMM	Gaussian Mixture Model
HAC	Hierarchical Agglomerative Clustering
HMM	Hidden Markov Model
k-NN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
Mocap	Motion Capture
NN	Neural Network
PCA	Principal Component Analysis
RFID	Radio Frequency Identification
RGBD	Red Green Blue Depth
SVM	Support Vector Machine
TSVM	Transductive Support Vector Machine

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC HÌNH VẼ

MỞ ĐẦU

1. Lý do chọn đề tài

1.1. Bối cảnh chung

Từ những năm 80 của thế kỷ trước, nhận dạng hành động người (human activity recognition) đã thu hút rất nhiều quan tâm, nghiên cứu của các nhà khoa học. Nó được sử dụng rộng rãi trong nhiều ứng dụng và trong các lĩnh vực khác như y học, xã hội học, giao tiếp người máy. Nhận dạng hành động được chia làm hai loại chính: loại thứ nhất sử dụng cảm biến (sensor-based) loại thứ hai sử dụng hình ảnh (vision-based).

Hoạt động nhận dạng sử dụng cảm biến kết hợp sự đa dạng của mạng lưới thiết bị cảm biến với việc khai phá dữ liệu và học máy để mô hình hoạt động của con người. Cấu hình các thiết bị di động hiện nay đủ mạnh để thu thập dữ liệu từ nhiều loại cảm biến khác nhau và xử lý các dữ liệu đó để có thể đưa ra ước lượng về năng lượng cần thiết cho các hoạt động hàng ngày của con người. Các nhà nghiên cứu tin rằng, với sự phát triển mạnh mẽ của các loại thiết bị và các lại cảm biến, việc theo dõi và nhận dạng hoạt động của con người sẽ trở nên dễ dàng hơn.

Vấn đề quan trọng và thách thức nhất đối với nhận dạng hành động là nhận biết được hành động của con người thông qua hình ảnh từ hệ thống các camera. Kỹ thuật chủ yếu được dùng để nhận dạng từ hình ảnh là thị giác máy tính (vision computer). Có rất nhiều phương pháp đã được áp dụng trong nhận dạng hành động dựa vào hình ảnh như optical flow, bộ lọc Kalman, mô hình Markov ẩn, sử dụng các dữ liệu khác nhau từ camera, sóng âm (stereo) và hồng ngoại.

Gần đây, một số nhà nghiên cứu đã sử dụng camera RGBD (Red, Green, Blue, Depth) như Kinect[1] để nhận dạng hoạt động của con người. Dữ liệu thu được từ các thiết bị chuyên dụng này là dữ liệu chuyển động 3D của cơ thể người. Những dữ liệu này sẽ là dữ liệu huấn luyện hữu ích cho các mô hình nhận dạng hành động.

Hình 0.1 Microsoft Kinect Camera

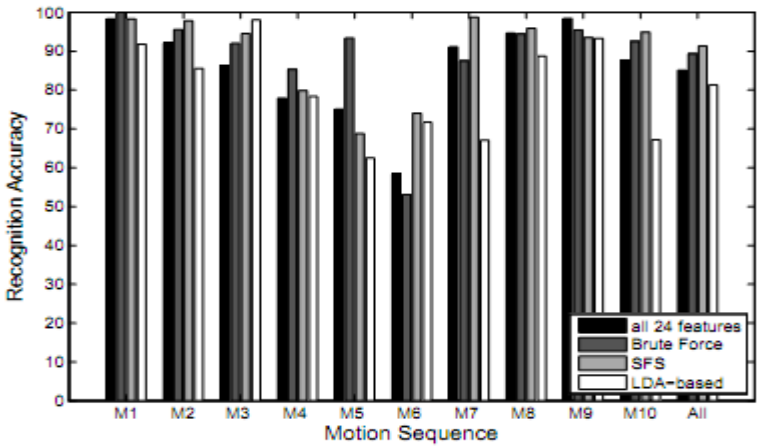
1.2. Các phương pháp trước đây

Những nghiên cứu gần đây trong lĩnh vực nhận dạng hoạt động người chủ yếu tập trung vào nghiên cứu và nhận dạng từ những video được quay bởi các camera thông dụng. Khó khăn lớn nhất đối với dữ liệu từ camera thông dụng là chỉ quay được ở một hướng, dẫn đến sự thiếu hụt dữ liệu, nếu kết hợp nhiều camera thì vẫn không đảm bảo thu được toàn bộ hoạt động, đồng thời giảm hiệu năng của quá trình nhận dạng. Mặc dù đã có rất nhiều nỗ lực trong những thập kỷ qua, lĩnh vực nhận dạng hoạt động người từ dữ liệu video vẫn còn nhiều khó khăn, thách thức.

Từ sau sự ra đời của các thiết bị cảm biến chiều sâu (depth sensor), đã có một hướng tiếp cận mới trong nhận dạng hành động người, đó là sử dụng dữ liệu chuyển động 3D. Trong 20 năm trở lại, một số phương pháp chính để thu thập dữ liệu 3D hoặc là sử hệ thống chụp chuyển động dựa vào marker[2] như

[1] Kinect là một thiết bị đầu vào, là cảm biến chuyển động do hãng Microsoft sản xuất dành cho Xbox 360 và máy tính sử dụng hệ điều hành Windows.
[2] Marker là một thiết bị đánh dấu, được gắn lên đối tượng cần theo dõi.

là Mocap[3] (Motion Capture) hoặc là dùng stereo camera[4] - chụp hình ảnh 2D từ nhiều hướng khác nhau để dựng thành mô hình 3D.



Hình 0.2 So sánh kết quả giữa các phương pháp trích chọn đặc tính

Sau khi đã thu thập được dữ liệu 3D, có rất nhiều phương pháp đã được đề xuất để hoàn thành quá trình nhận dạng. Điểm chung của các phương pháp này là cố gắng làm giảm số lượng thuộc tính của dữ liệu nhận dạng trước khi xây dựng mô hình huấn luyện. D. Gehrig [10] đã nghiên cứu, thực nghiệm trên ba phương pháp trích chọn và lựa chọn đặc tính khác nhau (Brute Force, SFS, LDA) (xem hình 0.2) và đã giảm đáng kể hiệu năng của quá trình nhận dạng so với dữ liệu ban đầu. L. Fengjun [5] phân tích và đưa ra bảy loại đặc tính khác nhau dựa vào tư thế và sự kết hợp giữa các khớp xương, sau đó dựng mô hình huấn luyện và nhận dạng dùng Markov ẩn (xem hình 0.3). K. Dana [11] đã phát triển một mô hình học tăng cường mới dựa trên mô hình Markov ẩn.

[3] Mocap (Motion Capture) là hệ thống chuyên biệt dùng để chụp chuyển động.
[4] Stereo camera là camera có hai hay nhiều ống kính với cảm biến ảnh hoặc khung phim riêng biệt cho mỗi ống kính.

action	walk	run	j upward	j forward	stand	sit	bow	lie
Exp.1	94.1%	95.5%	92.2%	91.2%	91.8%	92.4%	89.8%	88.7%
Exp.2	89.0%	91.3%	87.3%	86.6%	87.9%	90.5%	86.0%	84.8%
action	stand2sit	sit2stand	stand2bow	bow2stand	stand2lie	lie2stand	sit2lie	lie2sit
Exp.1	89.7%	89.8%	89.0%	88.3%	92.4%	88.2%	91.2%	91.8%
Exp.2	84.7%	86.6%	84.8%	86.5%	88.7%	84.5%	86.6%	86.1%
action	wave hand	point	lower arm	lift arm	nod	shake head		
Exp.1	95.8%	94.2%	92.7%	92.3%	97.9%	96.7%		
Exp.2	91.3%	92.8%	89.2%	89.4%	95.1%	94.8%		

Hình 0.3 Kết quả nhận dạng dùng bảy loại đặc tính do L. Fengjun đề xuất

1.3. Những vấn đề tồn tại

Việc xây dựng mô hình nhận dạng sử dụng dữ liệu chuyển động 3D vẫn còn nhiều điểm chưa tốt về hiệu năng cũng như chi phí. Các hạn chế đó là: dữ liệu chuyển động 3D là dữ liệu phức tạp, có số lượng thuộc tính lớn dẫn đến chi phí tính toán lớn do đó hiệu năng sẽ không cao và tỉ lệ nhận dạng đúng thấp, đặc biệt với những hoạt động phức tạp.

2. Mục tiêu và nhiệm vụ

Trước những vấn đề tồn tại phân tích ở trên, luận văn này sẽ nghiên cứu, xây dựng mô hình nhận dạng hành động người từ dữ liệu chuyển động 3D; trong đó trọng tâm là các phương pháp trích chọn và lựa chọn đặc tính nhằm làm giảm số chiều và độ lớn của dữ liệu, góp phần nâng cao độ chính xác và hiệu năng của mô hình.

3. Đối tượng và phạm vi nghiên cứu

3.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu trong luận văn này là mô hình nhận dạng hành động người, cụ thể hơn là dữ liệu chuyển động 3D trong định dạng Acclaim[5] (asf/amc) do trường đại học CMU (Carnegie Mellon University) thu thập bằng Mocap của họ; và các phương pháp trích chọn, lựa chọn đặc tính phù hợp.

3.2. Phạm vi nghiên cứu

Bộ dữ liệu chuyển động 3D do CMU cung cấp có rất nhiều hoạt động khác nhau, luận văn này chỉ nghiên cứu trên một số loại hoạt động đơn giản như đi bộ (walk), chạy (run, jog), nhảy (jump) và khiêu vũ (dance). Ngoài ra, luận văn chỉ tập trung nghiên cứu một số phương pháp trích chọn đặc tính thông dụng như PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis); và dùng máy vectơ hỗ trợ (SVM) trong học máy.

4. Phương pháp nghiên cứu

4.1. Nghiên cứu lý thuyết

Về phần lý thuyết, luận văn tập trung nghiên cứu tổng quan về nhận dạng hoạt động người trong không gian 3D, trong đó tập trung vào kỹ thuật thu thập dữ liệu chuyển động 3D bằng phương pháp sử dụng hệ thống chụp chuyển động; máy vectơ hỗ trợ (SVM) và các phương pháp trích chọn, lựa chọn đặc tính.

[5] Acclaim là tên một công ty game, đồng thời là tên một loại định dạng dữ liệu chuyển động 3D, một bộ gồm hai tệp tin asf/amc do chính công ty đó đề xuất.

4.2. Nghiên cứu thực nghiệm

Quá trình nghiên cứu thực nghiệm sử dụng ngôn ngữ lập trình C# trên nền tảng của thư viện học máy mã nguồn mở Accord.NET Framework[6] xây dựng mô hình nhận dạng với các phương pháp trích chọn đặc tính khác nhau so sánh kết quả thu được và đưa ra kết luận.

5. Ý nghĩa khoa học và thực tiễn của đề tài

5.1. Về mặt lý thuyết

Luận văn này củng cố các lý thuyết về nhận dạng hành động người trong không gian 3D, dữ liệu chuyển động 3D, các phương pháp trích chọn, lựa chọn đặc tính.

5.2. Về mặt thực tiễn

Đề xuất, xây dựng mô hình nhận dạng hành động người trong không gian 3D sử dụng các phương pháp trích chọn đặc tính và học máy.

6. Bố cục của luận văn

Ngoài phần mở đầu và kết luận, luận văn gồm có ba chương với các nội dung chính như sau:

Chương 1: Nghiên cứu tổng quan

Chương này trình bày tổng quan các vấn đề liên quan đến đề tài của luận văn. Nội dung chủ yếu xoay quanh các chủ đề chính. Hoạt động nhận dạng hành động người; mô hình chụp chuyển động; các phương pháp học máy; các phương pháp trích chọn và lựa chọn đặc tính.

---

[6] Accord.NET Framework là thư viện mã nguồn mở tổng hợp các vấn đề trong học máy (http://accord-framework.net/)

Chương 2: Giải pháp đề xuất

Chương này tập trung vào trình bày và giải thích chi tết mô hình nhận dạng đề xuất và các thành phần trong mô hình.

Chương 3: Thực nghiệm và đánh giá kết quả

Chương này trình bày chi tết quá trình thực nghiệm bao gồm môi trường thực nghiệm, các giai đoạn thực nghiệm và kết quả thực nghiệm qua từng giai đoạn, từ đó đưa ra các nhận xét, đánh giá.

CHƯƠNG 1
NGHIÊN CỨU TỔNG QUAN

1. Nhận dạng hành động người trong không gian 3D

Từ những năm 1980, nhận dạng hành động người luôn là lĩnh vực quan trọng trong các nghiên cứu về thị giác máy tính. Có rất nhiều hướng tiếp cận khác nhau đã được đề xuất và phần lớn trong số đó sử dụng hình ảnh từ các camera thông dụng. Các phương pháp thu thập dữ liệu chuyển động 3D đã ra đời và phát triển mạnh mẽ trong các thập niên gần đây. Phần này khái quát các kỹ thuật thu thập dữ liệu chuyển động 3D cũng như các phương pháp học máy phổ biến trong nhận dạng hành động người.

1.1. Các phương pháp thu thập dữ liệu chuyển động 3D

Từ sau sự ra đời của các thiết bị cảm biến chiều sâu (depth sensor), hoạt động nghiên cứu trên dữ liệu 3D đã có những chuyển biến tích cực. Nhìn chung có ba phương pháp chính trong việc thu thập dữ liệu chuyển động 3D. Đầu tiên là phương pháp sử dụng hệ thống chụp chuyển động dựa vào các marker như là Mocap (Motion Capture). Kế đến là phương pháp sử dụng stereo camera. Cuối cùng là phương pháp sử dụng range sensor[7]. Mục này sẽ trình bày chi tiết từng phương pháp.

1.1.1. Phương pháp sử dụng stereo camera

Thu thập dữ liệu 3D từ stereo camera là một trong những nghiên cứu kinh điển trong lĩnh vực thị giác máy tính. Trước đây các thiết bị range sensor

[7] Range sensor hay range camera là một loại cảm biến dùng để tạo ra một loại hình ảnh, trong đó giá trị của các điểm ảnh là khoảng cách của nó tới một điểm nhất định nào đó. Ảnh được tạo bởi range senor gọi là range image.

rất đắt đỏ và cồng kềnh, do đó stereo camera đã thu hút được sự quan tâm của các nhà nghiên cứu trong việc xây dựng các hệ thống thị giác. Một stereo camera được trang bị hai hay nhiều ống kính với cảm biến ảnh hoặc khung phim riêng biệt cho mỗi ống kính. Điều đó cho phép camera có thể mô phỏng thị giác của con người, đây là tiền đề cho khả năng xây dựng dữ liệu hình ảnh 3D. Hình 1.1 và hình 1.2 là hình ảnh về stereo camera của Kodak và Sputnik với nhiều ống kính riêng biệt.

Hình 1.1 Kodak stereo camera

Stereo camera có vai trò đặc biệt quan trọng trong các lĩnh vực như người máy (robotics) và được ứng dụng rộng rãi trong giải trí, truyền thông và các hệ thống tự hành. Điển hình như kỹ thuật stereo trắc quang của tiến sĩ M. Petrou [17]. Ông phát triển một hệ thống gồm một camera cố định và ba đèn chiếu sáng đối tượng từ các góc độ khác nhau. Tất cả dữ liệu được kết hợp thành một dạng dữ liệu 3D bằng cách phân tích các vùng tối sáng khác nhau. Nó đã được ứng dụng để tìm lỗi của các sản phẩm công nghiệp và mô phỏng mô hình 3D của khuôn mặt con người.

Hình 1.2 Sputnik stereo camera

Vì sự phức tạp của hình học, việc thu thập dữ liệu 3D từ stereo camera vẫn còn là một nhiệm vụ đầy thử thách. Với sự ra đời và phát triển mạnh mẽ của range sensor, hầu hết các thiết bị stereo camera đã không còn được sản xuất cho tới ngày nay.

1.1.2. Phương pháp sử dụng Mocap

Kỹ thuật thu thập dữ liệu 3D tiếp theo là sử dụng hệ thống chụp chuyển động – Mocap. Nó là một phương pháp quan trọng trong việc theo dõi và phân tích cấu trúc hình thể của con người. Mocap được sử dụng rộng rãi trong điện ảnh, hoạt hình và trò chơi điện tử. Ngoài ra, người ta còn sử dụng Mocap để phân tích và hoàn thiện các động tác trong thể thao, khiêu vũ, cũng như giám sát tiến độ phục hồi trong vật lý trị liệu. Hình 1.3 là hình ảnh một Mocap đang ghi nhận chuyển động của một đối tượng.

Hình 1.3 Một hệ thống chụp chuyển động

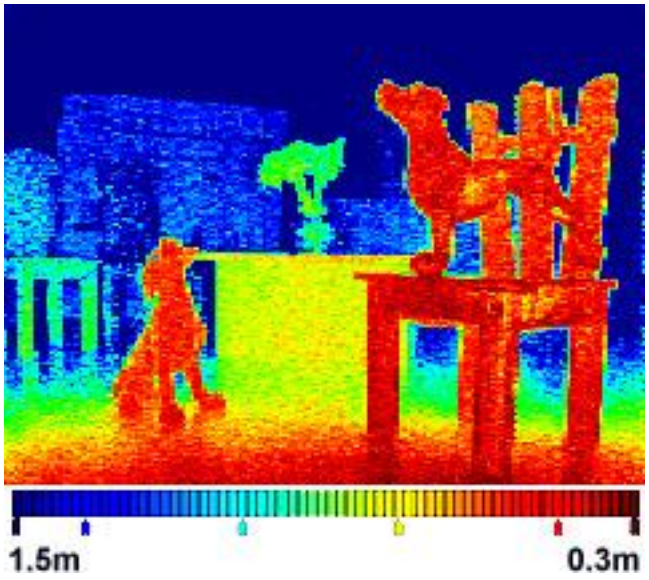
Có rất nhiều cách khác nhau để xây dựng một Mocap. Phổ biến nhất là sử dụng các marker cảm quang cố định trên chủ thể (thường là tại các khớp), đồng thời bố trí nhiều camera xung quanh để ghi nhận tọa độ và sự chuyển động của các marker khi chủ thể chuyển động. Ngoài ra, một số hệ thống sử dụng thẻ RFID (Radio Frequency Identification) hay các loại thẻ từ khác để thay thế cho marker. Chi tiết về hệ thống chụp chuyển động của trường đại học CMU (Carnegie Mellon University) sẽ được trình bày trong phần **2** của chương này.

