

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



**HỆ THỐNG HỖ TRỢ TẬP
VẬT LÝ TRỊ LIỆU THỜI GIAN THỰC
SỬ DỤNG KINECT**

LUẬN VĂN CỬ NHÂN

PHƯƠNG PHÁP TOÁN TRONG TIN HỌC

Thành phố Hồ Chí Minh - 2017

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

LUẬN VĂN CỬ NHÂN

**HỆ THỐNG HỖ TRỢ TẬP
VẬT LÝ TRỊ LIỆU THỜI GIAN THỰC
SỬ DỤNG KINECT**

SINH VIÊN: VŨ THÀNH HUY

MÃ SỐ SINH VIÊN: 1211062

KHOA: TOÁN - TIN HỌC

CHUYÊN NGÀNH: PHƯƠNG PHÁP TOÁN TRONG TIN HỌC

GIẢNG VIÊN HƯỚNG DẪN: TS. PHẠM THẾ BẢO

Thành phố Hồ Chí Minh - 2017

LỜI CÁM ƠN

Tôi xin chân thành cảm ơn thầy Phạm Thé Bảo vì sự hướng dẫn, chỉ bảo hết sức tận tình trong suốt quá trình thực hiện luận văn này. Những sự giúp đỡ của thầy thật sự rất quý báu.

Tôi cũng xin gửi lời cảm ơn đến quý thầy cô trong khoa Toán-Tin học cũng như trường đại học Khoa Học Tự Nhiên, đã tận tình giảng dạy, trang bị cho tôi những kiến thức quý báu trong suốt những năm học vừa qua.

Cuối cùng, xin được gửi lời cảm ơn của mình tới ba mẹ, gia đình, các bạn bè trong suốt thời gian tôi ngồi trên giảng đường đã hết lòng giúp đỡ, động viên.

Tuy rằng tôi đã cố gắng hết sức để hoàn thành luận văn này tốt nhất có thể, nhưng chắc chắn sẽ không tránh khỏi các thiếu sót. Rất mong nhận được sự chỉ bảo, góp ý từ thầy cô và các bạn.

TP. Hồ Chí Minh, 02/2017
Sinh viên thực hiện

Vũ Thành Huy

Email: vuthanhhuynh94@gmail.com
SĐT: 0909 906 735

NỘI DUNG

LỜI CẢM ƠN	2
DANH SÁCH CÁC BẢNG	5
DANH SÁCH CÁC HÌNH	6
DANH SÁCH CÁC THUẬT TOÁN	7
DANH SÁCH CÁC LUU ĐỒ	8
Chương 1 Giới thiệu bài toán	9
1.1 Bài toán xây dựng hệ thống hỗ trợ tập vật lý trị liệu	9
1.1.1 Vận động trị liệu	9
1.1.2 Các khó khăn trong vận động trị liệu	9
1.1.3 Đề xuất xây dựng hệ thống hỗ trợ tập vận động trị liệu	10
1.2 Hướng tiếp cận giải quyết bài toán	11
1.3 Đề xuất hướng giải quyết	13
1.4 Nội dung luận văn	13
Chương 2 Kiến thức cơ sở	15
2.1 Kiến thức cơ bản về vận động học trong vật lý trị liệu	15
2.2 Chức năng của Kinect	18
2.3 Biểu diễn một bài tập vật lý trị liệu	20
2.3.1 Dáng đứng và chuyển động	20
2.3.2 Trích đặc trưng khung xương	20
2.4 Phương pháp Dynamic Time Warping	23
2.4.1 Thuật toán canh chỉnh tín hiệu	23
2.4.2 Chuẩn hóa tín hiệu sau khi canh chỉnh	27
2.5 Bộ lọc trung bình	29
Chương 3 Xây dựng thuật giải quyết bài toán	33
3.1 Thuật toán thu thập thông tin bài tập mẫu	33
3.2 Thuật toán xử lý thông tin bài tập mẫu	38

3.2.1 Xử lý thông tin đáng đúng	38
3.2.2 Xử lý thông tin chuyển động	38
3.3 Thuật toán thu thập và đánh giá bài tập do bệnh nhân thực hiện	42
3.3.1 Thuật toán thu thập tín hiệu bài tập	42
3.3.2 Thuật toán đánh giá bài tập	43
3.4 Các tính năng đặc biệt	48
3.4.1 Thuật toán hướng dẫn đáng đúng tiếp theo	48
3.4.2 Thuật toán hướng dẫn vị trí động tác tiếp theo	48
Chương 4 Kết quả thực nghiệm và hướng phát triển	52
4.1 Cách thức đánh giá kết quả thực nghiệm	52
4.2 Kết quả thực nghiệm và nhận xét	54
4.3 Hướng phát triển	54

DANH SÁCH CÁC BẢNG

Bảng 1. Bảng tính độ lỗi giữa các lần thực hiện.....	39
Bảng 2. Bảng tính độ lỗi giữa các lần thực hiện so với lần thực hiện chuẩn.	39
Bảng 3. Kết quả thực nghiệm của hệ thống.	54

DANH SÁCH CÁC HÌNH

Hình 1. Hệ trục tọa độ cơ thể người	16
Hình 2. Minh họa các bài tập mẫu	17
Hình 3. Tầm hoạt động của Kinect.....	19
Hình 4. Các dáng đứng và chuyển động của động tác dang khép đứng	20
Hình 5. Các hệ trục tọa độ	21
Hình 6. Thứ tự các khớp được quy định	22
Hình 7. Tín hiệu chuyển động 1 của bài tập xoay khuỷu tay	23
Hình 8. Minh họa thuật toán Dynamic Time Warping.	24
Hình 9. Sử dụng Dynamic Time Warping để canh chỉnh hai tín hiệu.....	26
Hình 10. Minh họa cách thức chuẩn hóa Dynamic Time Warping	28
Hình 11. Minh họa thuật toán chuẩn hóa Dynamic Time Warping.	30
Hình 12. Minh họa lọc giảm nhiễu tín hiệu bằng bộ lọc trung bình.....	32
Hình 13. Minh họa thuật toán thu nhận bài tập mẫu	37
Hình 14. Cách thức phân loại chất lượng bài tập	44
Hình 15. Minh họa chức năng hướng dẫn dáng đứng tiếp theo.....	48
Hình 16. Minh họa thuật toán Dynamic Time Warping cục bộ.	49
Hình 17. Minh họa chức năng hướng dẫn bệnh nhân vị trí chi tiếp theo	51