Hiện nay có một số cơ sở dữ liệu chuyển động 3D được thu thập bởi hệ thống chụp chuyển động như CMU Motion Capture Database[8], MPI HDM05 Motion Capture Database2[9], CMU Kitchen DataSet[10], LACE Indoor Activity Benchmark Dataset[11], và TUM Kitchen Dataset[12]. Các cơ sở dữ liệu này được cung cấp miễn phí cho mục đích nghiên cứu.

[8] http://mocap.cs.cmu.edu/.
[9] http://www.mpi-inf.mpg.de/resources/HDM05/.
[10] http://kitchen.cs.cmu.edu/.
[11] http://www.cs.rochester.edu/%18spark/muri/.

1.1.3. Phương pháp sử dụng range sensor

Range sensor hay range camera là một loại cảm biến dùng để tạo ra range image. Range image là một dạng hình ảnh trong đó giá trị của các điểm ảnh là khoảng cách của nó tới một điểm cố định nào đó (cảm biến). Hình 1.4 là một ví dụ minh họa range image, những vùng càng gần với cảm biến được thể hiện bằng màu đỏ, vùng xa hơn là màu vàng và xa nhất là màu xanh. Dựa vào các khoảng cách này, chúng ta có thể dễ dàng dựng được cấu trúc ba chiều của vật thể.



Hình 1.4 Một ví dụ minh họa cho range image

Ngày nay một số công ty sử dụng range sensor kết hợp với các kỹ thuật khác đã tạo ra những thiết bị có thể dựng mô hình khung xương của chủ thể trong thời gian thực, điển hình là Microsoft Kinect.

[12] http://ias.in.tum.de/software/kitchen-activity-data/.

1.2. Các phương pháp học máy thường sử dụng

Học máy (machine learning) có rất nhiều phương pháp, thuật toán khác nhau. Có nhiều cách để phân loại các thuật toán học máy, cách phân loại phổ biến nhất chia học máy thành ba loại chính: học có giám sát, học không giám sát và học bán giám sát.

Học có giám sát (supervised learning) là phương pháp học máy sử dụng bộ dữ liệu huấn luyện có gán nhãn. Tức là dữ liệu bao gồm các cặp các đối tượng có đầu vào và đầu ra tương ứng. Một số thuật toán học có giám sát phổ biến như: máy vectơ hỗ trợ, mô hình Markov ẩn, thuật toán k láng giềng gần nhất (k-NN), mạng nơ-ron (NN).

Học không giám sát (unsupervised learning) là phương pháp học máy sử dụng tập dữ liệu không gán nhãn (gọi là tập dữ liệu quan sát), tức là không biết trước đầu ra tương ứng cho mỗi đối tượng. Một số phương pháp phổ biến trong học không giám sát như K-mean, Fuzzy C-means (FCM), Hierarchical Agglomerative Clustering (HAC), Gaussian Mixture Model (GMM).

Học bán giám sát (semi-supervised learning) là phương pháp học máy sử dụng cả dữ liệu đã gán nhãn và chưa gán nhãn để huấn luyện, điển hình là một lượng nhỏ dữ liệu có gán nhãn cùng với lượng lớn dữ liệu chưa gán nhãn. Một số thuật toán phổ biến gồm có cực đại kỳ vọng (EM), SVM truyền dẫn (TSVM).

Với tập hợp dữ liệu chuyển động 3D đã được gắn nhãn, phần lớn các nhà nghiên cứu áp dụng phương pháp học có giám sát để xây dựng mô hình nhận dạng. Các phương pháp thường dùng là máy vectơ hỗ trợ (SVM), mô hình Markov ẩn (HMM), kết hợp giữa SVM và HMM. Phần tiếp theo sẽ giới thiệu hai phương pháp phổ biến nhất: SVM và HMM.

1.2.1. Máy vectơ hỗ trợ

Máy vectơ hỗ trợ (Support Vector Machine – SVM) làm một giải thuật học máy dựa trên lý thuyết học thống kê. Bài toán cơ bản của SVM là bài toán phân loại hai lớp: Cho trước n điểm trong không gian d chiều (mỗi điểm thuộc vào một lớp kí hiệu là +1 hoặc -1, mục đích của giải thuật SVM là tìm một siêu phẳng (hyperplane) phân hoạch tối ưu cho phép chia các điểm này thành hai phần sao cho các điểm cùng một lớp nằm về một phía với siêu phẳng này.

Xét tập dữ liệu mẫu có thể tách rời tuyến tính $\{(x_1, y_1),\ (x_2, y_2), \dots, (x_n, y_n)\}$ với $x_i \in \boldsymbol{R}^d$ và $y_i \in \{-1, 1\}$. Siêu phẳng tối ưu phân tập dữ liệu này thành hai lớp là siêu phẳng có thể tách rời dữ liệu thành hai lớp riêng biệt với lề (margin) lớn nhất. Tức là, cần tìm siêu phẳng **H**: $v = w.x + \mathrm{b} = 0$ và hai siêu phẳng **H**1, **H**2 hỗ trợ song song với **H** và có cùng khoảng cách đến H. Với điều kiện không có phần tử nào của tập mẫu nằm giữa **H**1 và **H**2, khi đó:

$$\begin{cases} \mathrm{w}.x + \mathrm{b} \geq 1 \text{ với } \mathrm{y} = 1 \\ \mathrm{w}.x + \mathrm{b} \leq -1 \text{ với } \mathrm{y} = -1 \end{cases}$$

Kết hợp hai điều kiện trên ta có $y(\mathrm{w}.x + \mathrm{b}) \geq 1$.

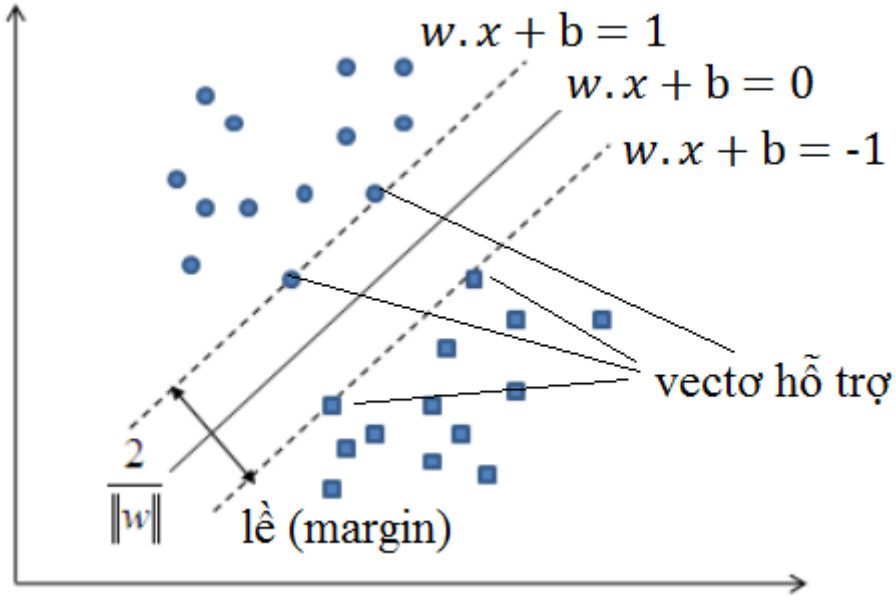
Khoảng cách của siêu phẳng **H**1 và **H**2 đến **H** là $\|\mathrm{w}\|$. Ta cần tìm siêu phẳng **H** với lề lớn nhất, tức là giải bài toán tối ưu tìm $\min_{w,b} \|\mathrm{w}\|$ với ràng buộc $y(\mathrm{w}.x + \mathrm{b}) \geq 1$. Bài toán này có thể chuyển sang bài toán tương đương dễ giải hơn là $\min_{w,b} \frac{1}{2}\|\mathrm{w}\|^2$ với ràng buộc $y(\mathrm{w}.x + \mathrm{b}) \geq 1$. Lời giải cho bài toán tối ưu này là cực tiểu hóa hàm Lagrange:

$$\mathrm{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i (w.x_i + b) - 1]$$

Trong đó α là các hệ số Lagrange, $\alpha \geq 0$. Sau đó người ta chuyển thành bài toán đối ngẫu là cực đại hóa hàm $W(\alpha)$:

$$\max_{\alpha} W(\alpha) = \max_{\alpha}(\min_{w,b} L(w, b, \alpha))$$

Từ đó giải để tìm được các giá trị tối ưu cho w, b và α. Về sau, việc phân loại một mẫu mới chỉ là việc kiểm tra hàm dấu sign(w.x + b). Hình 1.5 là một minh họa siêu phẳng với lề cực đại trong không gian hai chiều. Các phần tử nằm trên lề gọi là vectơ hỗ trợ.

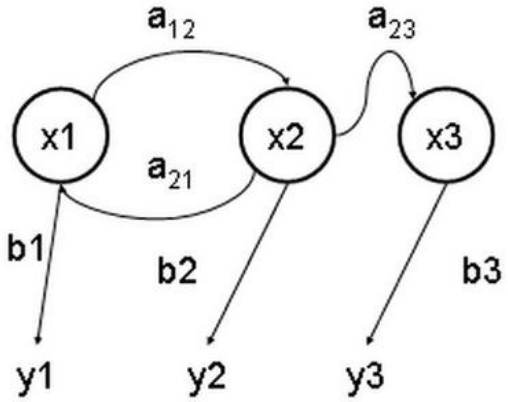


Hình 1.5 Siêu phẳng với lề cực đại trong không gian hai chiều

1.2.2. Mô hình Markov ẩn

Mô hình Markov ẩn (Hidden Markov Model) là một mô hình xác suất hữu hạn trạng thái theo kiểu phát sinh tiến trình bằng cách định nghĩa xác suất liên kết trên các chuỗi quan sát. Mỗi chuỗi quan sát được sinh ra bởi một chuỗi các phép chuyển trạng thái, bắt đầu từ trạng thái khởi đầu cho đến khi thu được trạng thái kết thúc. Tại mỗi trạng thái mỗi phần tử của chuỗi quan sát được phát sinh ngẫu nhiên trước khi chuyển sang trạng thái tiếp theo. Hình

1.6 biểu diễn các chuyển tiếp trạng thái trong HMM. Trong đó xi là các trạng thái trong mô hình; a_{ij} là xác suất chuyển tiếp từ trạng thái i sang trạng thái j; bi là các xác suất đầu ra, yi là dữ liệu quan sát.



Hình 1.6 Các chuyển tiếp trạng thái trong HMM

Các trạng thái của HMM được xem là ẩn bên trong mô hình vì tại mỗi thời điểm chỉ nhìn thấy các kí hiệu quan sát còn các trạng thái cũng như sự chuyển đổi trạng thái được vận hành ẩn bên trong mô hình.

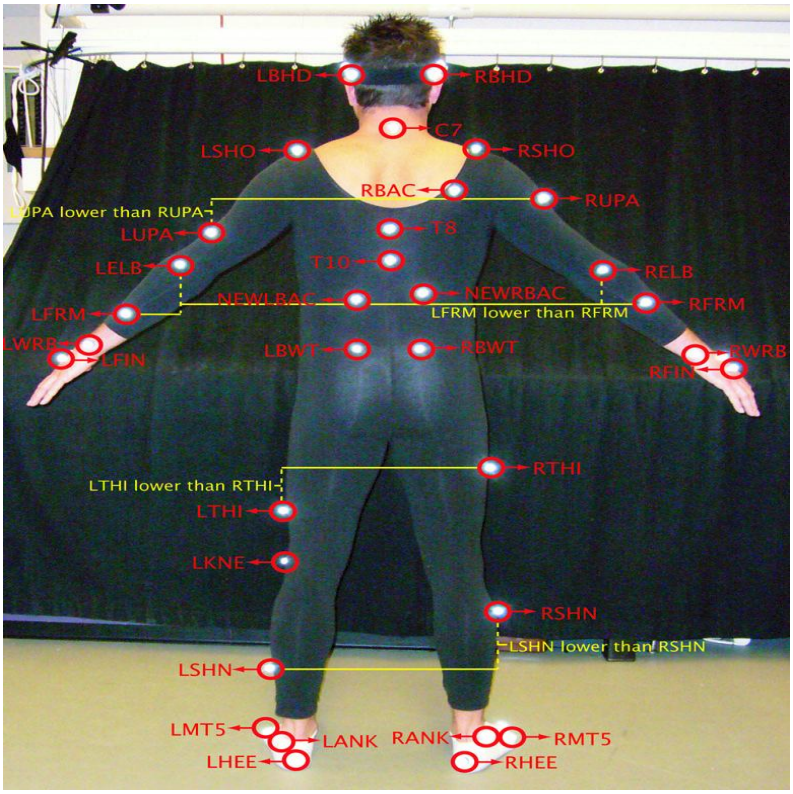
2. Hệ thống chụp chuyển động – Mocap

Dữ liệu đầu vào trong quá trình thực nghiệm của luận văn là dữ liệu chuyển động 3D (ở định dạng Acclaim - asf/amc) được thu thập từ hệ thống chụp chuyển động (Mocap) của trường đại học CMU. Phần này sẽ trình bày chi tiết về hệ thống chụp chuyển động của CMU cũng như dữ liệu thu được từ hệ thống này. Cuối cùng là chi tiết về cấu trúc dữ liệu Acclaim.

2.1. Mocap

Để theo dõi chuyển động của các đối tượng, các nhà nghiên cứu tại đại học Carnegie Mellon đã xây dựng hệ thống gồm 12 camera hồng ngoại MX-

40[13] lắp đặt xung quanh một không gian hình chữ nhật có kích thước 3m x 8m. Các chuyển động diễn ra trong vùng này sẽ được ghi lại. Đối với các chuyển động đòi hỏi sự tỉ mỉ và chi tiết như chuyển động của tay, các camera có thể di chuyển vào gần hơn. Người ta sử dụng một bộ quần áo đặc biệt có gắn 41 marker bên trên, chủ thể phải mặc bộ quần áo này và di chuyển trong vùng ghi nhận. Vị trí các marker được mô tả như hình 1.7. Các camera sẽ định vị marker bằng sóng hồng ngoại. Tín hiệu thu được từ hệ thống camera được xử lý và cho ra kết quả cuối cùng là dữ liệu dạng mô hình hóa 3D của cơ thể người. Có rất nhiều cấu trúc định dạng khác nhau được sử dụng như *asf/amc, vsk/v, c3d, bvh, txt*. Có một số phần mềm hỗ trợ việc chuyển đổi qua lại giữa các định dạng này[14].



Hình 1.7 Vị trí của một số marker

[13] MX-40 là sản phẩm của Vicon – một công ty chuyên cung cấp các thiết bị dùng trong hệ thống chụp chuyển động. MX-40 có thể quay với tốc độ 120Hz tức là ghi được 120 khung hình trong một giây.
[14] http://mocap.cs.cmu.edu/resources.php

2.2. Dữ liệu thu được từ Mocap

Dữ liệu thu được từ Mocap là dữ liệu dưới dạng mô hình hóa 3D của cơ thể người. Có rất nhiều định dạng khác nhau được dùng để lưu trữ loại dữ liệu này như *asf/amc*, *bvh*, *c3d*, *vsk/v*, *txt*. Trong quá trình thu nhận tín hiệu, các marker được định vị ở những vị trí định sẵn cho trước. Tín hiệu thu được từ các camera sẽ được phần mềm của Vicon (Vicon Bodybuilder) tổng hợp và mô hình hóa thành dữ liệu 3D.

Dưới góc độ người dùng có hai loại dữ liệu chính:

Loại thứ nhất là dạng dữ liệu nhị phân chứa vị trí của các marker trong không gian 3D với định dạng tệp là *c3d*. Định dạng này chỉ mô tả vị trí 3D của các marker trong quá trình chuyển động của đối tượng, nhưng lại không cho biết mối liên hệ giữa các marker với nhau.

Loại dữ liệu thứ hai chi tiết hơn, thường là một cặp gồm hai loại tệp văn bản (text) ở định dạng *asf/amc* hoặc *vsk/v*. Tệp thứ nhất chứa các thông tin về chi tiết về mô hình 3D của khung xương và các khớp nối; trong đó bao gồm các kết nối, vị trí, độ dài và hướng của các đoạn xương và độ tự do của các khớp. Tệp còn lại chứa thông tin chuyển động của mỗi đoạn xương. Nếu hành động của cùng một đối tượng được ghi nhận bằng nhiều clip thì sẽ có nhiều tệp *amc* hoặc *v* tương ứng với một tệp *asf* hoặc *vsk*. Nhìn chung dữ liệu dưới dạng văn bản (text) dễ đọc và dễ sử dụng hơn so với dữ liệu nhị phân. Luận văn này chọn dữ liệu dưới dạng *asf/amc* làm dữ liệu đầu vào để nghiên cứu. Phần tiếp theo sẽ trình bày chi tiết cấu trúc của loại dữ liệu này.