DANH SÁCH CÁC THUẬT TOÁN

Thuật toán 1. Thuật toán Dynamic Time Warping.....	25
Thuật toán 2. Thuật toán chuẩn hóa Dynamic Time Warping	29
Thuật toán 3. Thuật toán giảm nhiễu bằng bộ lọc trung bình.....	31
Thuật toán 4. Thu thập bài tập mẫu	36
Thuật toán 5. Thuật toán xử lý dữ liệu đáng đúng và chuyển động	40
Thuật toán 6. Thuật toán đánh giá bài tập mới.....	45
Thuật toán 7. Thuật toán Dynacmic Time Warping cục bộ.....	50

DANH SÁCH CÁC LUU ĐỒ

Lưu đồ 1. Các bước thu thập một bài tín hiệu mẫu.....	34
Lưu đồ 2. Biểu diễn sơ đồ khói thuật toán thu thập tín hiệu mẫu	35
Lưu đồ 3. Các bước hệ thống thực hiện khi đánh giá bài tập mới	43
Lưu đồ 4. Biểu diễn sơ đồ khói thuật toán đánh giá bài tập mới	47

Chương 1

Giới thiệu bài toán

1.1 Bài toán xây dựng hệ thống hỗ trợ tập vật lý trị liệu

1.1.1 Vận động trị liệu

Theo định nghĩa của Liên Đoàn Vật Lý Trị Liệu Thế Giới (WCPT), “Vật lý trị liệu là một chuyên ngành trong Y Học Phục Hồi cung cấp cho mọi người những phương pháp điều trị nhằm duy trì, phát triển và hồi phục chức năng tối đa trong những trường hợp chân thương, bệnh tật, suy giảm về vận động và các chức năng trong quá trình phát triển của con người. Các phương pháp điều trị trong Vật lý trị liệu chú trọng vào sự hồi phục, cải thiện, phòng ngừa, điều trị các chức năng vận động càng nhiều càng tốt.” [1]. Vật lý trị liệu sử dụng các tác nhân vật lí tự nhiên hay nhân tạo như: nước, không khí, nhiệt độ, khí hậu, độ cao, điện, tia X, tia cực tím, tia hồng ngoại, siêu âm, các chất đồng vị phóng xạ, xoa bóp, thể dục - thể thao, đi bộ, dưỡng sinh...

Trong số đó, vận động trị liệu là một phương pháp vật lý trị liệu dùng sự vận động để điều trị nhằm phục hồi chức năng cho người bệnh, khi họ không thể thực hiện chức năng một cách độc lập do bệnh lý hay do thương tật gây ra [2]. Vận động trị liệu như một bài tập thể dục, phối hợp giữa Đông Y và Tây Y mà không cần phải dùng thuốc. Với những bệnh nhân có những căn bệnh như tai biến mạch máu não, đau lưng, tay, chân, thay các khớp xương, mất thăng bằng, đi lại không bình thường,... vận động trị liệu có thể giúp cho bệnh nhân đi đứng trở lại bình thường, khỏe mạnh hơn [2]. Vận động trị liệu không những giúp chữa bệnh, mà còn có thể luyện tập như một cách phòng bệnh. Thực vậy, những bài tập vận động trị liệu không chỉ giúp hồi phục sức khỏe, mà còn dạy bệnh nhân cách huấn luyện cơ thể và phòng ngừa tổn thương.

1.1.2 Các khó khăn trong vận động trị liệu

Tuy vậy, vẫn có những hạn chế bên cạnh những lợi ích của phương pháp vận động trị liệu:

- Khó khăn đầu tiên đó là thời gian tập rất dài. Mặc dù không cần phải dùng thuốc, nhưng bù lại, bệnh nhân phải tốn rất nhiều thời gian tập cho đến khi hoàn toàn bình phục. Điều này là một thách thức lớn với sự kiên nhẫn của bệnh nhân.

- Thứ hai, vì yêu cầu tiên quyết khi tập vận động trị liệu là bệnh nhân luôn phải thực hiện đúng bài tập, tập sai có thể gây đau đớn, kéo dài có nguy cơ chuyển thành tật. Do đó, bất cứ khi nào luyện tập, bệnh nhân cũng cần một hộ tá ở bên cạnh để kiểm tra rằng mình đang thực hiện các bài tập đúng cách. Vì vậy, để đáp ứng nhu cầu của một số lượng lớn các bệnh nhân tập vận động trị liệu, cũng cần một số lượng lớn các hộ tá có tay nghề.
- Thứ ba, việc bệnh nhân di chuyển từ nhà đến bệnh viện (hoặc hộ tá di chuyển từ bệnh viện tới nhà bệnh nhân) mỗi khi bệnh nhân cần tập vận động trị liệu dẫn tới nhiều khó khăn. Đầu tiên là về vấn đề chi phí, xét về kinh tế lẩn thời gian. Bên cạnh đó là vấn đề về các tác hại của khi bắt bệnh nhân di chuyển nhiều. Ví dụ như các bệnh nhận mới thay tim, cơ thể họ rất yếu, đi lại không vững, nếu di chuyển quá nhiều sẽ gây ảnh hưởng sức khỏe nghiêm trọng.