2.3. Cấu trúc Acclaim

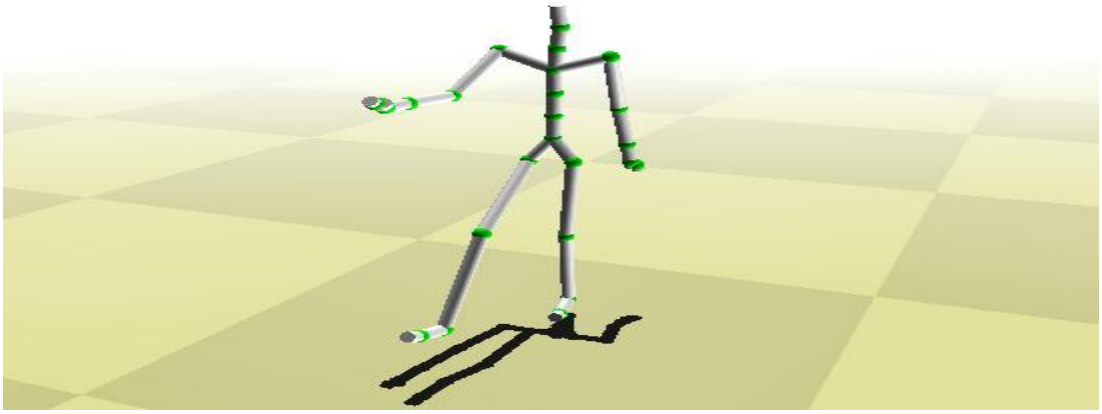
Acclaim là một công ty game (2006 – 2010) có rất nhiều nghiên cứu trong lĩnh vực theo dõi chuyển động. Họ phát triển và sử dụng cấu trúc riêng

trong việc lưu trữ dữ liệu chuyển động 3D. Đó là một bộ gồm hai tệp văn bản. Sau này, họ công bố và cho phép sử dụng rộng rãi cấu trúc đó trên toàn thế giới. Có rất nhiều công ty, tổ chức sử dụng cấu trúc Acclaim làm dữ liệu đầu ra cho các hệ thống theo dõi chuyển động của họ.

Acclaim là một bộ gồm hai tệp văn bản có cấu trúc. Tệp thứ nhất lưu trữ cấu trúc tổng quan của đối tượng, tiệp thứ hai chứa dư liệu chuyển động tương ứng của đối tượng theo thời gian. Nếu là chuyển động của con người thì tệp thứ nhất thể hiện mô hình 3D của xương người gồm các đoạn xương, chiều dài, hướng, độ tự do (degrees of freedom - dof) của mỗi xương, tệp thứ hai là giá trị cụ thể của mỗi dof với mỗi mốc thời gian tương ứng. Tệp cấu trúc là ASF (Acclaim Skeleton File), tệp chuyển động là AMC (Acclaim Motion Capture).

2.3.1. Cấu trúc tệp ASF

Tệp cấu trúc ASF mô tả sự gắn kết giữa các xương trong cơ thể và độ tự do (degrees of freedom) của các khớp. ASF chính là trạng thái ban đầu của dữ liệu chuyển động. Hình 1.8 mô tả mô hình 3D của bộ xương người được dựng lại bằng tệp *.asf* .



Hình 1.8 Mô hình 3D của xương người được dựng lại từ tệp .asf

Thành phần cụ thể trong ASF được mô tả như sau:

- Các chú thích được bắt đầu bởi dấu thăng (#).
- Các từ khóa bắt đầu bởi dấu hai chấm (**:**). Từ khóa có thể được dùng cho các giá trị toàn cục hoặc bắt đầu cho một mục dữ liệu.
- Từ khóa **version** cho biết phiên bản hiện tại của tệp tin.
- Từ khóa **name** được dùng để đặt tên cho dữ liệu, tên này có thể khác với tên tệp.
- Mục **units** định nghĩa đơn vị cho một số loại dữ liệu trong tệp. Nó cũng có thể chứa giá trị mặc định cho các đại lượng.
- Mục **documentation** lưu trữ các thông thêm về dữ liệu
- Mục **root** định nghĩa một khớp xương đặc biệt của ASF, đây chính là nút gốc trong hệ thống cây với nút là các khớp và nhánh là các đoạn xương giữa các khớp. Từ khóa **axis** trong mục **root** cho biết thứ tự quay của các trục tọa độ X, Y, Z của nút gốc. Từ khóa **order** cho biết các kênh chuyển động của nút gốc cũng như thứ tự xuất hiện của các đại lượng này trong tệp AMC. Hai từ khóa còn lại **position** và **orientation** lưu giữ tọa độ và phương hướng ban đầu của nút gốc, thường thì giá trị của các đại lượng này bằng không (0).
- Mục **bonedata** là nơi chứa thông tin chi tiết của mỗi đoạn xương trong cây hệ thống. Các thông tin của mỗi đoạn xương được đặt trong một cặp từ khóa **begin** và **end.** Với mỗi cặp, chúng ta có: **id, name, direction** là hướng của xương, **length** là độ dài, **axis** là góc quay của trục tọa độ tương đối của mỗi đoạn xương, **dof** là độ tự do của đoạn xương, **limits** là giới hạn quay của các khớp.
- Mục cuối cùng **hierarchy** định nghĩa sự liên kết giữa các khớp xương để tạo nên một bộ xương hoàn chỉnh.

Hình 1.9 mô tả một đoạn tệp ASF.

```
# AST/ASF file generated using VICON BodyLanguage
# ---------------------------------------------
:version 1.10
:name VICON
:units
  mass 1.0
  length 0.45
  angle deg
:documentation
   .ast/.asf automatically generated from VICON data using
   VICON BodyBuilder and BodyLanguage model FoxedUp or BRILLIANT.MOD
:root
   order TX TY TZ RX RY RZ
   axis XYZ
   position 0 0 0
   orientation 0 0 0
:bonedata
  begin
     id 1
     name lhipjoint
     direction 0.693937 -0.600361 0.397515
     length 2.37164
     axis 0 0 0  XYZ
  end
  begin
     id 2
     name lfemur
     direction 0.34202 -0.939693 0
     length 6.18497
     axis 0 0 20  XYZ
    dof rx ry rz
    limits (-160.0 20.0)
```

rmal text file length : 7264 lines : 339

Hình 1.9 Một đoạn tệp ASF

2.3.2. Cấu trúc tệp AMC

Tệp chuyển động AMC chứa các đại lượng có thể thay đổi trong hệ thống các khớp xương được định nghĩa ở tệp ASF. Là dữ liệu chuyển động theo thời gian nên trên tệp AMC được tạo thành bởi nhiều frame, mỗi frame thể hiện dữ liệu cho một thời điểm. Dữ liệu bên trong mỗi frame là giá trị theo thứ tự của đại lượng **dof** trên tệp ASF. Hình 1.10 mô tả một đoạn tệp AMC.

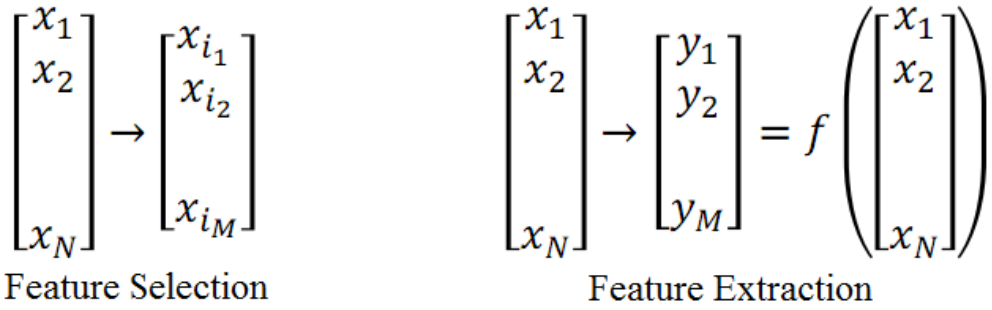
```
#!OML:ASF F:\VICON\USERDATA\INSTALL\rory3\rory3.ASF
:FULLY-SPECIFIED
:DEGREES
1
root -0.307087 17.6356 -28.2214 2.36076 1.44212 -4.54601
lowerback 15.4094 -0.182495 1.65268
upperback 1.54579 0.0318172 -0.110122
thorax -6.9977 -0.0335751 -1.06068
lowerneck -3.24163 -0.676991 -1.34632
upperneck -9.28199 -0.818331 1.08102
head -2.3551 -0.388697 0.578143
rclavicle 1.74931e-014 -4.77083e-015
rhumerus -42.2757 19.3184 -90.6312
rradius 79.2191
rwrist 2.46902
rhand -35.8906 32.487
rfingers 7.12502
rthumb -9.00425 2.69918
lclavicle 1.74931e-014 -4.77083e-015
lhumerus -46.581 -10.5126 91.072
lradius 108.082
lwrist 30.7395
lhand -39.5085 13.512
lfingers 7.12502
lthumb -12.4939 43.1185
rfemur 4.30283 -1.72433 25.7796
rtibia 82.7602
rfoot 27.83 -8.73877
rtoes 20.2614
lfemur -27.49 -2.09007 -20.1015
ltibia 38.398
lfoot -7.19848 -5.78026
ltoes 5.97973
2
root -0.303728 17.5624 -27.7253 2.02549 1.77071 -4.33872
```

Hình 1.10 Một đoạn tệp AMC

3. Trích chọn, lựa chọn đặc tính

Một khâu quan trọng trong quá trình xây dựng mô hình nhận dạng hành động người là trích chọn, lựa chọn đặc tính. Mục đích chung của trích chọn hay lựa chọn đặc tính là làm giảm độ lớn của dữ liệu, hay nói cách khác là

làm giảm số chiều của dữ liệu. Quá trình làm giảm số chiều của dữ liệu được chia làm hai loại chính: trích chọn đặc tính (feature extraction - FE) và lựa chọn đặc tính (feature selection - FS). Sự khác nhau cơ bản giữa hai phương pháp này là trích chọn đặc tính sử dụng phương pháp biến đổi tuyến tính hoặc phi tuyến tính để biến đổi dữ liệu sang không gian mới, trong khi đó lựa chọn đặc tính chọn một tập con của tập dữ liệu cho trước. Một bên biến đổi dữ liệu, một bên không biến đổi. Hình 1.11 mô tả sự khác nhau giữa hai phương pháp này. Kết quả của lựa chọn đặc tính là một tập con của dữ liệu ban đầu, kết quả của trích chọn đặc tính là một tập dữ liệu trong không gian mới ít chiều hơn $y = f(x)$.



Hình 1.11 Mô tả lựa chọn đặc tính (trái) và trích chọn đặc tính (phải)

Đối với bài toán nhận dạng, mục đích của hai phương pháp này là giống nhau nên từ đây, trong luận văn này xin dùng "trích chọn đặc tính" làm tên gọi chung cho cả hai phương pháp.

Có rất nhiều phương pháp trích chọn đặc tính khác nhau được đề xuất sử dụng trong nhận dạng hành động. Luận văn này chọn một số phương pháp tiêu biểu để nghiên cứu như: phân tích thành phân chính (PCA), biệt thức tuyến tính (LDA). Nội dung chính của từng phương pháp được trình bày ở phần tiếp theo.

3.1. Phương pháp phân tích thành phần chính – PCA

Phương pháp phân tích thành phần chính (Principal Components Analysis - PCA) là một thuật toán thống kê sử dụng phép biến đổi trực giao để biến đổi một tập hợp dữ liệu từ một không gian nhiều chiều sang một không gian mới ít chiều hơn nhằm tối ưu hóa việc thể hiện sự biến thiên của dữ liệu (maximize the variability).

Các đặc tính của PCA:

Giúp giảm số chiều của dữ liệu.

Thay vì giữ lại các trục tọa độ của không gian cũ, PCA xây dựng một không gian mới ít chiều hơn, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương không gian cũ, nghĩa là đảm bảo độ biến thiên (variability) của dữ liệu trên mỗi chiều mới.

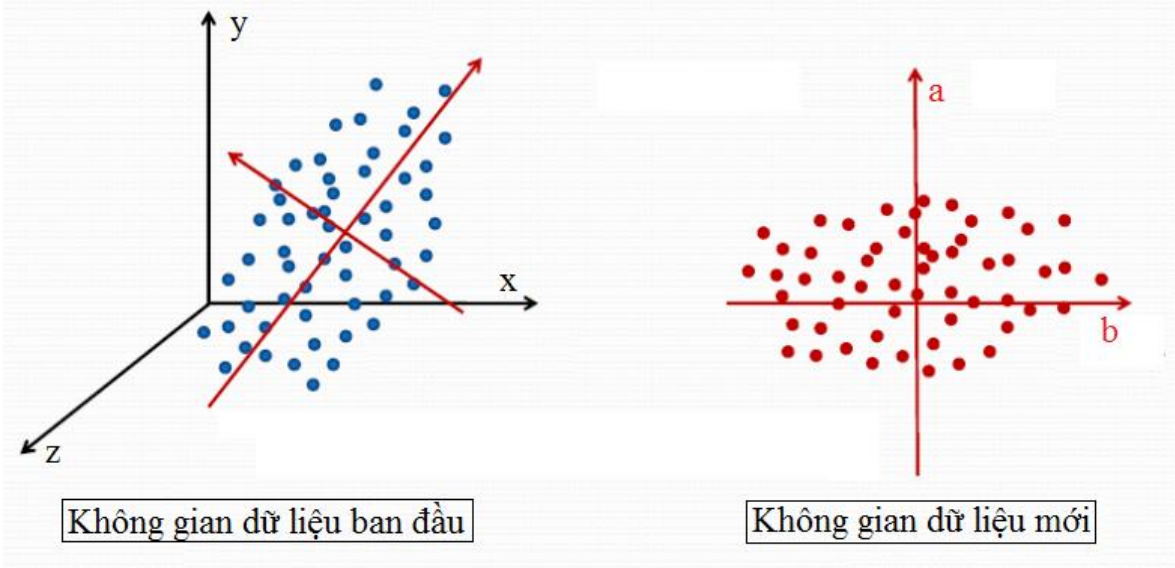
Các trục tọa độ trong không gian mới là tổ hợp tuyến tính của không gian cũ, do đó về mặt ngữ nghĩa, PCA xây dựng các thuộc tính mới dựa trên các thuộc tính hiện có. Những thuộc tính này vẫn biểu diễn tốt dữ liệu ban đầu.

Trong không gian mới, các liên kết tiềm ẩn của dữ liệu có thể được khám phá, mà nếu đặt trong không gian cũ thì khó phát hiện hơn, hoặc những liên kết như thế không thể hiện rõ.

Hình 1.12 Hình ảnh con lạc đà qua 2 góc độ khác nhau

Hình 1.12 là một ví dụ kinh điển để minh họa PCA. Cùng là một con lạc đà nhưng nếu nhìn từ bên hông thì ta có được đầy đủ thông tin nhất, trong khi nhìn từ phía trước thì thật khó để nói nó là con lạc đà.

Một ví dụ thuyết phục hơn được minh họa như hình 1.13



Hình 1.13 Minh họa PCA

Giả sử tập dữ liệu ban đầu (tập điểm màu xanh) được quan sát trong không gian ba chiều (trục màu đen) như hình bên trái. Rõ ràng ba trục này

không biểu diễn được tốt nhất mức độ biến thiên của dữ liệu. PCA do đó sẽ tìm hệ trục tọa độ mới (là hệ trục màu đỏ trong hình bên trái). Sau khi tìm được không gian mới, dữ liệu sẽ được chuyển qua không gian này để được biểu diễn như trong hình bên phải. Rõ ràng hình bên trái chỉ cần hai trục tọa độ nhưng biểu diễn tốt hơn độ biến thiên của dữ liệu so với hệ trục ba chiều ban đầu.

Đặc trưng của PCA là các trục tọa độ trong không gian mới luôn đảm bảo trực giao đôi một với nhau, mặt dù trong không gian ban đầu, các trục có thể không trực giao.

Thuật toán PCA:

Về cơ bản, thuật toán PCA gồm có ba bước: tiền xử lí, xây dựng không gian mới, chuyển dữ liệu từ không gian ban đầu sang không gian mới. Cho ma trận $\mathbf{X} = \{x_{ij}\} \in \boldsymbol{R}^{n\,x\,p}$ là tập dữ liệu ban đầu x_{ij} là thuộc tính thứ j của dữ liệu i. Các bước của PCA lần lượt như sau:

Bước 1: Tiền xử lí

Dữ liệu ban đầu có thể có giá trị thay đổi bất thường. Ví dụ một thuộc tính có giá trị thay đổi trong khoảng (0, 1) nhưng trên thuộc tính khác lại biến thiên trong đoạn (-100, 100). Rõ ràng cần phải có một bước tiền xử lí để chuẩn hóa các giá trị trên các cột của ma trận X. Có hai cách tiền xử lí thường được dùng cho PCA là **Centered PCA** và **Normed PCA** (một số tài liệu gọi là Standardize PCA).

Centered PCA mang tất cả các thuộc tính (các cột của X) về cùng một gốc tọa độ:

$\widehat{\mathbf{X}} = \{\widehat{x_{ij}}\}$,

$$\widehat{x_{ij}} = \frac{x_{ij} - g_j}{\sqrt{n}}$$

Trong đó n là số dòng của **X**, g_j là mean của cột thứ j của **X**, được tính như sau:

$$g_j = \frac{\sum_{i=1}^{n} x_{ij}}{n}.$$

Normed PCA mang tất các các thuộc tính về cùng một gốc tọa độ, đồng thời chuẩn hóa về cùng một quãng có độ lệch chuẩn (standard deviation) là 1:

$\widehat{\mathbf{X}} = \{\widehat{x_{ij}}\}$,

$$\widehat{x_{ij}} = \frac{x_{ij} - g_j}{\sqrt{n}\sigma_j}$$

Trong đó σ_j là phương sai của cột thứ j trong **X**.