1.1.3 Đề xuất xây dựng hệ thống hỗ trợ tập vận động trị liệu

Để giải quyết những khó khăn trên, một phương án áp dụng công nghệ thông tin được đưa ra, đó là xây dựng một hệ thống phần mềm và phần cứng có thể kiểm tra, đánh giá động tác của người tập vận động trị liệu một cách chính xác mà không cần tới sự hiện diện của hộ tá.

Phương pháp này được mong chờ sẽ giải quyết được cả ba khó khăn nêu trên của vận động trị liệu. Thứ nhất, với hệ thống này, bệnh nhân có thể tự tập tại nhà mình mà không cần tới bệnh viện, và tập bất cứ lúc nào mình muốn. Việc này giúp bệnh nhân luôn cảm thấy thoải mái về mặt không gian lẩn thời gian trong khi tập, giúp họ có thể tập hiệu quả trong một thời gian dài, tránh việc mất kiên nhẫn. Thứ hai, hệ thống này sẽ giúp giảm áp lực lên số lượng hộ tá yêu cầu, vì khi này thời gian mỗi hộ tá dành cho một bệnh nhân sẽ ít đi, và do đó giúp đỡ điều trị được cho nhiều bệnh nhân hơn. Và thứ ba, hệ thống này giúp tiết kiệm rất nhiều chi phí về thời gian, tiền bạc đi lại của bệnh nhân và hộ tá.

Mục đích của luận văn này là đề xuất một cách thức xây dựng một hệ thống như trên đáp ứng các yêu cầu về độ chính xác, dễ dàng sử dụng, và giúp đỡ người tập hiệu quả nhất.

Bài toán xây dựng hệ thống tập vận động trị liệu có thể được chia thành ba bài toán nhỏ, tương ứng với ba giai đoạn làm việc cơ bản của hệ thống:

- Bài toán thu thập thông tin bài tập mẫu

Để thu dữ liệu cho một bài tập mẫu, đầu tiên ta cần bác sĩ (hoặc người tập mẫu) phải thực hiện đúng ít nhất 5 lần bài tập đó trước Kinect. Ta sẽ thu thập dữ liệu những lần thực hiện mẫu này để làm cơ sở chuẩn, dùng đánh giá chất lượng những lần bệnh nhân tập sau. Lưu ý rằng chỉ ở bước này ta mới cần sự hiện diện của bác sĩ, còn ở những bước sau, toàn bộ quá trình thu hình bài tập, đánh giá bài tập, ... đều được thực hiện tự động bằng phần mềm mà không cần bác sĩ.

Bởi vì trong quá trình tập luyện để bình phục, bệnh nhân sẽ phải tập nhiều bài tập khác nhau, với các biên độ động tác thay đổi liên tục theo thời gian, nên bác sĩ/người làm mẫu sẽ phải sử dụng chức năng thu thập động tác mẫu mới rất nhiều lần. Do đó, ta cần xây dựng chức năng này sao cho thật dễ sử dụng, nhanh chóng, tiết kiệm thời gian. Đây cũng chính là yêu cầu cao nhất ta cần phải đáp ứng.

- Bài toán xử lý thông tin bài tập mẫu

Sau khi thu thập dữ liệu thô của bài tập mẫu, ta sẽ tiến hành xử lý dữ liệu này và lưu vào cơ sở dữ liệu, và dùng chúng để so sánh, đánh giá tính chính xác của những lần thực hiện sau. Ở bài toán này, thời gian thực hiện không cần phải thật nhanh, nhưng yêu cầu đặt ra là chọn một cách biểu diễn dữ liệu thích hợp sao cho đặc trưng những lần thực hiện đúng luôn luôn giống nhau và giống với những lần thực hiện mẫu, còn các lần thực hiện sai thì luôn khác với bài tập mẫu.

- Bài toán thu và đánh giá bài tập do bệnh nhân thực hiện

Khi bệnh nhân thực hiện lại bài tập này (khi ở nhà, không có hộ tá), hệ thống của ta cần phải thu động tác này, xử lý, và so sánh với động tác mẫu trong cơ sở dữ liệu để đánh giá xem động tác vừa thực hiện có đúng không.

Với hệ thống được đề xuất trong luận văn, nếu động tác đúng, hệ thống sẽ trả về chất lượng lần thực hiện đúng đó (rất tốt, khá tốt, tạm, ...). Ngược lại, nếu động tác sai và hệ thống phát hiện được nguyên nhân sai, hệ thống sẽ trả về cách sửa động tác sai này.

1.2 Hướng tiếp cận giải quyết bài toán

Có hai phương án tiếp cận chính để xây dựng hệ thống kiểm tra tính chính xác một bài tập vận động trị liệu [3]. Cách thứ nhất biểu diễn các bài tập chuẩn dưới dạng các tập “luật”, là tập các điều kiện cần thỏa mãn trong khi bệnh nhân thực hiện động tác. Ta tạm gọi đây là cách tiếp cận sử dụng “bộ luật”. Các tập điều kiện này được bác sĩ đặt ra một cách thủ công, hoàn toàn không dựa trên tín hiệu thu bài tập mẫu. Ví dụ, yêu

cầu trong bài tập vai, “bộ luật” yêu cầu cánh tay phải ngang 90° so với trục thẳng đứng, cánh tay phải luôn nằm trong mặt phẳng Oxy,...

Cách thứ hai biểu diễn các bài tập chuẩn dựa trên tín hiệu thu được từ bài tập đúng thật sự do bác sĩ hoặc chính bệnh nhân thực hiện. Ta tạm gọi đây là cách tiếp cận theo kiểu “bài mẫu”. Tín hiệu của bài tập mẫu đúng sẽ được ghi nhận, xử lý và lưu vào cơ sở dữ liệu. Khi bệnh nhân tự tập lại ở nhà, tín hiệu từ bệnh nhân được thu lại và so sánh với bài tập mẫu đã được lưu trữ trước đó để kiểm tra độ chính xác của động tác. Hai phương pháp tiếp cận “bộ luật” và “bài tập mẫu” có những ưu và nhược điểm khác nhau [3].