Thông thường, Normed PCA hay được dùng, sau bước tiền xử lý, ma trận $\widehat{\mathbf{X}}$ sẽ là đầu vào cho bước tiếp theo.

Bước 2: Xây dựng không gian mới

Tính ma trận hiệp phương sai (covariance) của các thuộc tính trong $\widehat{\mathbf{X}}$:

$$\mathbf{V} = \widehat{\mathbf{X}}^{\mathbf{T}}\widehat{\mathbf{X}}$$

Do là tích của ma trận $\widehat{\mathbf{X}}$ với một chuyển vị của nó nên $\mathbf{V} \in \boldsymbol{R}^{p\ x\ p}$ là ma trận có kích thước $p\ x\ p$. Hơn nữa **V** có p trị riêng $\lambda_i \geq 0$, i=1..p.

Tiếp theo, PCA tìm trị riêng và vector riêng tương ứng của V, sắp xếp theo thứ tự giảm dần của trị riêng. Giả sử p trị riêng của **V** là $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_p$, và vector riêng tương ứng là $\mathbf{H} = \boldsymbol{u}_1, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_p$. Khi đó các trục của không gian mới chính là các vector riêng $\boldsymbol{u}_i$ ở trên, đương nhiên các vector riêng hoàn toàn độc lập tuyến tính (nghĩa là trực giao đôi một).

Bước 3: Chuyển dữ liệu từ không gian ban đầu sang không gian mới

Thông thường không gian mới không được xây dựng bằng tất cả các p vector riêng trong **H**, mà thông thường chỉ từ k vector riêng đầu tiên.

Như vậy gọi $\mathbf{U} = [\ \boldsymbol{u}_1 \mid \boldsymbol{u}_2 \mid \ldots \mid \boldsymbol{u}_k\ ] \in \boldsymbol{R}^{p\ x\ k}$.

Khi đó tọa độ các điểm trong hệ tọa độ mới là

$$\mathbf{F} = \widehat{\mathbf{X}}\mathbf{U}$$

3.2. Phương pháp phân tích biệt thức tuyến tính – LDA

Phân tích biệt thức tuyến tính (Linear Discriminant Analysis – LDA) là phương pháp được sử dụng trong thống kê và học máy để giải quyết bài toán phân lớp hoặc trích chọn đặc tính. Trong phân loại hai lớp dữ liệu, LDA tìm kiếm trục đường thẳng sao cho khi tất cả dữ liệu của 2 lớp ánh xạ lên trục này có độ phân ly 2 lớp là cao nhất. Độ phân ly các lớp dữ liệu được định nghĩa bởi tỷ lệ phương sai giữa các lớp và phương sai giữa các dữ liệu trong từng lớp.

Độ phân ly được biểu diễn như sau:

$$S = \frac{\sigma^2_{between}}{\sigma^2_{within}} \rightarrow max$$

Trong đó, $\sigma^2_{between}$ là phương sai giữa hai lớp, được tính bởi công thức:

$$\sigma^2_{between} = \frac{1}{n}\left(n_{y=0}\left(w \cdot \mu_{y=0} - w \cdot \mu\right)^2 + n_{y=1}\left(w \cdot \mu_{y=0} - w \cdot \mu\right)^2\right)$$

$$= \frac{1}{n} w^{\mathrm{T}} \left(n_{y=0}\left(\mu_{y=0} - \mu\right)\left(\mu_{y=0} - \mu\right)^{\mathrm{T}} + n_{y=1}\left(\mu_{y=1} - \mu\right)\left(\mu_{y=1} - \mu\right)^{\mathrm{T}}\right) w$$

$$= w^{\mathrm{T}} \sum\nolimits_{between} w$$

với:

$n_{y=0}$ và $n_{y=1}$ lần lượt là số phần tử dữ liệu của lớp có nhãn $y = 0$ và $y = 1$.

$\mu_{y=0}$ và $\mu_{y=1}$ lần lượt là các trọng tâm của lớp có nhãn $y = 0$ và lớp có nhãn $y = 1$.

$$\mu_{y=0} = \frac{1}{n_{y=0}} \sum_{y_i=0} w \cdot \mathrm{x}_i$$

$$\mu_{y=1} = \frac{1}{n_{y=1}} \sum_{y_i=1} w \cdot \mathrm{x}_i$$

$$\mu = \frac{1}{n} \sum_{i=1}^{n} w \cdot \mathrm{x}_i$$

Và σ^2_{within}là phương sai giữa các dữ liệu trong từng lớp, được tính bởi:

$$\sigma^2_{within} = \frac{1}{n}\left(n_{y=0} w^{\mathrm{T}} \sum_{y=0} w + n_{y=1} w^{\mathrm{T}} \sum_{y=1} w\right)$$

$$= w^{\mathrm{T}} \sum_{within} w$$

$\sum_{y=0}$và $\sum_{y=1}$ lần lượt là hiệp phương sai có nhãn $y = 0$ và$y = 1$.

$$\sum_{y=0} = \frac{1}{n_{y=0}} \sum_{y_i=1} \left(x_i - \mu_{y=0}\right)\left(x_i - \mu_{y=0}\right)^{\mathrm{T}}$$

$$\sum_{y=1} = \frac{1}{n_{y=1}} \sum_{y_i=1} \left(\mathrm{x}_i - \mu_{y=1}\right)\left(\mathrm{x}_i - \mu_{y=1}\right)^{\mathrm{T}}$$

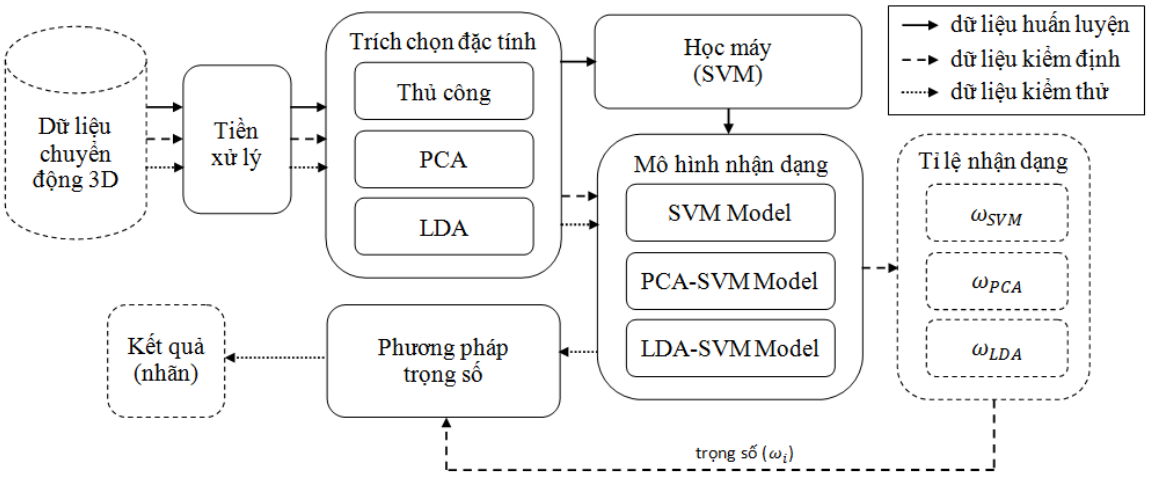
3.3. Sử dụng hàm nhân

Các phương pháp trích chọn đặc tính được trình bày ở trên chỉ có thể áp dụng hiệu quả với dữ liệu tuyến tính đơn giản, không phù hợp với các bài toán có dữ liệu phi tuyến, phức tạp. Do đó người ta dùng một phép biến đổi để chuyển dữ liệu phi tuyến tính thành tuyến tính. Phép biến đổi đó gọi là hàm nhân (kernel). Phương pháp PCA kết hợp hàm nhân gọi là KPCA (Kernel Principal Components Analysis), phương pháp LDA kết hợp hàm nhân gọi là KDA (Kernel Discriminant Analysis). Các phương pháp trích chọn đặc tính sử dụng trong luận văn này đều được kết hợp với hàm nhân.

CHƯƠNG 2
GIẢI PHÁP ĐỀ XUẤT

Hình 2.1 mô tả tổng quan hệ thống đề xuất, bao gồm năm khối chức năng chính: *tiền xử lý*, *trích chọn đặc tính*, *học máy*, *mô hình nhận dạng* và *phương pháp trọng số*. Có ba phương pháp chính trong *trích chọn đặc tính*: lựa chọn thủ công, PCA và LDA. *Mô hình nhận dạng* chính là mô hình học máy xây dựng được từ dữ liệu huấn luyện. Mỗi phương pháp trích chọn đặc tính xây dựng được một mô hình nhận dạng.

Dữ liệu đầu vào của hệ thống được chọn lựa trong cở sở dữ liệu của CMU Mocap và được phân chia ngẫu nhiên thành ba nhóm: nhóm thứ nhất được dùng làm dữ liệu huấn luyện (training data) cho giải thuật học máy, nhóm thứ hai được dùng để kiểm định độ chính các của mỗi mô hình học máy ứng với một phương pháp trích chọn đặc tính khác nhau (gọi là nhóm dữ liệu kiểm định - validating data), nhóm dữ liệu cuối cùng là dữ liệu kiểm thử (testing data) dùng để kiểm tra và đánh giá kết quả của hệ thống.



Hình 2.1 Mô hình hệ thống đề xuất

Trong hệ thống đề xuất, có tất cả ba luồng dữ liệu chính. Thứ nhất là luồng dữ liệu huấn luyện, được thể hiện bằng hình mũi tên nét liền. Kết quả cuối cùng của luồng dữ liệu huấn luyện là mô hình nhận dạng xây dựng được sau khi đã áp dụng giải thuật học máy. Thứ hai là luồng dữ liệu kiểm định, được thể hiện bằng hình mũi tên nét đứt. Kết quả cuối cùng của luồng dữ liệu kiểm định là tỉ lệ nhận dạng tương ứng với mỗi phương pháp trong trích chọn đặc tính. Các tỉ lệ này sẽ là đầu vào cho phương pháp trọng số. Thứ ba là luồng dữ liệu kiểm thử (hay bất kỳ dữ liệu mới nào), được mô tả bằng hình mũi tên chấm liền. Dữ liệu của luồng này sau khi qua các khối chức năng tiền xử lý, trích chọn đặc tính, mô hình nhận dạng, phương pháp trọng số sẽ cho ra kết quả là nhãn của hành động cần nhận dạng. Chức năng học máy chỉ được sử dụng bởi luồng dữ liệu huấn luyện, các luồng dữ liệu còn lại chỉ sử dụng mô hình nhận dạng xây dựng bởi học máy.

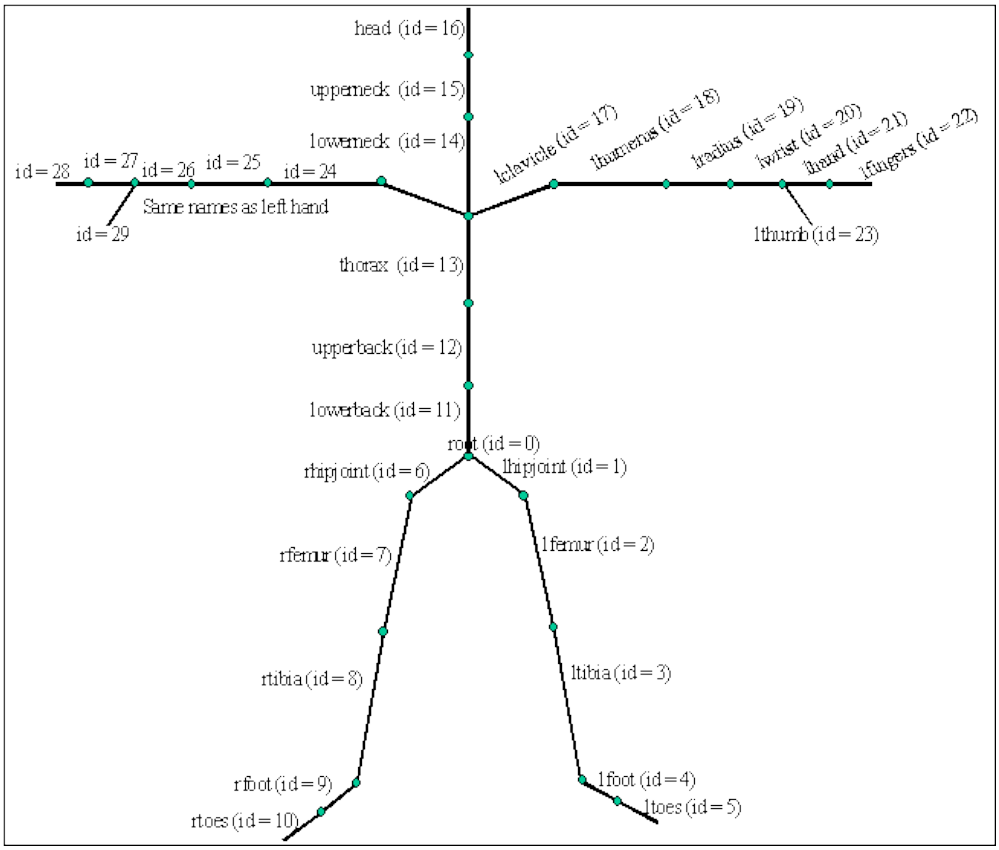
Các mục sau sẽ trình bày chi tiết từng khối chức năng trong hệ thống đồng thời giải thích vai trò của chúng với mỗi luồng dữ liệu khác nhau.

1. Tiền xử lý

Như đã trình bày ở các phần trước, dữ liệu chuyển động 3D sử dụng trong luận văn này được thu nhận từ Mocap, dưới định dạng Acclaim (asf/amc). Tệp *asf* là cấu trúc mô hình hóa 3D của bộ xương người, tệp *amc* chứa góc quay của các xương trong quá trình di chuyển của đối tượng. Tín hiệu video được quay bởi một loại camera chuyên dụng MX-40, có tốc độ 120Hz, nghĩa là dữ liệu thu được sẽ bao gồm 120 khung hình (frame) trong một giây. Tuy nhiên thời gian thực hiện của mỗi loại hành động trong mỗi thí nghiệm là khác nhau, do đó dữ liệu thu được có độ dài ngắn khác nhau ứng với số lượng nhiều hay ít các khung hình.

Ngoài ra, mô hình bộ xương 3D của con người có một số lượng lớn các đoạn xương, kết hợp với độ tự do của mỗi khớp sẽ làm tăng số chiều của thuộc tính.

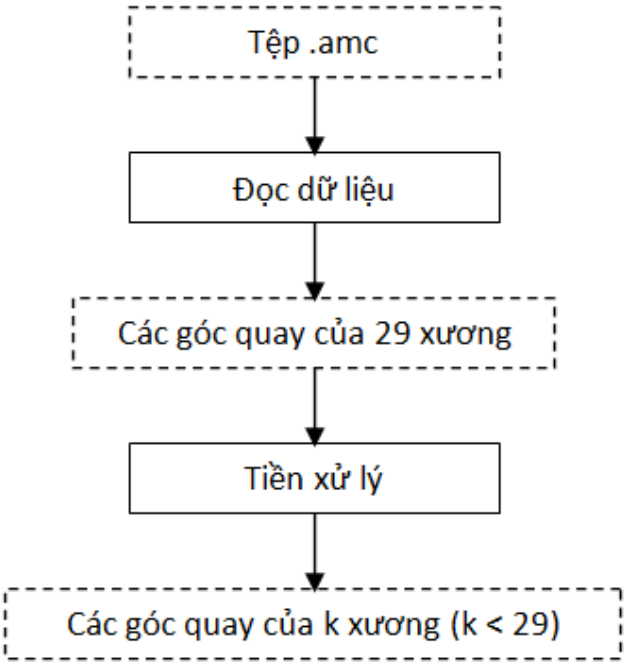
Thứ nhất, trước sự không đồng bộ của dữ liệu và độ lớn của số lượng thuộc tính, hệ thống cần có chức năng *tiền xử lý* để phần nào giải quyết hai vấn đề trên. Thứ nhất, để làm giảm số lượng các xương, K. Adistambha trong một nghiên cứu của mình [18] đã chỉ ra một nhóm các xương có thể thay thế cho toàn bộ xương trong quá trình nhận dạng mà vẫn đảm bảo độ chính xác của mô hình. Luận văn này chọn nhóm có 13 đoạn xương thay cho toàn bộ các xương. Các xương đó là: root, lowerback, upperback, thorax, lowerneck, upperneck, head, trái và phải clavicle, trái và phải humerus, trái và phải femur. Vị trí của các xương được mô tả trong hình 2.2.



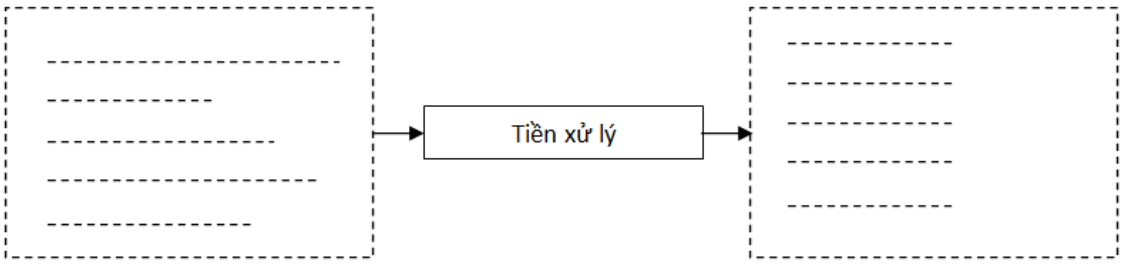
Hình 2.2 Mô tả vị trí các xương

Thứ hai, để chuẩn hóa dữ liệu, luận văn này chọn phương pháp đơn giản nhất là chỉ sử dụng một số lượng nhất định các khung hình (frame) đầu tiên của dữ liệu, số lượng khung hình cần thiết sẽ được quyết định thông qua thực nghiệm.

Hình 2.3 và 2.4 lần lượt mô tả quá trình xử lý dữ liệu nhằm giảm số chiều và chuẩn hóa dữ liệu.



Hình 2.3 Quá trình giảm số lượng xương

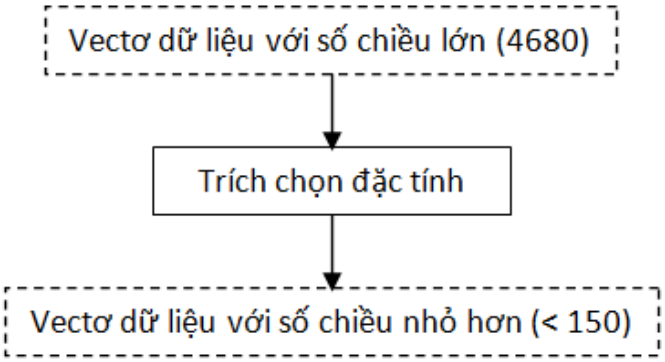


Hình 2.4 Quá trình chuẩn hóa dữ liệu

Dữ liệu sau khi được xử lý sẽ được một lần nữa tinh chỉnh bằng các phương pháp trích chọn đặc tính sẽ được trình bày trong phần tiếp theo.

2. Trích chọn đặc tính

Nội dung chính của luận văn này là nghiên cứu giai đoạn trích chọn đặc tính. Luận văn này sử dụng một số phương pháp trích chọn đặc tính phổ biến và kết hợp các phương pháp với nhau sử dụng trọng số. Hình 2.5 mô tả quá trình trích chọn đặc tính chung cho các phương pháp. Mục đích chung là tìm vectơ biểu diễn dữ liệu với số chiều nhỏ hơn dữ liệu ban đầu nhưng vẫn đảm bảo thể hiện đầ đủ các đặc trưng của dữ liệu.



Hình 2.5 Mô tả quá trình trích chọn đặc tính

Trước khi đi vào chi tiết các phương pháp trích chọn đặc tính, chúng ta hãy cùng xem xét độ lớn của dữ liệu của hệ thống hiện tại. Trong quá trình chuẩn hóa dữ liệu, kết quả cuối cùng là một vectơ có số hướng được xác định như sau: Ta dùng tất cả 13 xương, nếu mỗi xương đều có thể quay tự do theo 3 hướng thì số chiều của vectơ sẽ là 13 x 3 x f = (39 x f). Trong đó f là số lượng khung hình của một dữ liệu. Giả sử ta sử dụng 120 khung hình ban đầu, khi đó số chiều của vectơ sẽ là 39 x 120 = 4680. Đây là một con số lớn đối

với số chiều của một vectơ. Tuy nhiên số chiều của vectơ sẽ giảm đáng kể với các phương pháp trích chọn đặc tính sau:

2.1. Lựa chọn thủ công

Mô hình 3D cơ thể người được cấu thành từ tất cả 29 xương khác nhau. Tuy nhiên rất nhiều trong số đó không phải là đặc trưng của một số loại hành động. Đo đó luận văn này sẽ tiến hành thực nghiệm với một tập con các xương trong tổng số 29 xương ở trên. Quá trình thực nghiệm sẽ tiến hành với một số các xương cơ bản trước, sau đó sẽ từng bước thêm vào các xương khác nhau để tìm ra tập con thích hợp nhất.

Phân chia theo tầng suất hoạt động, các xương trong mô hình 3D của cơ thể người được chia làm hai nhóm. Nhóm thứ nhất gồm các xương có tầng suất di chuyển thấp, đó là các xương nằm trên trục xương sống của cơ thể. Nhóm thứ hai gồm các xương có tầng suất di chuyển cao hơn, đó là các xương nằm trên tay và chân. Quá trình lựa chọn sẽ bắt đầu với các xương nằm trên trên trục xương sống của cơ thể, sau đó thêm các xương khác vào sau mỗi lần thực nghiệm và đánh giá kết quả, từ đó đưa ra được nhóm xương có vai trò lớn nhất với bài toán nhận dạng hành động. Tỉ lệ nhận dạng của mỗi nhóm xương sẽ được so sánh với tỉ lệ nhận dạng khi thực nghiệm trên tất cả các xương. Việc đánh giá kết quả một nhóm xương tốt hay xấu phụ thuộc vào sai số giữa nó với kết quả của toàn bộ xương và phụ thuộc vào số lượng các xương trong nhóm. Sai số và số lượng xương càng nhỏ càng tốt. Dự kiến thứ tự thực nghiệm và số lượng xương trong các nhóm thự nghiệm như sau:

Nhóm đầu tiên có ít xương nhất, gồm ba xương trên trục xương sống của cơ thể.

```
// 3 bones
string[] boneNames = new string[] { "root", "lowerback", "upperback"};
```

Nhóm thứ hai có bốn xương, bao gồm các xương ở nhóm thứ nhất và thêm vào xương phần ngực.

```
// 4 bones
string[] boneNames = new string[] { "root", "lowerback", "upperback", "thorax"};
```

Nhóm thứ ba có bảy xương, bao gồm các xương ở nhóm thứ hai và thêm vào các xương trên vùng cổ, đầu.

```
// 7 bones
string[] boneNames = new string[] { "root", "lowerback", "upperback", "thorax",
"lowerneck", "upperneck", "head"};
```

Nhóm thứ tư có 13 xương, bao gồm các xương ở nhóm thứ ba và thêm vào các xương trên vai, cánh tay và đùi.

```
// 13 bones
string[] boneNames = new string[] { "root", "lowerback", "upperback", "thorax",
"lowerneck", "upperneck", "head", "rclavicle", "lclavicle", "rhumerus",
"lhumerus", "rfemur", "lfemur"};
```

Nhóm thứ năm có 23 xương, bao gồm các xương ở nhóm thứ tư và thêm vào các xương trên cẳng tay, cẳng chân, cổ tay, bàn tay và bàn chân.

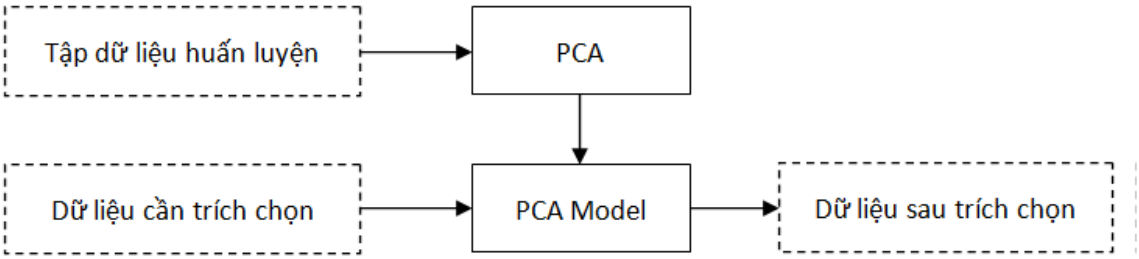
```
// 23 bones
string[] boneNames = new string[] { "root", "lowerback", "upperback", "thorax",
"lowerneck", "upperneck", "head", "rclavicle", "lclavicle", "rhumerus",
"lhumerus", "rfemur", "lfemur", "rradius", "lradius", "rtibia", "ltibia",
"lwrist", "rwrist", "lhand", "rhand", "lfoot", "rfoot" };
```

Bằng việc so sánh kết quả giữa hai nhóm xương liền kề, chúng ta có thể thấy được sự ảnh hưởng của các xương được thêm vào trong bài toán nhận dạng hành động, qua đó sẽ lựa chọn được tập con thích hợp trong tất cả các xương.

2.2. PCA

PCA là phương pháp thích hợp để ứng dụng trong vectơ dữ liệu có số chiều lớn như dữ liệu của bài toán nhận dạng hành động. Vì dữ liệu đã được xử lý ở giai đoạn trước nên việc áp dụng PCA vào trong bài toán nhận dạng hành động hoàn toàn giống như áp dụng trên các bài toán khác. Vấn đề cần giải quyết ở đây là tìm số lượng vectơ riêng (hay nói cách khác là số chiều của dữ liệu sau trích chọn) thích hợp sao cho độ chính xác của mô hình là lớn nhất.Vấn đề này có thể đơn giản được giải quyết bằng thực nghiệm. Chương trình sẽ thử lần lược các giá trị từ nhỏ đến lớn cho đến khi tìm được giá trị thích hợp.

Hình 2.6 mô tả mô hình trích chọn đặc tính dùng PCA. Từ tập dữ liệu huấn luyện, qua PCA ta sẽ thu được mô hình trích chọn PCA. Ta dùng mô hình này để làm giảm số lượng thuộc tính của mọi dữ liêu đầu vào sau khi đã được xử lý ở bước tiền xử lý.



Hình 2.6 Mô hình trích chọn đặc tính dùng PCA

Đầu vào của mô hình PCA ngoài dữ liệu cần trích chọn, còn có một tham số quan trọng nữa là số lượng thuộc tính hay số chiều dữ liệu sau trích chọn. Tham số này sẽ có giá trị khác nhau ở các lần thực nghiệm khác nhau. Quá trình xây dựng mô hình PCA được mô tả bằng đoạn mã nguồn sau:

```
class PCA
{
    KernelPrincipalComponentAnalysis PCAModel;
    public PCA(double[,] trainingData)
    {
        // Create a new linear kernel
        IKernel kernel = new Linear();
        // Creates the Kernel Principal Component Analysis of the given data
        PCAModel = new KernelPrincipalComponentAnalysis(trainingData, kernel);
        // Compute the Kernel Principal Component Analysis
        PCAModel.Compute();
    }
    public double[] Transform(double[] data, int dimensions)
    {
        return PCAModel.Transform(data, dimensions);
    }
}
```

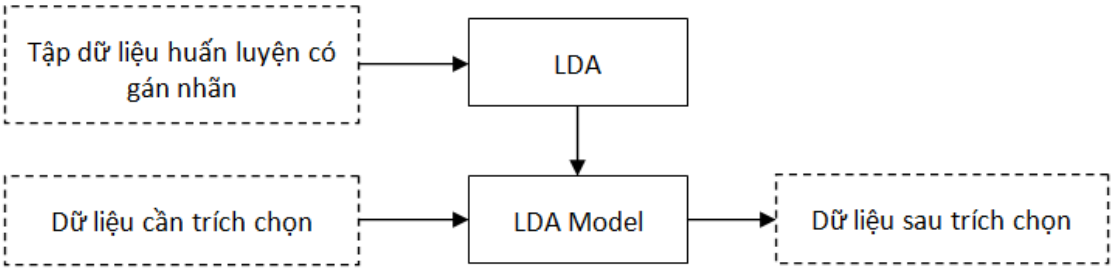
Đoạn mã sau mô tả việc sử dụng mô hình PCA xây dựng ở trên để trích chọn đặc tính dữ liệu:

```
// Create PCA model
PCA pca = new PCA(trainingData);
// Transfrom data
double[] result = pca.Transform(data, dimensions);
```

2.3. LDA

Khác với PCA, LDA không chỉ là một phương pháp trích chọn đặc tính thông thường mà còn là một thuật toán phân loại, nghĩa là nó có khả năng phân loại dữ liệu. Do đó LDA cần một bộ dữ liệu huấn luyện (có gẫn nhãn) cho thuật toán trích chọn đặc tính. Nếu như tham số cần tìm của PCA là số vectơ riêng thì với LDA đó là số lượng các đặc trưng khác nhau của dữ liệu. Tham số này cũng được xác định thông qua thực nghiệm.

Hình 2.7 mô tả mô hình trích chọn đặc tính dùng LDA. Dữ liệu đầu vào của mô hình là dữ liệu đã gán nhãn. LDA sẽ sử dụng tập dữ liệu có gán nhãn này để phân tích và xây dựng mô hình trích chọn đặc tính. Dữ liệu hành động sau bước tiền xử lý sẽ được đưa vào mô hình này để biến đổi sang không gian khác có ít chiều hơn.



Hình 2.7 Mô hình trích chọn đặc tính dùng LDA

Mô hình LDA có đầu vào là dữ liệu cần trích chọn và số chiều trong không gian dữ liệu mới. Đoạn mã sau mô tả quá trình cài đặt xây dựng mô hình LDA:

```
class LDA
{
    private KernelDiscriminantAnalysis LDAModel;
    public LDA(double[,] trainingData, int[] dataLabel)
    {
       // use linear kernel.
       IKernel kernel = new Linear();
       // Then, we will create a KDA using this linear kernel.
       LDAModel = new KernelDiscriminantAnalysis(trainingData, dataLabel, kernel);
       LDAModel.Compute(); // Compute the analysis
    }
    public double[] transform(double[] data, int dimensions)
    {
       return LDAModel.Transform(data, dimensions);
    }
}
```

Đoạn mã sau mô tả việc sử dụng mô hình LDA xây dựng ở trên để trích chọn đặc tính dữ liệu:

```
// Create LDA model
LDA lda = new LDA(trainingData, dataLabel);
// Transform data
double[] result = lda.transform(data, dimensions);
```

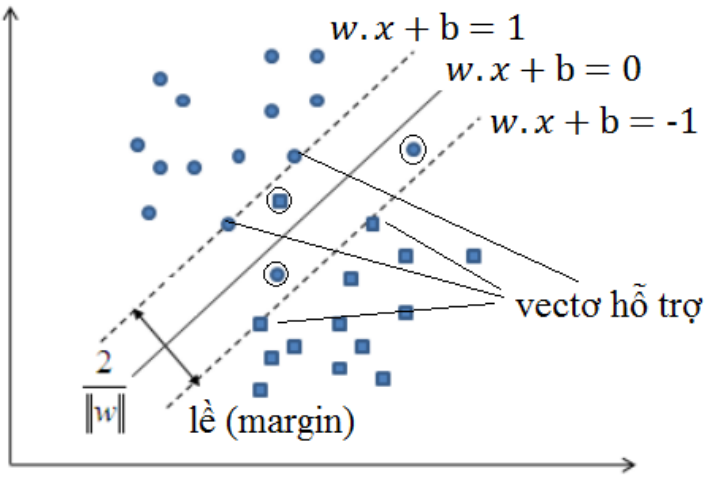
Trong bước xây dựng mô hình, khác với PCA, LDA cần biết nhãn của tập dữ liệu huấn luyện. Ngoài khả năng trích chọn đặc tính, LDA còn có thể được dùng trong phân loại dữ liệu, tuy nhiên luận văn này chỉ nghiên cứu khía cạnh trích chọn đặc tính của LDA.

3. Học máy

Giải thuật học máy sử dụng trong luận văn là SVM. Thông thường, thuật toán thường được dùng trong các bài toán có dữ liệu không đồng bộ về độ dài các vectơ như nhận dạng hành động, nhận dạng tiếng nói là HMM (Hidden Markov Model) hoặc là một dạng kết hợp giữa SVM và HMM. Tuy nhiên với dữ liệu đã được chuẩn hóa như trình bày ở các mục trên, chúng ta hoàn toàn có thể áp dụng giải thuật SVM trong quá trình huấn luyện.

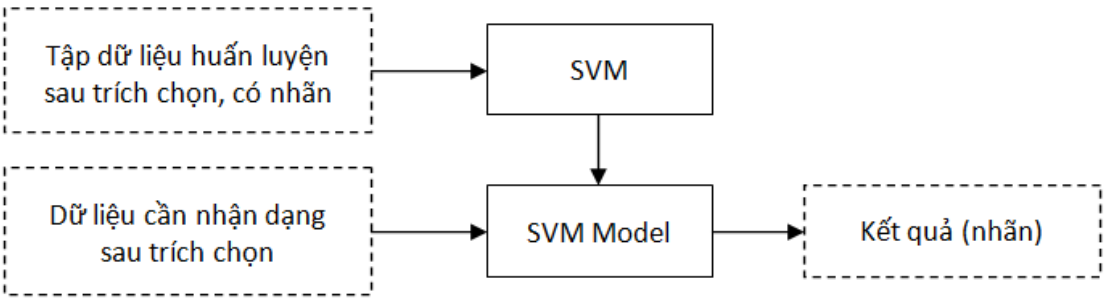
Về cơ bản, SVM được dùng cho các bài toán tuyến tính, có nghĩa là tồn tại một siêu phẳng tối ưu và lề cực đại phân chia hoàn toàn hai các lớp dữ liệu. Tuy nhiên trong rất nhiều trường hợp miền trong của lề trong tập dữ liệu huấn luyện có thể có một số lượng nhỏ các điểm, dẫn đến việc hông thể phân chia tập dữ liệu bằng một siêu phẳng tuyến tính. Hình 2.8 minh họa cho trường hợp này. Để giải quyết các bài toán phi tuyến tính người ta dùng một kỹ thuật để biến đổi siêu phẳng từ phi tuyến tính trở thành tuyến tính, phép biến đổi này gọi là hàm nhân. Một số hàm nhân thường dùng là hàm nhân

tuyến tính (linear kernel) và hàm nhân đa thức (polynomial kernel). Luận văn này sử dụng hàm nhân tuyến tính cho thuật toán SVM.



Hình 2.8 Không thể phân chia dữ liệu bằng một siêu phẳng tuyến tính

Dữ liệu đầu vào của học máy là tập dữ liệu huấn luyện sau khi đã được trích chọn đặc tính kèm theo nhãn của chúng. Tập dữ liệu này được SVM sử dụng để xây dựng mô hình nhận dạng. Hình 2.9 mô tả các bước xử lý trong học máy.