Những lợi thế của phương pháp sử dụng “bộ luật”:

- Về thời gian tính toán, phương pháp “bộ luật” yêu cầu ít tính toán hơn, vì không cần phải sử dụng các thuật toán xử lý tín hiệu, canh chỉnh tín hiệu phức tạp như phương pháp “bài tập mẫu”.
- Phương pháp “bài tập mẫu” yêu cầu phải chuẩn hóa tín hiệu, để những người tập khác nhau (với vóc dáng cơ thể khác nhau) khi tập cùng một động tác đều tạo ra cùng một tín hiệu. Ngược lại, phương pháp bộ luật chỉ tập trung các điều kiện vào các biến không phụ thuộc vào vóc dáng người.
- Phương pháp “bộ luật” cho phép trả về các hướng dẫn chỉnh sửa cụ thể, chính xác hơn khi bệnh nhân tập sai, vì các điều kiện của bài tập đúng đã được ghi rõ ràng trong “bộ luật”. Việc này rất ý nghĩa vì các hướng dẫn chỉnh sửa cụ thể sẽ có ích hơn nhiều việc chỉ thông báo bệnh nhân tập đúng hay sai.

Ngược lại, phương pháp sử dụng “bài tập mẫu” cũng có các ưu điểm (và cũng là nhược điểm của phương pháp “bộ luật”):

- Phương pháp “bài tập mẫu” cho phép ta xây dựng bài tập chuẩn theo một cách rất dễ dàng, tự nhiên. Cụ thể, ta chỉ cần thực hiện đúng động tác đó trước Kinect để thu lại, xử lý và lưu vào bộ nhớ. Ngược lại, việc định nghĩa các bộ luật cho một động tác là một cách không tự nhiên, và yêu cầu bác sĩ có trình độ cao, hiểu rõ về các góc độ của các chi, cơ thể của bài tập đúng.
- Với các bài tập phức tạp, việc soạn các “bộ luật” liên quan tới nhiều biến số của cơ thể tồn tại rất nhiều thời gian, công sức. Trong khi đó, trong phương pháp “bài tập mẫu”, tất cả các biến số được ghi lại tự động khi thực hiện bài tập mẫu với đầy đủ các tính chất, tương quan giữa các biến số, giúp miêu tả bài tập chính xác hơn.
- Trong quá trình tập, bài tập của bệnh nhân thường được thay đổi liên tục theo quá trình phục hồi sức khỏe. Việc xây dựng các “bộ luật” “cứng” làm cho việc thay đổi, mở rộng bài tập trở nên khó khăn. Ngược lại, để thay đổi bài tập, với phương pháp

“bài tập mẫu”, ta chỉ cần bệnh nhân hoặc bác sĩ tập đúng bài tập mới để ghi nhận lại.

1.3 Đề xuất hướng giải quyết

Luận văn này đề xuất một hệ thống hỗ trợ tập vật lý trị liệu sử dụng phương pháp “bài tập mẫu” nhằm khai thác các lợi thế sẵn có của nó. Đồng thời, đề xuất các phương pháp cải tiến để giảm tối đa các bất lợi của phương pháp tiếp cận này. Cụ thể, các cải tiến trong luận văn giải quyết các vấn đề sau:

- Đề ra một phương pháp so sánh tín hiệu mẫu và tín hiệu mới chính xác, hiệu quả và tốn ít chi phí tính toán, có thể chạy thời gian thực.
- Cung cấp các hướng dẫn chỉnh sửa khi bệnh nhân tập sai. Tuy rằng các hướng dẫn không được cụ thể về mặt số liệu như phương pháp “bộ luật”, nhưng các hướng dẫn vẫn rất cụ thể, hữu ích giúp bệnh nhân sửa sai.
- Bên cạnh đó, luận văn còn đề ra một phương pháp thu tín hiệu mẫu rất nhanh gọn, dễ sử dụng, tự nhiên, tạo cảm giác thoải mái cho bệnh nhân.

Dựa trên những yêu cầu đặt ra bên trên, đề tài được chia thành các bài toán nhỏ sau nhằm đơn giản hóa vấn đề:

- Bài toán thu nhập thông tin bài tập mẫu
- Bài toán xử lý thông tin bài tập mẫu
- Bài toán thu thập và đánh giá bài tập do bệnh nhân thực hiện
- Các tính năng đặc biệt của hệ thống (bao gồm thuật toán hướng dẫn đáng đứng tiếp theo và hướng dẫn vị trí chi tiếp theo)

* Chú ý: Do giới hạn về kinh phí và thời gian, luận văn này chỉ giới hạn xử lý các bài tập quan tới các cơ khớp ở tay (khớp vai, khớp khuỷu tay). Và đối tượng bệnh nhân nhắm tới là thanh niên khỏe mạnh, bình thường, nhưng có nhu cầu tập vật lý trị liệu do gặp phải chấn thương, tai nạn.

1.4 Nội dung luận văn

Luận văn được chia làm 4 chương với nội dung chi tiết như sau:

- Chương 1 giới thiệu chung về bài toán xây dựng hệ thống hỗ trợ tập vật lý trị liệu và sự cần thiết, nhu cầu có thực trong thực tế của hệ thống. Tiếp theo đó giới thiệu về những kết quả nghiên cứu hiện tại của những cá nhân, nhóm trên thế giới về hệ thống này. Từ việc so sánh, đánh giá các phương pháp đã có, rút ra hướng tiếp cận và từ đó đề xuất hướng giải quyết vấn đề.

- Chương 2 trình bày các kiến thức cơ bản liên quan đến bài toán trình bày trong luận văn. Bao gồm các kiến thức về: vật lý trị liệu, Kinect, cách thức biểu diễn dữ liệu của một bài tập, phương pháp Dynamic Time Warping và bộ lọc trung bình.
- Chương 3 đi sâu vào mô tả thuật toán giải quyết các bài toán của hệ thống. Bao gồm: bài toán thu thập thông tin bài tập mẫu; bài toán xử lý thông tin bài tập mẫu; bài toán thu thập và đánh giá bài tập mới; bài toán xây dựng các tiện ích hỗ trợ bệnh nhân khi tập.
- Chương 4 trình bày về việc đánh giá chất lượng của hệ thống để xuất thông qua kiểm thử và hướng phát triển. Cụ thể, trình bày về cách thức tiến hành thử nghiệm, bộ thử nghiệm, kết quả thử nghiệm và cuối cùng là hướng phát triển của luận văn trong tương lai.

Chương 2

Kiến thức cơ sở

2.1 Kiến thức cơ bản về vận động học trong vật lý trị liệu

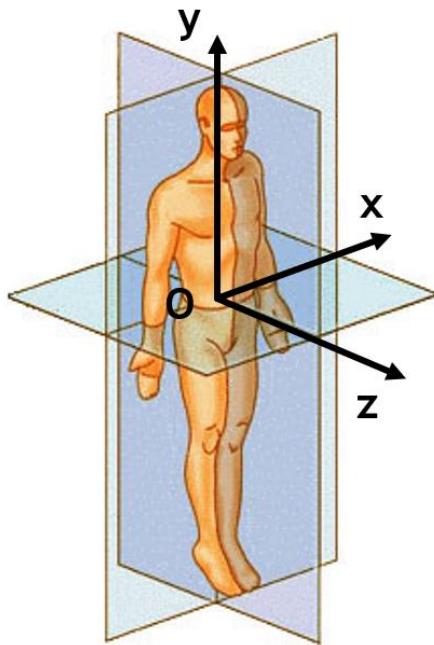
Bài tập vận động trị liệu được chia làm ba loại [2]:

- Bài tập thụ động: là bài tập ở các cơ khớp cần chữa trị mà động lực chính là ngoại lực (đến từ người hỗ trợ điều trị hoặc máy móc cơ học), bệnh nhân không chủ động tập.
- Bài tập chủ động: là các bài tập bệnh nhân tự làm chuyển động các cơ khớp của mình để hồi phục. Bài tập chủ động được chia thành:
 - Chủ động trợ giúp: khi người hướng dẫn hỗ trợ và bệnh nhân cũng vừa tự thực hiện động tác.
 - Chủ động tự do: khi bệnh nhân tự thực hiện động tác mà không cần bất cứ sự trợ giúp nào.
 - Chủ động có đòn kháng: khi bệnh nhân thực hiện động tác nhằm kháng lại một ngoại lực đòn kháng như lực cản của bác sĩ hoặc máy móc cơ học.
- Bài tập cường bách: là những bài tập kéo dãn chủ động hay thụ động.

Các bài tập được đề cập tới trong luận văn là các bài tập chủ động tự do. Cụ thể, ta sẽ xem xét các bài tập vận động trị liệu liên quan tới các khớp tay bao gồm khớp vai và khớp khuỷu tay.

Để mô tả các bài tập được đề cập, ta dựa trên các kiến thức cơ bản sau về hình thể người. Với một mẫu người đứng thẳng, ta xét ba mặt phẳng sau:

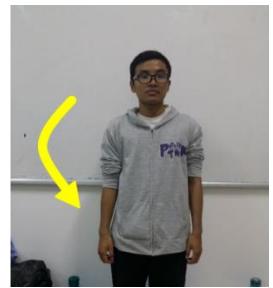
- Mặt phẳng đứng dọc: là mặt phẳng chia cơ thể làm hai nửa, nửa trái và nửa phải. Nếu xét hệ trục tọa độ như Hình 1, thì đây chính là mặt phẳng Oyz.
- Mặt phẳng đứng ngang: là mặt phẳng chia cơ thể làm hai nửa, trước bụng và sau lưng. Nếu xét hệ trục tọa độ như hình Hình 1, thì đây chính là mặt phẳng Oxy.
- Mặt phẳng nằm ngang: là mặt phẳng chia cơ thể làm hai nửa, thân trên và thân dưới. Nếu xét hệ trục tọa độ như Hình 1, thì đây chính là mặt phẳng Oxz.



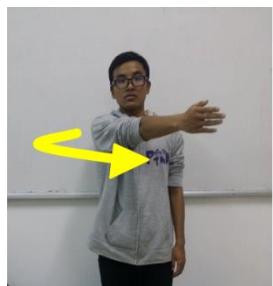
Hình 1. Hệ trục tọa độ cơ thể người

Trong giới hạn luận văn này, các động tác sẽ được xem xét và thí nghiệm là:

- Dang khép đứng: bệnh nhân bắt đầu ở tư thế chuẩn (là tư thế đứng thẳng, hai tay thả lỏng), dang tay lên và qua đầu trong mặt phẳng Oxy. Sau đó hạ tay xuống trở về tư thế chuẩn, vẫn giữ trong mặt phẳng Oxy. Hình 2a minh họa bài tập này.
- Dang khép nằm: bệnh nhân bắt đầu ở tư thế tay giơ ngang 90° trong mặt phẳng Oxy. Bệnh nhân tự di chuyển cánh tay trong mặt phẳng Oyz về phía ngược lại của cơ thể (di chuyển tay trái qua phía phải, hoặc di chuyển tay phải qua trái). Trong khi di chuyển, cánh tay luôn phải giữ thẳng và nằm trong mặt phẳng Oyz. Sau đó, bệnh nhân di chuyển cánh tay về vị trí ban đầu, và hoàn thành bài tập. Hình 2b minh họa bài tập này.
- Xoay khuỷu tay trong và ngoài: bệnh nhân bắt đầu ở tư thế chuẩn bị - cùi chỏ ép sát người, cẳng tay giơ ngang 90° trong mặt phẳng Oxy. Bệnh nhân tự di chuyển cẳng tay trong mặt phẳng Oxz về phía ngược lại của cơ thể (di chuyển tay trái qua phía phải, hoặc ngược lại). Trong khi di chuyển, cẳng tay luôn phải nằm trong mặt phẳng Oxz và giữ cùi chỏ ép sát người. Sau đó, bệnh nhân di chuyển cánh tay về vị trí ban đầu, và hoàn thành bài tập. Hình 2c minh họa bài tập này.
- Gấp duỗi: bệnh nhân bắt đầu ở tư thế chuẩn (là tư thế đứng thẳng, hai tay thả lỏng), đưa tay từ từ lên và qua đầu trong mặt phẳng Oyz. Sau đó hạ tay xuống trở về tư thế chuẩn, vẫn giữ trong mặt phẳng Oyz. Hình 2d minh họa bài tập này.



a)



b)



c)



d)

Hình 2. Minh họa các bài tập mẫu

- a) *Bài tập dang khép đứng;* b) *Bài tập dang khép nằm;*
- c) *Bài tập xoay khuỷu tay;* d) *Bài tập gấp duỗi.*

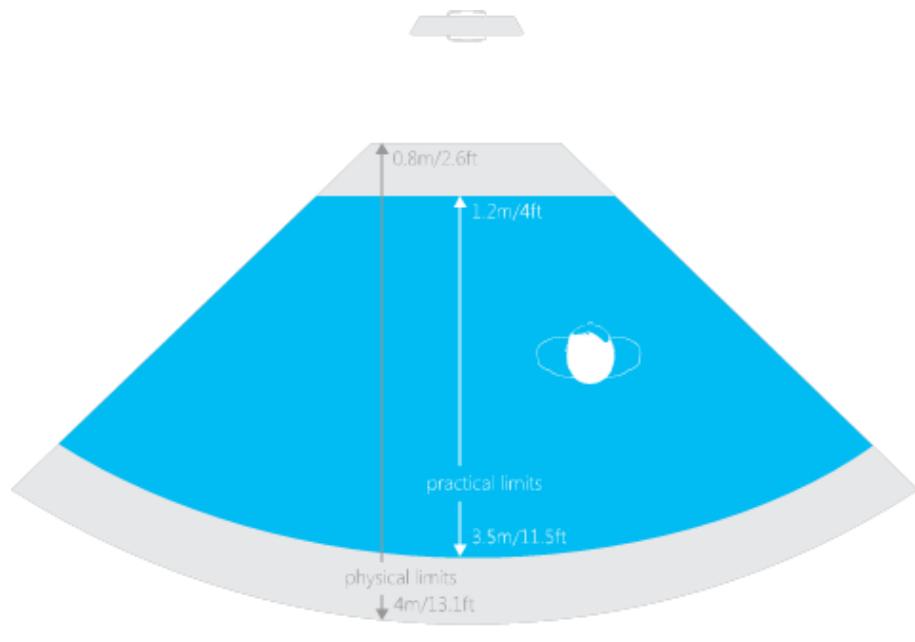
2.2 Chức năng của Kinect

Kinect là một thiết bị cảm biến chuyển động được Microsoft sản xuất, và thường được sử dụng kèm với máy chơi game Xbox 360 [4]. Tuy nhiên, hiện nay Microsoft cũng đã sản xuất dòng Kinect có thể được sử dụng và lập trình trên Windows, giúp các lập trình viên có thể tận dụng thiết bị này để viết ra những phần mềm hữu ích. Kinect cung cấp cho người dùng một giao diện giao tiếp với máy tính một cách tự nhiên, thông qua chuyển động cơ thể, mà không cần phải đeo, cầm bất kỳ phụ kiện nào.

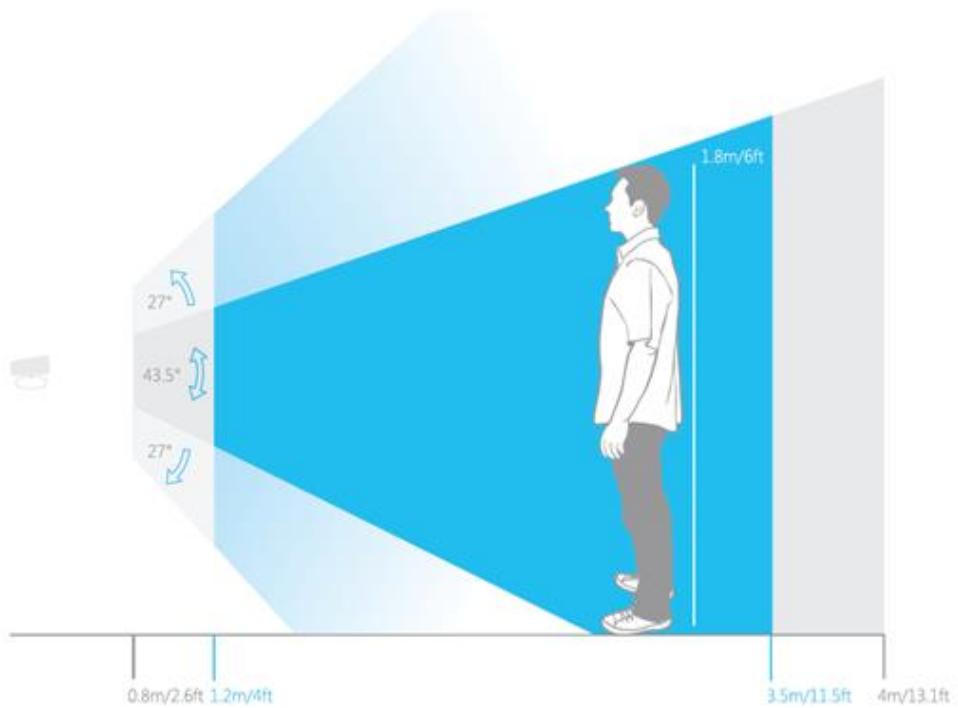
Khi sử dụng, Kinect có thể thu được các tín hiệu: ảnh màu sắc, ảnh độ sâu và âm thanh. Tín hiệu được thu nhận với tốc độ 30 frame mỗi giây. Dựa trên những thông tin này, Kinect còn có thể nhận dạng được con người trong video, và phân tích được vị trí khung xương của người này. Đây là một tính năng rất mạnh và quan trọng và được sử dụng trong rất nhiều các ứng dụng sử dụng Kinect. Trong hệ thống hỗ trợ tập vật lý trị liệu, chúng ta cũng sẽ tận dụng tính năng này của Kinect để trích tín hiệu khung xương từ người tập.