Hình 2.9 Mô hình học máy

Đoạn mã sau mô tả việc sử dụng hàm nhân và dữ liệu huấn luyện có gán nhãn để xây dựng mô hình nhận dạng.

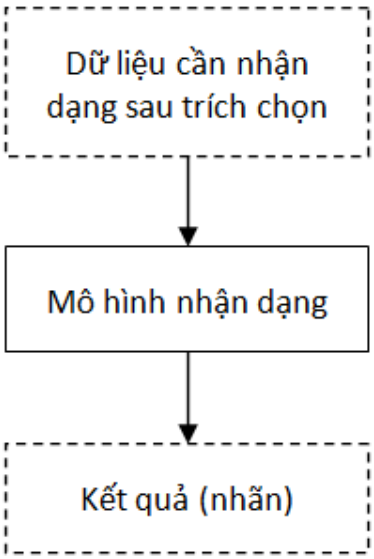
```
// Create a new Linear kernel
IKernel kernel = new Linear();
// Create a new Multi-class Support Vector Machine
var machine = new MulticlassSupportVectorMachine(dimensions, kernel, numClasses);
// Create the Multi-class learning algorithm for the machine
var teacher = new MulticlassSupportVectorLearning(machine, trainingData, labels);
// Configure the learning algorithm to use SMO to train the
//  underlying SVMs in each of the binary class subproblems.
teacher.Algorithm = (svm, classInputs, classOutputs, i, j) =>
    new SequentialMinimalOptimization(svm, classInputs, classOutputs);
// Run the learning algorithm
teacher.Run();
```

Như đã trình bày ở trên, luận văn này sử dụng hàm nhân tuyến tính và thuật toán tối thiểu tuần tự (Sequential minimal optimization - SMO) trong SVM đa lớp để xây dựng mô hình nhận dạng. Học máy cũng sử dụng dữ liệu huấn luyện có gán nhãn như trong LDA, tuy nhiên dữ liệu huấn luyện ở đây (`trainingData`) là tập dữ liệu sau khi đã được trích chọn đặc tính bằng các phương pháp trích chọn đặc tính. Sau khi áp dụng và thực thi thuật toán huấn luyện SMO (`teacher.Run();`) mô hình nhận dạng thu được chính là đối tượng `machine`. Đối tượng này sẽ được sử dụng để nhận dạng các loại hành động sau này.

4. Mô hình nhận dạng

Mô hình nhận dạng ở đây chính là mô hình xây dựng được từ dữ liệu huấn luyện sau khi áp dụng giải thuật học máy. Với mỗi phương pháp trích chọn đặc tính sẽ có một mô hình nhận dạng tương ứng. Mô hình nhận dạng sẽ được sử dụng bởi luồng dữ liệu kiểm định và luồng dữ liệu kiểm thử.

Hình 2.10 mô tả sự tương tác của mô hình nhận dạng với dữ liệu.



Hình 2.10 Mô tả mô hình nhận dạng

Tuy đã có thể cho ra kết quả nhận dạng, nhưng *mô hình nhận dạng* không phải là khối chức năng cuối cùng của hệ thống đề xuất. Các mô hình nhận dạng sẽ được thực nghiệm bằng dữ liệu thực nghiệm, độ chính xác của mỗi mô hình sẽ là dữ liệu đầu vào cho khối chức năng cuối cùng chính là *phương pháp trọng số*. Đối với dữ liệu kiểm thử hay dữ liệu mới bất kỳ, tất cả các mô hình nhận dạng lần đều được sử dụng để tìm nhãn của hành động. Các kết quả này sau đó cũng được đưa vào *phương pháp trọng số* để tổng hợp. Phần tiếp theo sẽ trình bày chi tiết về lý luận và cài đặt phương pháp trọng số.

5. Phương pháp trọng số

Trọng số là độ chính xác của các mô hình nhận dạng sau khi kiểm định bằng dữ liệu kiểm định. Ví dụ, với một hành động cần nhận dạng, nếu mỗi phương pháp cho ra một kết quả khác nhau thì kết quả cuối cùng được chọn là kết quả của phương pháp có trọng số cao nhất. Trong trường hợp khác, nếu có hai hay nhiều phương pháp cho ra cùng một hành động thì xác suất để chọn

hành động đó làm kết quả cuối cùng bằng tổng các trọng số. Phương pháp trọng số nhận đầu vào là các tỉ lệ nhận dạng thu được khi sử dụng dữ liệu kiểm định và kết quả của mỗi mô hình nhận dạng. Đầu ra của phương pháp trọng số là nhãn của hành động cần nhận dạng.

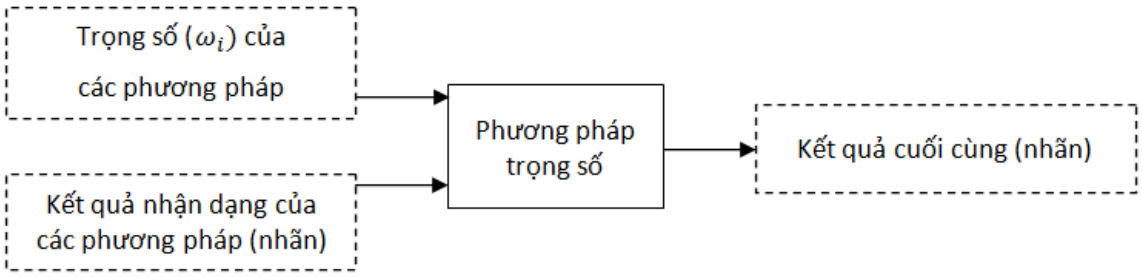
Cụ thể hàm xác suất và cách gán nhãn cho hành động cần nhận dạng được mô tả như sau: Giả sử ta có n phương pháp trích chọn đặc tính và m loại hành động khác nhau (cụ thể trong luận văn này n = 3, m = 4). Các phương pháp được đánh số thứ tự từ 1 đến n; các loại hành động được gán nhãn từ 1 đến m. Gọi f_i là nhãn của hành động nhận dạng được từ mô hình sử dụng phương pháp trích chọn đặc tính i, ω_i là trọng số của phương pháp trích chọn đặc tính i, $i = \{1..n\}$, $f_i = \{1..m\}$. Gọi p_j là xác suất để gán nhãn j cho hành động cần nhận dạng, $j = \{1..m\}$. Khi đó xác suất p_j được xác định như sau:

$$\forall\, i \in \{1..n\}, p_{(f_i)} += \omega_i$$

Nhãn k cần tìm được xác định như sau:

$$\begin{cases} k = j \\ với\ p_j = \max\limits_{j=1..m}(p_j) \end{cases}$$

Hình 2.11 mô tả đầu vào và đầu ra của phương pháp trọng số.



Hình 2.11 Mô tả phương pháp trọng số

Đoạn mã sau mô tả cách cài đặt phương pháp trọng số khi đã có đầy đủ tỉ lệ nhận dạng của mỗi phương pháp và kết quả nhận dạng (nhãn) của mỗi phương pháp ứng với một hành động đầu vào bất kỳ.

```
private int doCombine(int[] f, double[] w)
{
    double[] p = new double[numClasses];
    int size = f.Length;
    for (int i = 0; i < size; i++)
    {
        p[f[i]] += w[i];
    }
    double maxp = p.Max();
    for (int i = 0; i < numberClasses; i++)
    {
        if (p[i] == maxp) return i;
    }
    return -1;
}
```

CHƯƠNG 3
THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

Quá trình thực nghiệm được chia làm hai giai đoạn. Giai đoạn thứ nhất xây dựng mô hình nhận dạng với dữ liệu đầu vào ứng với các phương pháp trích chọn đặc tính khác nhau. Dữ liệu huấn luyện được sử dụng trong quá trình xây dựng mô hình. Bên cạnh đó sử dụng dữ liệu kiểm định để kiểm tra mức độ chính xác của mô hình (tính theo phần trăm). Ở giai đoạn này, một số giá trị tham số đầu vào được thay đổi bằng thực nghiệm để tìm ra giá trị phù hợp cho kết quả (độ chính xác) cao nhất ứng với mỗi phương pháp. Kết quả thu được ở giai đoạn thực nghiệm này là độ chính xác của mô hình tương ứng với các phương pháp trích chọn đặc tính, kết quả này là dữ liệu đầu vào cho phương pháp trọng số.

Với các trọng số thu được trong giai đoạn thứ nhất, giai đoạn thực nghiệm thứ hai kết hợp tất cả các phương pháp trích chọn đặc tính theo tỉ lệ tương ứng để gán nhãn cho hành động cần nhận dạng. Lúc này dữ liệu kiểm thử sẽ được sử dụng để đánh giá kết quả. Các phần sau sẽ mô tả chi tiết hai giai đoạn thực nghiệm và kết quả thu được.

1. Môi trường thực nghiệm

1.1. Dữ liệu sử dụng

Như đã trình bày ở các phần trước, dữ liệu sử dụng trong các giai đoạn thực nghiệm là dữ liệu chọn lọc từ cơ sở dữ liệu của CMU Mocap. Luận văn này chỉ sử dụng dữ liệu thuộc bốn hành động sau: chạy (run), đi (walk), nhảy (jump) và khiêu vũ (dance). Dữ liệu sau khi chọn lọc được chia ngẫu nhiên

thành ba nhóm khác nhau: dữ liệu huấn luyện, dữ liệu kiểm định và dữ liệu kiểm thử. Bảng 3.1 thống kê số lượng dữ liệu của từng hành động trong mỗi nhóm.

Bảng 3.1 Thống kê số lượng dữ liệu

Hành động	**run**	**walk**	**jump**	**dance**
Dữ liệu huấn luyện	49	90	48	55
Dữ liệu kiểm định	39	90	40	45
Dữ liệu kiểm thử	10	20	12	12
Tất cả	98	200	100	112
Tổng số	510			

1.2. Môi trường triển khai

Quá trình thực nghiệm được thực hiện trên một máy tính xách tay sử dụng hệ điều hành Microsoft Windows 7 (64bit) có cấu hình tương đương một máy tính phổ thông hiện nay.

Ngôn ngữ sử dụng trong chương trình thực nghiệm là ngôn ngữ lập trình C#. Bên cạnh đó, chương trình thực nghiệm có sử dụng mộ số thư viện mã nguồn mở như Accord.NET[15], AForge.NET[16].

2. Các giai đoạn thực nghiệm

Quá trình thực nghiệm được chia làm hai giai đoạn, giai đoạn thứ nhất xây dựng mô hình nhận dạng, giai đoạn thứ hai kiểm tra độ chính xác của mô

[15] http://accord-framework.net/
[16] http://www.aforgenet.com/

hình đã xây dựng và đánh giá kết quả. Phần sau trình bày chi tiết các bước trong mỗi giai đoạn.

2.1. Giai đoạn thứ nhất

Mục đích của giai đoạn này là tìm các tham số thích hợp cho mỗi phương pháp trích chọn đặc tính để mô hình nhận dạng xây dựng được có độ chính xác cao nhất. Độ chính xác của mỗi mô hình được kiểm nghiệm bằng tập dữ liệu kiểm định. Tất cả các kết quả thực nghiệm trong giai đoạn này được trình bày dưới dạng bảng. Ứng với mỗi phương pháp trích chọn đặc tính sẽ có hai loại bảng.

Loại thứ nhất thể hiện tỉ lệ nhận dạng của các nhóm hành động ứng với các giá trị tham số khác nhau. Mỗi dòng của bảng này là kết quả nhận dạng với một giá trị của tham số. Các ô trong mỗi dòng là tỉ lệ nhận dạng của một nhóm hành động ứng với giá trị tham số đó. Ví dụ trong bảng 3.2, giá trị 96.0% là tỉ lệ nhận dạng đúng của nhóm hành động walk khi sử dụng 13 xương. Cột cuối cùng của loại bảng này là tỉ lệ nhận dạng trung bình của tất cả các nhóm hành động.

Loại thứ hai là bảng chi tiết của mỗi dòng trong loại bảng thứ nhất. Bảng này là một ma trận vuông với dòng là nhãn các nhóm hành động cần nhận dạng, cột là nhãn kết quả hành động nhận dạng được tương ứng. Ví dụ trong bảng 3.3, giá trị của dòng run, cột walk là 86.9%, nghĩa là 86.9% hành động thuộc nhóm run nhưng được gán nhãn walk sau khi nhận dạng. Đường chéo của bảng này chính là một dòng trong loại bảng thứ nhất. Khác với bảng thứ nhất chỉ thể hiện tỉ lệ nhận dạng đúng, bảng này chi tiết hơn, có cả tỉ lệ nhận dạng sai của mỗi nhóm hành động. Các giá trị trên đường chéo là tỉ lệ nhận dạng đúng, các giá trị còn lại là tỉ lệ nhận dạng sai.

Sau đây là các kết quả tương ứng cho từng phương pháp.

2.1.1. Lựa chọn thủ công

Đối với phương pháp lựa chọn thủ công, thông số cần xác định là số lượng các đoạn xương được chọn để mô hình SVM thuần túy cho ra kết quả khả quan nhất. Luận văn này thực nghiệm với năm nhóm dữ liệu khác nhau từ kết quả nghiên cứu của K. Adistambha. Nhóm 1 có 3 xương: root, lowerback và upperback. Nhóm 2 có 4 xương: root, lowerback, upperback và thorax. Nhóm 3 gồm có 7 xương: root, lowerback, upperback, thorax, lowerneck, upperneck và head. Nhóm 4 gồm có 13 xương: root, lowerback, upperback, thorax, lowerneck, upperneck, head, rclavicle, lclavicle, rhumerus, lhumerus, rfemur và lfemur. Nhóm 5 gồm có 23 xương: root, lowerback, upperback, thorax, lowerneck, upperneck, head, rclavicle, lclavicle, rhumerus, lhumerus, rfemur, lfemur, rradius, lradius, rtibia, ltibia, lwrist, rwrist, lhand, rhand, lfoot và rfoot.

Việc phân chia các nhóm dựa trên nguyên tắc nhóm sau là nhóm trước thêm vào một số xương khác. Việc thực nghiệm cũng theo thứ tự này, có nghĩa là bắt đầu với nhóm có ít xương nhất, sau đó thêm các xương vào để sinh ra nhóm mới. Bảng 3.2 là kết quả các lần thực nghiệm với các nhóm xương khác nhau. Ngoài 5 nhóm như đã trình bày, luận văn còn thực nghiệm với tất cả xương được chọn (29 xương). Trong bảng 3.2 còn có sự xuất hiện của một nhóm xương mới với 11 xương, không theo thứ tự nhỏ đến lớn. Nhóm này được chọn dựa trên kết quả các lần thực nghiệm trước đó, chi tiết xin xem phần giải thích sau bảng 3.7.

Bảng 3.2 Kết quả thực nghiệm với phương pháp lựa chọn thủ công

Số xương	Run	Walk	Jump	Dance	Trung bình
3	0.0%	93.3%	14.3%	18.2%	49.4%
4	0.0%	93.3%	14.3%	13.6%	48.8%
7	0.0%	94.7%	30.9%	13.6%	53.7%
13	0.0%	96.0%	28.5%	18.2%	54.3%
23	78.3%	98.7%	81.0%	31.9%	**82.1%**
11	78.3%	98.7%	81.0%	36.4%	**82.7%**
Tất cả (29)	78.3%	98.7%	81.0%	41.0%	**83.3%**

Kết quả cho thấy nếu chỉ chọn các xương trên trục xương sống của cơ thể (ở các nhóm 3, 4, 7, 13) thì các loại hành động sử dụng chân và tay như jump và dance sẽ bị mất các thuộc tính quan trọng, do đó kết quả nhận dạng trên các hoạt động này rất thấp. Với nhóm xương được bổ sung các xương ở chân và tay (nhóm có 23 xương) thì kết quả đã tốt hơn rất nhiều, xấp xỉ với tỉ lệ nhận dạng dùng toàn bộ các xương (81.2% so với 83.3%). Cá biệt, ta thấy các hành động thuộc nhóm dance luôn có tỉ lệ nhận dạng rất thấp (cao nhất là 41%). Lý do cho điều này là các động tác của dance có nhiều điểm tương đồng với run và walk cho nên hoạt động này thường được nhận dạng nhầm thành jump hoặc walk. Bảng chi tết với từng nhóm xương sẽ thể hiện rõ hơn điều đó. Bảng 3.3, 3.4, 3.5, 3.6 lần lượt là bảng ma trận chi tiết tương ứng với các nhóm có 3, 4, 7 và 13 xương.

Bảng 3.3 Kết quả chi tiết với nhóm có 3 xương

Hoạt động	Run	Walk	Jump	Dance
Run	**0.0%**	86.9%	8.7%	4.4%
Walk	0.0%	**93.3%**	5.4%	1.3%
Jump	0.0%	83.3%	**14.3%**	2.4%
Dance	0.0%	72.7%	9.1%	**18.2%**

Nhóm này chỉ sử dụng 3 đoạn xương sống ở lưng nên tỉ lệ nhận dạng thấp do rất nhiều đặc tính của các loại hành động đã bị loại bỏ. Dữ liệu sử dụng trong nhóm này chỉ có thể mô tả chuyển động tới, lui của cơ thể, do đó phần các loại hành động đều được nhận dạng thành walk. Không một hành động nào thuộc nhóm run được nhận dạng đúng mặc dù run và walk là hai chuyển động khá tương đồng, chỉ khác nhau về tốc độ di chuyển. Điều này cho thấy SVM không thích hợp trong các loại dữ liệu tuần hoàn.

Bảng 3.4 Kết quả chi tiết với nhóm có 4 xương

Hoạt động	Run	Walk	Jump	Dance
Run	**0.0%**	86.9%	8.7%	4.4%
Walk	0.0%	**93.3%**	5.4%	1.3%
Jump	0.0%	83.3%	**14.3%**	2.4%
Dance	0.0%	77.3%	9.1%	**13.6%**

Tương tự nhóm có 3 xương, nhóm có 4 xương vẫn cho kết quả rất thấp do dữ liệu không thể hiện đầy đủ đặc tính của các lớp. Trong trường hợp này, tỉ lệ của nhóm dance có thay đổi nhỏ nhờ sự góp mặt của thorax (vùng ngực) do động tác trong dance có sự chuyển động lên xuống của ngực.

Bảng 3.5 Kết quả chi tiết với nhóm có 7 xương

Hoạt động	Run	Walk	Jump	Dance
Run	**0.0%**	65.2%	30.4%	4.4%
Walk	0.0%	**94.7%**	4.0%	1.3%
Jump	0.0%	66.7%	**30.9%**	2.4%
Dance	0.0%	77.3%	9.1%	**13.6%**

Mặc dù đã thêm các xương vùng đầu và cổ nhưng kết quả vẫn chưa có sự thay đổi. Các xương này vẫn chuyển động tịnh tiến khá giống nhau trong các loại hành động. Chỉ có walk và jump phụ thuộc vào các đặc tính của nhóm xương này. Các đặc tính của run và walk vẫn chưa xuất hiện.

Bảng 3.6 Kết quả chi tiết với nhóm có 13 xương

Hoạt động	Run	Walk	Jump	Dance
Run	**0.0%**	86.9%	8.7%	4.4%
Walk	0.0%	**96.0%**	2.7%	1.3%
Jump	0.0%	69.0%	**28.5%**	2.5%
Dance	0.0%	72.7%	9.1%	**18.2%**

Nhóm này có sự hiện diện của xương vai, xương cánh tay và xương đùi nhưng chuyển động của các xương này vẫn còn giống nhau trong các loại hành động nên kết quả vẫn giống với các lần thực nghiệm trước. Điều này cho thấy xương vai, cánh tay và xương đùi chưa thể hiện tốt sự khác nhau của các loại hành động. Khi số lượng các xương tăng lên, tỉ lệ nhận dạng có tăng nhưng chưa đáng kể. Tỉ lệ trung bình 54.3% của lần thực nghiệm này chưa thật sự có ý nghĩa. Với dữ liệu thực nghiệm hiện tại, số lượng hành động

trong các nhóm khác xa rất nhiều, do đó tỉ lệ nhận dạng chung là trung bình có trọng số, tỉ lệ này phụ thuộc vào tỉ lệ của nhóm hành động có nhiều thực thể nhất.

Bảng 3.7 là ma trận chi tiết lần thực nghiệm sử dụng nhóm 23 xương.

Bảng 3.7 Kết quả chi tiết với nhóm có 23 xương

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	21.7%	0.0%	0.0%
Walk	0.0%	**98.7%**	0.0%	1.3%
Jump	0.0%	19.0%	**81.0%**	0.0%
Dance	9.1%	50.0%	9.0%	**31.9%**

Rõ ràng tỉ lệ nhận dạng ở nhóm 23 xương được cải thiện đáng kể so với các nhóm trước. Đây là kết quả của việc bổ sung các chuyển động của cẳng tay, cẳng chân, cổ tay, bàn tay, và bàn chân. Vì vậy có thể kết luận rằng các chuyển động ở tay và chân ảnh hưởng rất lớn trong việc trích chọn đặc tính của dữ liệu chuyển động 3D.

Sau kết quả của lần thực nghiệm với nhóm 23 xương, nhận thấy tầm ảnh hưởng lớn của các chuyển động ở tay chân trong việc phân loại các hành động, luận văn đã thực hiện thêm một lần thực nghiệm với 11 xương bao gồm root đại diện cho nhóm chuyển động tịnh tiến và 10 xương được bổ sung trong nhóm 23 xương ở trên. Cụ thể 11 xương đó là: root, rradius, lradius, rtibia, ltibia, lwrist, rwrist, lhand, rhand, lfoot, rfoot. Bảng 3.8 là kết quả của lần thực nghiệm thêm này.

Bảng 3.8 Kết quả chi tiết khi sử dụng 11 xương

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	8.7%	13.0%	0.0%
Walk	0.0%	**98.7%**	1.3%	0.0%
Jump	0.0%	19.0%	**81.0%**	0.0%
Dance	4.5%	45.5%	13.6%	**36.4%**

Tỉ lệ nhận dạng đúng trung bình thu được ở lần thực nghiệm với nhóm 11 xương là **82.7%**, chỉ cao hơn 0.6% so với nhóm 23 xương nhưng số thuộc tích giảm đi một nữa. Đây là một phát hiện đáng giá. Tuy nhiên vẫn thấp hơn một chút so với việc sử dụng tất cả các xương. Điều này cho thấy vẫn có một số lượng nhỏ sự ảnh hưởng của các xương không được chọn với việc phân loại hành động. Bảng 3.9 là kết quả chi tiết khi thực nghiệm sử dụng tất cả xương.

Bảng 3.9 Kết quả chi tiết khi sử dụng tất cả xương

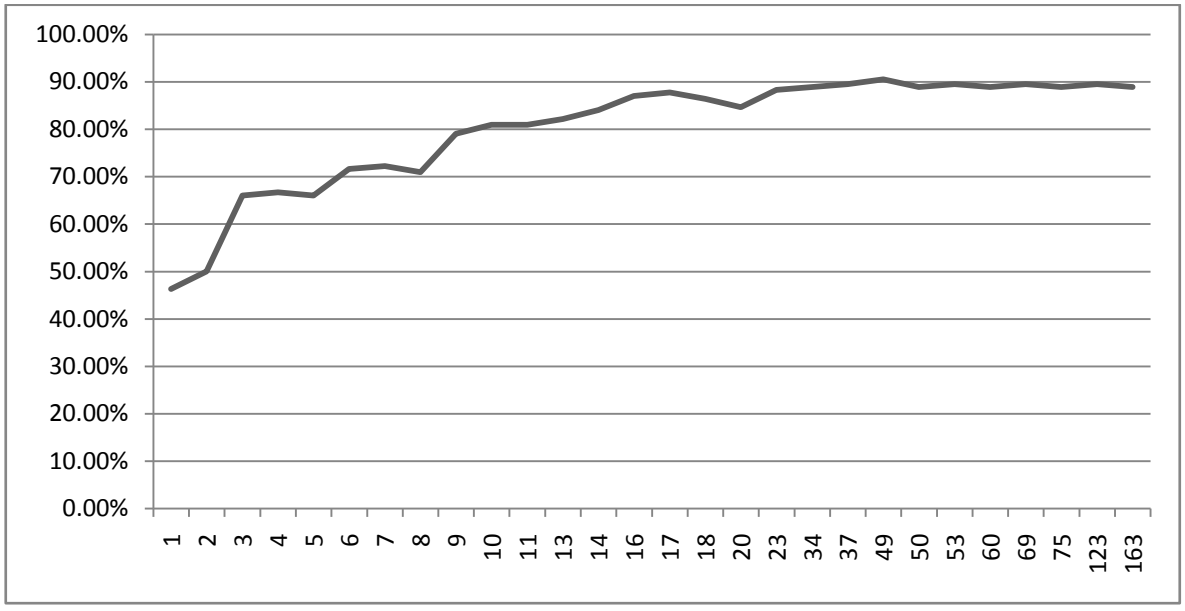
Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	21.7%	0.0%	0.0%
Walk	0.0%	**98.7%**	1.3%	0.0%
Jump	0.0%	19.0%	**81.0%**	0.0%
Dance	0.0%	50.0%	9.0%	**41.0%**

Với việc sử dụng tất cả các thuộc tính (tất cả các xương) thì kết quả thu được là tỉ lệ nhận dạng đúng trung bình cao nhất. Tuy nhiên không khác biệt nhiều so với nhóm 23 xương và nhóm 11 xương vừa thêm vào cuối giai đoạn thực nghiệm (83.3% so với 82.1% và 82.7%). Vì vậy có thể sử dụng nhóm dữ

liệu gồm 11 xương như một phương pháp lựa chọn đặc tính trong việc xây dựng mô hình nhận dạng hành động người trong không gian 3D. Tuy kết quả nhận dạng trung bình khá cao nhưng đối với các loại hành động phức tạp như dance, đặc tính của dữ liệu không thể hiện rõ trong không gian hiện tại, dẫn tới tỉ lệ nhận dạng đúng rất thấp (41.0%). Các đặc tính của hành động phức tạp này sẽ thể hiện rõ hơn nếu ta áp dụng các phương pháp biến đổi để tìm ra một không gian mới, mà ở đó độ biến thiên của dữ liệu là cao nhất. PCA và LDA là hai thuật toán phổ biến có thể thực hiện điều này. Phần tiếp theo là kết quả thực nghiệm sử dụng các phương pháp PCA và LDA.

2.1.2. Phương pháp PCA

Đối với phương pháp PCA, thông số cần xác định là số lượng các vectơ riêng (chính là số chiều trong không gian mới) để mô hình nhận dạng có độ chích xác cao nhất. Hình 3.1 mô tả sự biến thiên của tỉ lệ nhận dạng dùng phương pháp trích chọn đặc tính PCA khi số chiều thay đổi.



Hình 3.1 Sự biến thiên của tỉ lệ nhận dạng trong PCA

Ta thấy với số chiều tăng dần, độ chính xác tăng theo hình răng cưa đến một giá trị ngưỡng (90.1% với số chiều bằng 49), sau đó bắt đầu giảm và dần như trở thành đường thẳng khi số chiều vượt qua ngưỡng. Kết quả rõ hơn được trình bày trong bảng 3.10. Các bảng tiếp theo từ 3.11 đến 3.16 là bảng kết quả chi tiết các lần thử nghiệm với số chiều khác nhau trong PCA.

Bảng 3.10 Kết quả thực nghiệm với các giá trị khác nhau của số chiều dữ liệu sau trích chọn trong PCA

Số chiều trong PCA	Run	Walk	Jump	Dance	Trung bình
1	0.0%	100%	0.0%	0.0%	46.3%
2	0.0%	94.7%	14.3%	18.2%	50.0%
3	60.9%	80.0%	69.0%	18.2%	66.0%
4	78.3%	77.3%	66.7%	18.2%	66.7%
5	78.3%	76.0%	66.7%	18.2%	66.0%
6	78.3%	85.3%	69.0%	22.7%	71.6%
7	78.3%	84.0%	73.8%	22.7%	72.2%
9	78.3%	92.0%	69.0%	54.5%	79.0%
11	78.3%	94.6%	69.0%	71.4%	80.9%
13	78.3%	96.0%	73.8%	54.5%	82.1%
14	78.3%	96.0%	76.2%	63.6%	84.0%
16	78.3%	97.3%	85.7%	63.6%	87.0%

17	78.3%	97.3%	85.7%	68.2%	87.7%
18	78.3%	94.7%	85.7%	68.2%	86.4%
20	78.3%	94.7%	83.3%	59.0%	84.6%
23	78.3%	96.0%	85.7%	77.3%	88.3%
34	78.3%	96.0%	85.7%	81.8%	88.9%
37	78.3%	97.3%	85.7%	81.8%	89.5%
49	78.3%	98.7%	85.7%	81.8%	**90.1%**
50	78.3%	96.0%	85.7%	81.8%	88.9%
53	78.3%	97.3%	85.7%	81.8%	89.5%
60	78.3%	96.0%	85.7%	81.8%	88.9%
69	78.3%	97.3%	85.7%	81.8%	89.5%
75	78.3%	96.0%	85.7%	81.8%	88.9%
123	78.3%	97.3%	85.7%	81.8%	89.5%
163	78.3%	96.0%	85.7%	81.8%	88.9%

Trong các lần thực nghiệm với số chiều nhỏ, tất cả các loại hành động đều được nhận dạng thành walk. Khi số chiều tăng lên, tỉ lệ của các hành động khác cũng dần tăng theo. Đầu tiên là nhóm hành động walk, sau đó là jump và cuối cùng là dance. Ở các lần thực nghiệm có số chiều nhỏ, tỉ lệ nhận dạng đúng của nhóm dance rất thấp, điều này cho thấy dance là hoạt động phức tạp và có nhiều động tác giống với các hoạt động còn lại. Một điều đặc biệt nữa là tỉ lệ nhận dạng không tăng dần hay giảm dần mà tăng, giảm theo

hình răng cưa. Tuy nhiên nếu làm mịn hơn thì biểu đồ tăng dần đến một điểm nhất định, sau đó quay đầu giảm.

Bảng 3.11 Kết quả nhận dạng chi tiết với số chiều bằng 1 trong PCA

Hoạt động	Run	Walk	Jump	Dance
Run	**0.0%**	100%	0.0%	0.0%
Walk	0.0%	**100%**	0.0%	0.0%
Jump	0.0%	100%	**0.0%**	0.0%
Dance	0.0%	100%	0.0%	**0.0%**

Việc chỉ chọn một vectơ riêng rõ ràng không đảm bảo việc biểu diễn độ biến thiên của dữ liệu. Do đó mọi hành động đều được gán nhãn walk.

Bảng 3.12 Kết quả nhận dạng chi tiết với số chiều bằng 2 trong PCA

Hoạt động	Run	Walk	Jump	Dance
Run	**0.0%**	87.0%	8.7%	4.3%
Walk	0.0%	**94.7%**	4.0%	1.3%
Jump	0.0%	83.3%	**14.3%**	2.4%
Dance	0.0%	81.8%	0.0%	**18.2%**

Khi số chiều tăng lên bằng 2, đã xuất hiện kết quả nhận dạng rơi vào các nhóm hoạt động khác. Tỉ lệ nhận dạng đúng lúc này sẽ tỉ lệ thuận với số chiều.

Bảng 3.13 Kết quả nhận dạng chi tiết với số chiều bằng 11 trong PCA

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	13.04%	8.7%	0.0%
Walk	0.0%	**94.7%**	4.0%	1.3%
Jump	0.0%	28.6%	**71.4%**	0.0%
Dance	0.0%	27.3%	18.2%	**54.5%**

Với số chiều bằng 11, có vẻ như tất cả các đặc trưng đã được thể hiện trong mọi nhóm hành động. Tỉ lệ nhận dạng đúng đã tăng đáng kể với các lần thực nghiệm trước. Tuy nhiên với hoạt động có các động tác phức tạp như dance, tỉ lệ vẫn còn thấp.

Bảng 3.14 Kết quả nhận dạng chi tiết với số chiều bằng 34 trong PCA

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	17.4%	4.3%	0.0%
Walk	0.0%	**96.0%**	2.7%	1.3%
Jump	0.0%	14.3%	**85.7%**	0.0%
Dance	0.0%	4.6%	13.6%	**81.8%**

Khi tăng số chiều lên 34, kết quả nhận dạng trung bình lúc này là 88.9%. Các tỉ lệ nhận dạng thành phần đều tăng so với các lần thử nghiệm trước. Hành động phức tạp như dance lúc này cũng đã tìm được đặc trưng riêng so với các hoạt động khác. Kết quả này cũng cao hơn so với việc chọn lựa thủ công (88.9% so với 83.3%). Có thể nói lúc này phương pháp PCA đã bắt đầu cho thấy tính hiệu quả.

Bảng 3.15 Kết quả nhận dạng chi tiết với số chiều bằng 49 trong PCA

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	17.4%	4.3%	0.0%
Walk	0.0%	**98.7%**	0.0%	1.3%
Jump	0.0%	14.3%	**85.7%**	0.0%
Dance	0.0%	9.1%	9.1%	**81.8%**

Đây là lần thử nghiệm có kết quả cao nhất đối với phương pháp PCA. Đặc biệt tỉ lệ nhận dạng đúng của nhóm hành động walk gần như tuyệt đối, chỉ có hơn 1% tỉ lệ nhận dạng sai. Kết quả lần này không giống với tỉ lệ 100% ở bảng 3.11 mà là tỉ lệ thực tế. Nếu như ở bảng 3.11 tất cả hành động đều được gán nhãn walk, do đó khi thống kê sẽ thấy tỉ lệ là 100% với walk nhưng thật ra tỉ lệ đó không thể hiện đúng bản chất của nó. Ở lần thực nghiệm này, tỉ lệ nhận dạng đúng của tất cả các nhóm hành động đều cao. Điều đó cho thấy với số chiều bằng 49, đặc trưng của các loại hành động được thể hiện rõ nhất.

Bảng 3.16 Kết quả nhận dạng chi tiết với số chiều bằng 50 trong PCA

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	17.4%	4.3%	0.0%
Walk	0.0%	**96.0%**	2.7%	1.3%
Jump	0.0%	14.3%	**85.7%**	0.0%
Dance	0.0%	4.6%	13.6%	**81.8%**

Khi đã vượt qua ngưỡng, tỉ lệ đã có chiều hướng giảm khi ta tiếp tục tăng số chiều dữ liệu. Lúc này tất cả đặc trưng đã được thể hiện, vì thế nếu ta tăng số chiều thì sẽ dẫn tới trường hợp nhiễu hoặc dư thừa dữ liệu.