Theo nhà sản xuất, giới hạn làm việc của Kinect Xbox 360 được sử dụng trong thí nghiệm có thể thu nhận tín hiệu trong khoảng cách giới hạn 0.8 - 4m tính từ Kinect, trong đó, tín hiệu thu được tốt nhất nếu nằm trong khoảng từ 1.2 - 3.5m. Góc thu nhận của Kinect tính theo chiều ngang là 57° , và chiều dọc là 43.5° ($\pm 27^\circ$ do tính năng quay theo trực dọc của Kinect).

Trong thí nghiệm, ta sử dụng bộ phát triển phần mềm cho Kinect (Kinect SDK – software development kit) để xây dựng hệ thống hỗ trợ tập vật lý trị liệu.



a)



b)

Hình 3. Tầm hoạt động của Kinect

a) Tầm khoảng cách hoạt động của Kinect.

b) Tầm góc hoạt động của Kinect

2.3 Biểu diễn một bài tập vật lý trị liệu

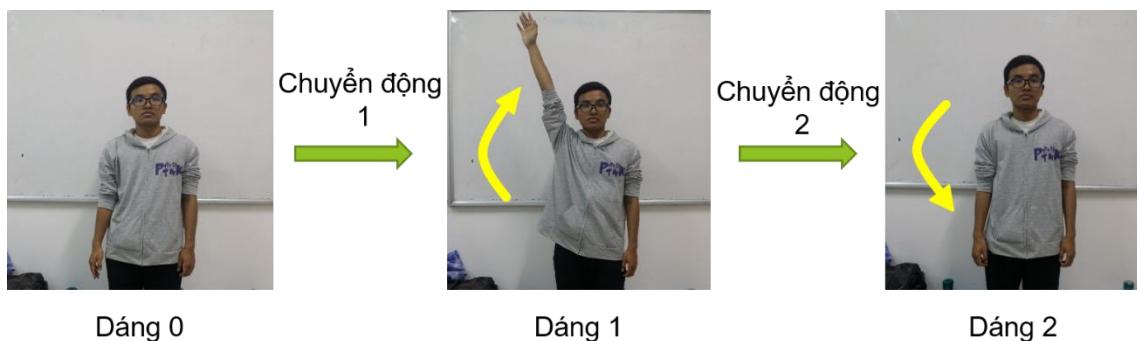
2.3.1 Dáng đứng và chuyển động

Trong hệ thống được xây dựng, một bài tập vật lý trị liệu được chia thành các dáng đứng (pose) và các chuyển động (movement) giữa các dáng đứng. Tổng quát, một bài tập sẽ có n dáng đứng và $(n-1)$ chuyển động. Trong đó, dáng đứng đầu tiên là tư thế chuẩn bị của bài tập, và dáng đứng cuối cùng là tư thế kết thúc bài tập. Người bệnh đi từ dáng đứng này sang dáng tiếp theo thông qua các chuyển động xen giữa các dáng đứng.

Ta đề xuất cách thức quy ước đánh số các dáng đứng và chuyển động như sau, nếu bài tập có (n) dáng đứng:

- Dáng đứng đầu tiên, cũng là tư thế chuẩn bị được đánh số 0 (*pose 0*), các dáng đứng tiếp theo được đánh số *pose 1, pose 2, pose 3, ... pose (n-1), pose n-1*.
- Chuyển động giúp bệnh nhân đi từ tư thế thứ k sang tư thế $(k+1)$ được đánh số là chuyển động thứ $(k+1)$. Như vậy, ta sẽ có các chuyển động là *movement 1, movement 2, movement 3... movement n-1*.

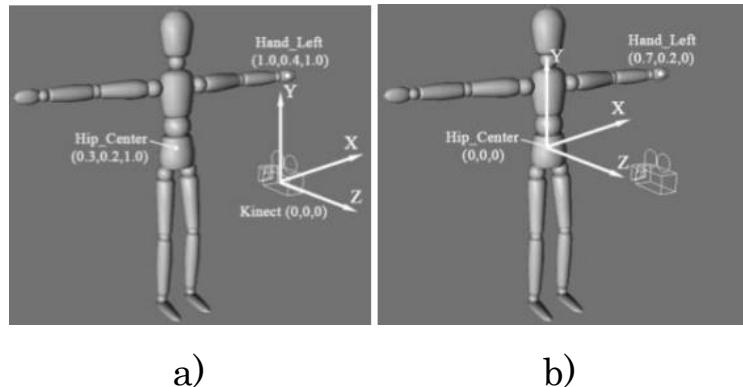
Hình 4 minh họa các dáng đứng và chuyển động của bài tập đang khép đứng.