2.1.3. Phương pháp LDA

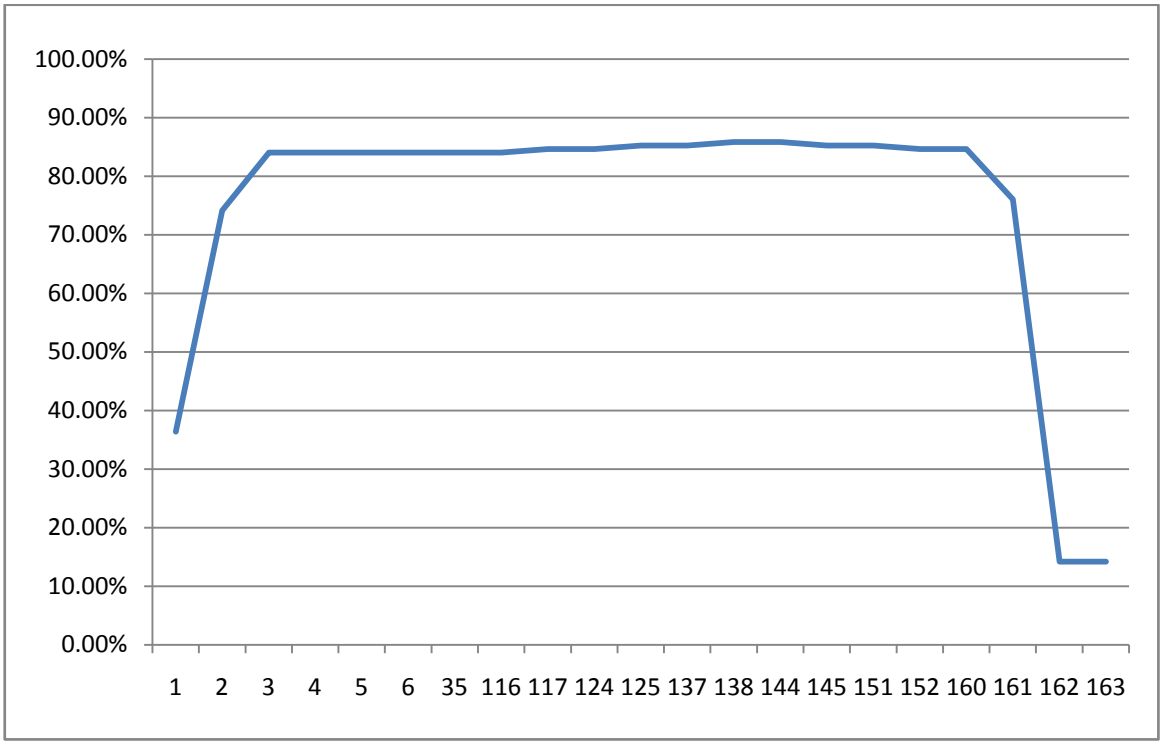
Tương tự như PCA, phương pháp LDA cũng cần xác định số chiều dữ liệu sau trích chọn để mô hình nhận dạng xây dựng được có độ chính xác cao nhất. Bảng 3.17 tổng hợp kết quả từ các lần thử nghiệm với các giá trị khác nhau của thông số này. Các bảng tiếp theo từ 3.18 đến 3.23 là bảng kết quả chi tiết các lần thử nghiệm với số chiều khác nhau trong LDA.

Bảng 3.17 Kết quả thực nghiệm với các giá trị khác nhau của số chiều dữ liệu sau trích chọn trong LDA

Số chiều trong LDA	Run	Walk	Jump	Dance	Trung bình
1	52.1%	6.7%	57.1%	81.8%	36.4%
2	69.6%	64.0%	90.5%	81.8%	74.1%
3	73.9%	84.0%	92.9%	77.3%	84.0%
4	73.9%	84.0%	92.9%	77.3%	84.0%
5	73.9%	84.0%	92.9%	77.3%	84.0%
6	73.9%	84.0%	92.9%	77.3%	84.0%
35	73.9%	84.0%	92.9%	77.3%	84.0%
116	73.9%	84.0%	92.9%	77.3%	84.0%
117	73.9%	85.3%	92.9%	77.3%	84.6%
124	73.9%	85.3%	92.9%	77.3%	84.6%
125	78.3%	85.3%	92.9%	77.3%	85.2%
137	78.3%	85.3%	92.9%	77.3%	85.2%

138	78.3%	86.7%	92.9%	77.3%	**85.8%**
144	78.3%	86.7%	92.9%	77.3%	85.8%
145	78.3%	86.7%	92.9%	72.7%	85.2%
151	78.3%	86.7%	92.9%	72.7%	85.2%
152	78.3%	85.3%	92.9%	72.7%	84.6%
160	78.3%	85.3%	92.9%	72.7%	84.6%
161	78.3%	88.0%	92.9%	0.0%	76.0%
162	100%	0.0%	0.0%	0.0%	14.2%
163	100%	0.0%	0.0%	0.0%	14.2%

Ta thấy với LDA, ở những lần thí nghiệm với số chiều nhỏ đã không xảy ra hiện tượng sai cục bộ. Có nghĩa là tỉ lệ nhận dạng được phân bố đều, không dồn vào một loại hành động nào. Một khác biệt nữa của LDA so với PCA là hoạt động nào càng phức tạp, đặc trưng càng dễ thể hiện với số chiều nhỏ. Như trong bảng 3.17 hành động trong nhóm dance thường gây khó khăn cho phương pháp PCA thì ở đây tỉ lệ nhận dạng đúng cho dance rất cao (81.8%) với số chiều bằng 1. Điều đáng tiếc là khi số chiều tăng lên thì tỉ lệ này lại giảm dần. Hình 3.2 thể hiện sự biến thiên của tỉ lệ nhận dạng trong LDA.



Hình 3.2 Sự biến thiên của tỉ lệ nhận dạng trong LDA

Sự biến thiên của tỉ lệ nhận dạng trung bình trong LDA hoặc là tăng dần, hoặc là giảm dần, không xuất hiện đường răng cưa như ở PCA.

Bảng 3.18 Kết quả nhận dạng chi tiết với số chiều bằng 1 trong LDA

Hoạt động	Run	Walk	Jump	Dance
Run	**52.1%**	4.4%	39.1%	4.4%
Walk	45.3%	**6.7%**	44.0%	4.0%
Jump	30.9%	11.8%	**57.1%**	0.0%
Dance	13.6%	0.0%	4.6%	**81.8%**

Đây là lần thực nghiệm với số chiều nhỏ nhất nhưng kết quả vẫn rất tốt. Ít nhất với nhóm hành động dance, tỉ lệ nhận dạng đúng là rất cao, 81.8%.

Điều đáng nói ở đây là LDA đã phân ly tốt dữ liệu, chỉ với 1 phép chiếu mà đã có thể thể hiện đặc trưng của dữ liệu phức tạp như dance.

Bảng 3.19 Kết quả nhận dạng chi tiết với số chiều bằng 2 trong LDA

Hoạt động	Run	Walk	Jump	Dance
Run	**69.6%**	8.7%	17.4%	4.3%
Walk	2.7%	**64.0%**	30.7%	2.6%
Jump	0.0%	9.5%	**90.5%**	0.0%
Dance	13.6%	0.0%	4.6%	**81.8%**

Có thể dễ dàng nhận thấy chỉ cân tăng số chiều lên 1 nhưng kết quả nhận dạng trung bình được cải thiện đáng kể, tăng từ 36.4% lên 74.1%. Như vậy các đặc tính đặc trưng nhất của mỗi loại hành động được trích chọn trước.

Bảng 3.20 Kết quả nhận dạng chi tiết với số chiều bằng 3 trong LDA

Hoạt động	Run	Walk	Jump	Dance
Run	**73.9%**	13.1%	8.7%	4.3%
Walk	2.7%	**84.0%**	10.7%	2.6%
Jump	0.0%	7.1%	**92.9%**	0.0%
Dance	18.2%	45.5%	0.0%	**77.3%**

Dựa vào kết quả ở lần thực nghiệm này có thể thấy LDA tỏ ra rất hiệu quả trong việc làm giảm số chiều của dữ liệu. Chỉ với số chiều bằng 3, LDA đã thể hiện được tất cả các đặc trưng của dữ liệu. Tỉ lệ nhận dạng trung bình lần này là 84%, cao hơn so với lần thực nghiệm sử dụng toàn bộ thuộc tính của dữ liệu (83.3%).

Bảng 3.21 Kết quả nhận dạng chi tiết với số chiều bằng 125 trong LDA

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	13.0%	4.3%	4.4%
Walk	1.3%	**85.3%**	10.7%	2.7%
Jump	0.0%	7.1%	**92.9%**	0.0%
Dance	13.6%	4.5%	4.6%	**77.3%**

Kể từ lần thực nghiệm với số chiều bằng 3, mãi đến lần này tỉ lệ nhận dạng mới có sự thay đổi nhỏ, tăng từ 84% lên 85.2%. Điều này cho thấy còn rất ít các đặc trưng ảnh hưởng đến quá trình phân ly dữ liệu.

Bảng 3.22 Kết quả nhận dạng chi tiết với số chiều bằng 138 trong LDA

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	13.0%	4.3%	4.4%
Walk	0.0%	**86.7%**	10.7%	2.6%
Jump	0.0%	7.1%	**92.9%**	0.0%
Dance	9.1%	4.5%	9.1%	**77.3%**

Đây là lần thực nghiệm có kết quả nhận dạng đúng cao nhất của phương pháp LDA. Tuy nhiên số chiều lúc này đã lên tới 138. Kể từ lần này, càng tăng số chiều sẽ làm giảm tỉ lệ nhận dạng.

Bảng 3.23 Kết quả nhận dạng chi tiết với số chiều bằng 145 trong LDA

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	13.0%	4.3%	4.4%
Walk	1.3%	**85.3%**	10.7%	2.7%
Jump	0.0%	7.1%	**92.9%**	0.0%
Dance	13.6%	4.5%	4.6%	**77.3%**

Tỉ lệ nhận dạng đã bắt đầu giảm, nếu số chiều càng tăng thì thuộc tính nhiễu và thuộc tính dư thừa sẽ tăng, làm giảm độ chính xác cả mô hình.

2.2. Giai đoạn thứ hai

Với kết quả của giai đoạn thực nghiệm thứ nhất, ta thu được trọng số của các phương pháp trích chọn đặc tính cho mô hình nhận dạng kết hợp như mô tả trong bảng 3.24.

Bảng 3.24 Trọng số thu được sau giai đoạn kiểm định

Phương pháp	Tham số	Trọng số
Lựa chọn thủ công	11	0.83
PCA	49	0.9
LDA	138	0.86

Giai đoạn thực nghiệm thứ hai sử dụng trọng số thu được, kết hợp các mô hình nhận dạng được xây dựng từ các phương pháp trích chọn đặc tính với nhau. Sau đó dùng dữ liệu kiểm thử để kiểm tra độ chính xác của mô hình. Bảng 3.25 là kết quả thực nghiệm với mô hình nhận dạng kết hợp.

Bảng 3.25 Kết quả giai đoạn thực nghiệm thứ hai

Hoạt động	Run	Walk	Jump	Dance
Run	**78.3%**	17.4%	4.3%	0.0%
Walk	0.0%	**100%**	0.0%	0.0%
Jump	0.0%	14.3%	**85.7%**	0.0%
Dance	0.0%	9.1%	9.1%	**81.8%**

Khi kết hợp các phương pháp trích chọn đặc tính, tỉ lệ nhận dạng có tăng lên. Tỉ lệ nhận dạng của mô hình kết hợp là 90.7% cao hơn 0.6% so với phương pháp trích chọn đặc tính tốt nhất (90.1%).

3. Đánh giá

Với bài toán nhận dạng hành động trong không gian 3D, với phương pháp trích chọn đặc tính thủ công có thể sử dụng nhóm có 11 xương để xây dựng mô hình huấn luyện và nhận dạng.

Từ kết quả thực nghiệm với hai phương pháp trích chọn đặc tính PCA, LDA ta thấy: với cùng cở sở dữ liệu thì độ chính xác cao nhất của PCA là 90.1%, độ chính xác cao nhất của LDA là 85.8%. Như vậy trong trường hợp này, PCA cho kết quả tốt hơn LDA. Kết quả thực nghiệm cũng cho thấy PCA hiệu quả hơn trong việc tối ưu hóa việc thể hiện sự biên thiên của dữ liệu. Còn LDA hiệu quả hơn trong việc phân ly dữ liệu dựa vào sự đo lường các đặc trưng.

So sánh kết quả giữa áp dụng trích chọn đặc tính và không áp dụng trích chọn đặc tính có thể thấy sự khác biệt lớn. Vì vậy trích chọn đặc tính luôn là một bước quan trọng trong việc xây dựng hệ thống nhận dạng hành động người.

Ngoài ra, nếu sử dụng kết hợp có trọng số các phương pháp trích chọn đặc tính khác nhau thì độ chính xác của mô hình nhận dạng sẽ được cải thiện (90.7% so với phương pháp tốt nhất là 90.1%). Nếu sử dụng nhiều phương pháp hơn thì tỉ lệ nhận dạng sẽ cao hơn vì mô hình kết hợp trọng số khai thác tốt các thế mạnh của từng phương pháp riêng lẻ. Tuy nhiên việc sử dụng tất cả các phương pháp sẽ làm giảm hiệu năng của mô hình nhận dạng, thời gian nhận dạng sẽ lâu hơn. Bảng 3.26 so sánh thời gian thực hiện của mỗi phương pháp.

Bảng 3.26 So sánh thời gian giữa các phương pháp

Phương pháp	**Thủ công**	**PCA**	**LDA**	**Tổng hợp**
Thời gian (giây)	1	9	8	12

KẾT LUẬN

Nhận dạng hành động người được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau của cuộc sống. Kết hợp hai phương pháp nghiên cứu lý thuyết và thực nghiệm, luận văn này đã trình bày tổng quan về nhận dạng hành động người trong không gian 3D. Bao gồm các phương pháp thu thập dữ liệu chuyển động 3D, các giải thuật học máy thường sử dụng và đặc biệt là các phương pháp trích chọn đặc tính. Bên cạnh đó, luận văn đã nghiên cứu và đề xuất mô hình nhận dạng kết hợp có trọng số các phương pháp trích chọn đặc tính khác nhau với độ chính xác cao. Kết quả thực nghiệm cho thấy mô hình đề xuất cho kết quả nhận dạng tốt hơn so với mô hình truyền thống.

Ngoài ra quá trình thực nghiệm đã tìm ra được một nhóm các xương có ảnh hưởng lớn đến việc thể hiện các đặc trưng của hành động người trong không gian 3D.

Tuy nhiên, việc kết hợp sử dụng song song các phương pháp đã làm giảm hiệu năng của mô hình nhận dạng. Do vậy hướng nghiên cứu tiếp theo đối với đề tài này là nâng cao hiệu năng của hệ thống khi sử dụng kết hợp nhiều phương pháp khác nhau.

TÀI LIỆU THAM KHẢO

[1] J.K Aggarwal, Lu Xia (2014), "Human Activity Recognition from 3D Data-A Review", *Pattern Recognition Letters, Elsevier B.V*, USA.

[2] Aggarwal J.K, Ryoo M.S (2011), "Human Activity Analysis: A Review", *ACM Comput. Surv*, page 16.

[3] Kohei Arai, Rosa Andrie Asmara (2013), "3D Skeleton model derived from Kinect Depth Sensor Camera andits application to walking style quality evaluations", *IJARAL – International Journal of Advanced in Artificial Intelligence*.

[4] Turaga P, Chellappa R, Subrahmanian V.S, Udrea O (2008), "Machine Recognition of Human Activities: A survey", *Circuits Syst. Video Technol. IEEE Trans 18*, pages 1473-1488.

[5] Fengjun Lv, Ramakant Nevatia (2006), "Recongnition and Segmentation of 3D Human Action Using HMM and Multi-class AdaBoost", *Lecture Notes in Computer Science Vlolumn 3954, 2006, pp 359-372*.

[6] Rizwan Chaudhry, Ferda Ofli, Gregorij Kurillo, Ruzena Bajcsy, René Vidal (2013), "Bio-inspired Dynamic 3D Discriminative Skeletal Features for Human Action Recognition", *CVPR-2013*.

[7] Raviteja Vemulapalli, Felipe Arrate, Rama Chellappa (2014), "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group", *CVPR-2014*.

[8] http://mocap.cs.cmu.edu/

[9] Vasileios Argyriou, Maria Petrou, Svetlana Barsky (2010), "Photometric Stereo with an Arbitrary Number of Illuminants", *Computer Vision and Image Understanding 114*, pages 887-900.

[10] D. Gehrig, T. Schultz (2008), “Selecting Relevant Features for Human Motion Recognition”, *ICPR 2008, IEEE*.

[11] Dana Kulic, Wataru Takano, Yoshihiko Nakamura (2008), “Incremental Learning, Clustering and Hierarchy Formation of Whole Body Motion Patterns using Adaptive Hidden Markov Chains”, *The International Journal of Robotics Research vol. 27 no. 7* 761-784.

[12] Gita Sukthankar, Katia Sycara (2005), “A Cost Minimization Approach to Human Behavior Recognition”.

[13] Ahmad Jalal, Shaharyar Kamal, Daijin Kim (2014), “A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments”, *Sensors-2014*.

[14] Lasitha Piyathilaka, Sarah Kodagoda (2013), “Human Activity Recognition for Domestic Robots”.

[15] Mi Zhang, Alexander A. Sawchuk (2012), “Motion Primitive-Based Human Activity Recognition Using a Bag-of-Features Approach”.

[16] Md. Zia Uddin, Nguyen Duc Thang, Jeong Tai Kim, Tae-Seong Kim (2011), “Human Activity Recognition Using Body Joint-Angle Features and Hidden Markov Model”, *ETRI Journal, Volume 33*.

[17] V. Argyriou, M. Petrou, S. Barsky (2010), “Photometric stereo with an arbitrary number of illuminants”, *CVIU 114 887–900*.

[18] K. Adistambha, C. H. Ritz , I. S. Burnett (2008), “Motion Classification Using Dynamic Time Warping”, *ICPR 2008, IEEE*.