Hình 4. Các dáng đứng và chuyển động của động tác dang khép đứng

2.3.2 Trích đặc trưng khung xương

Tín hiệu trả về từ Kinect là tọa độ ba chiều của các khớp cơ thể người trong không gian với gốc tọa là Kinect. Hình 5a minh họa hệ trực tọa độ này. Tuy nhiên, do ta mong muốn các tín hiệu này phải bất biến với vị trí của người tập trong không gian (bất biến với phép tịnh tiến, xoay của cơ thể người,...) nên ta sẽ chuyển các tọa độ ba chiều của các khớp về hệ tọa độ gắn trên cơ thể người như Hình 5b. Hệ tọa độ này có thể tính được từ tọa độ của các khớp phần thân trên cơ thể như đầu, cột sống, vai trái, phải, trung tâm, hông trái, phải, trung tâm. (Hình 5 sử dụng hình ảnh từ nguồn [5])



a)

b)

*Hình 5. Các hệ trục tọa độ**a) Hệ trục tọa độ của Kinect.**b) Hệ trục tọa độ của người dùng*

Tiếp đến, với mỗi cặp khớp chi liền kề nhau (vai và cùi chỏ, cùi chỏ và cổ tay,...) ta trừ hai tọa độ (sau khi đã đổi hệ tọa độ) của cặp khớp này để có được vector nối hai khớp đó. Trong thực tế, vector này chỉ phần cánh tay/cẳng tay, cánh chân/cẳng chân của người dùng. Ta sử dụng vector này để biểu diễn các tư thế của bệnh nhân.

Tuy nhiên, ta không sử dụng ngay vector vừa tính làm đặc trưng. Mà ta sẽ chuẩn hóa nó bằng cách chia cho độ dài của vector đó. Bằng cách này, đặc trưng thu được của ta sẽ không phụ thuộc vào dáng vóc (chiều cao, mập ốm,...) của người tập. Do đó, mặc dù khi thu bài tập mẫu, ta có thể thu của bác sĩ, và dùng nó để so sánh, đánh giá độ chính xác của bài tập thực hiện bởi bệnh nhân ở nhà. Hơn nữa, việc này còn giúp hệ thống của ta độc lập với khoảng cách giữa người tập tới Kinect. Vì khi người dùng tới gần Kinect, hình ảnh người lớn lên và độ dài các chi cũng thay đổi, và khi bước ra xa, độ dài các chi cũng giảm xuống. Việc chuẩn hóa vector tay/chân giúp đặc trưng khung xương của ta không phụ thuộc vào vị trí của người tập so với Kinect.

Trong hệ thống thử nghiệm giới thiệu trong luận văn này, do ta chỉ thử nghiệm với các bài tập tay, nên để đơn giản, ta chỉ thu tín hiệu từ các khớp tay. Tôi đề xuất đặc trưng trích từ khung xương sẽ có dạng sau đây. Có tổng cộng 12 biến tín hiệu, bao gồm: tọa độ x, y, z của vector cánh tay trên phải, tọa độ x, y, z của vector cẳng tay phải; tọa độ x, y, z của vector cánh tay trên trái, tọa độ x, y, z của vector cẳng tay trái. Như vậy, tại mỗi frame, thông tin ta trích xuất từ tín hiệu khung xương như công thức (1) do tôi đề xuất.

$$signal = \left(\frac{LeftArm1}{\|LeftArm1\|_2}, \frac{LeftArm2}{\|LeftArm2\|_2}, \frac{RightArm1}{\|RightArm1\|_2}, \frac{RightArm2}{\|RightArm2\|_2} \right) \quad (1)$$

với

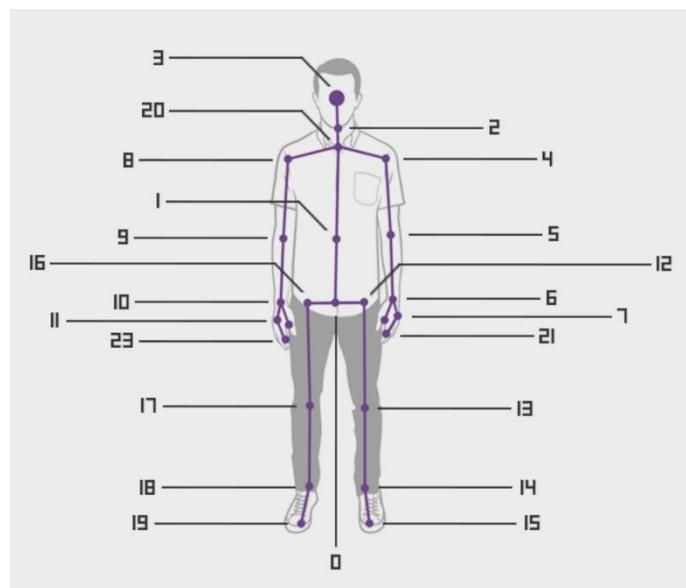
$$\begin{aligned} LeftArm1 &= skeleton[5] - skeleton[4]; \\ LeftArm2 &= skeleton[6] - skeleton[5]; \\ RightArm1 &= skeleton[9] - skeleton[8]; \\ RightArm2 &= skeleton[10] - skeleton[9]; \end{aligned}$$

$\|\cdot\|_2$ là chuẩn Euclid

skeleton là cấu trúc chứa tọa độ x, y, z của tất cả các khớp khung xương người (đã được chuyển về hệ tọa độ gắn trên bệnh nhân). Cụ thể, mỗi phần tử trong *skeleton* có dạng (2).

$$skeleton[i] = (x_i, y_i, z_i) \quad (2)$$

biểu diễn tọa độ (tính theo đơn vị mét) của khớp thứ *i*. Hình 6 mô tả dữ liệu của các phần tử trong cấu trúc *skeleton*.

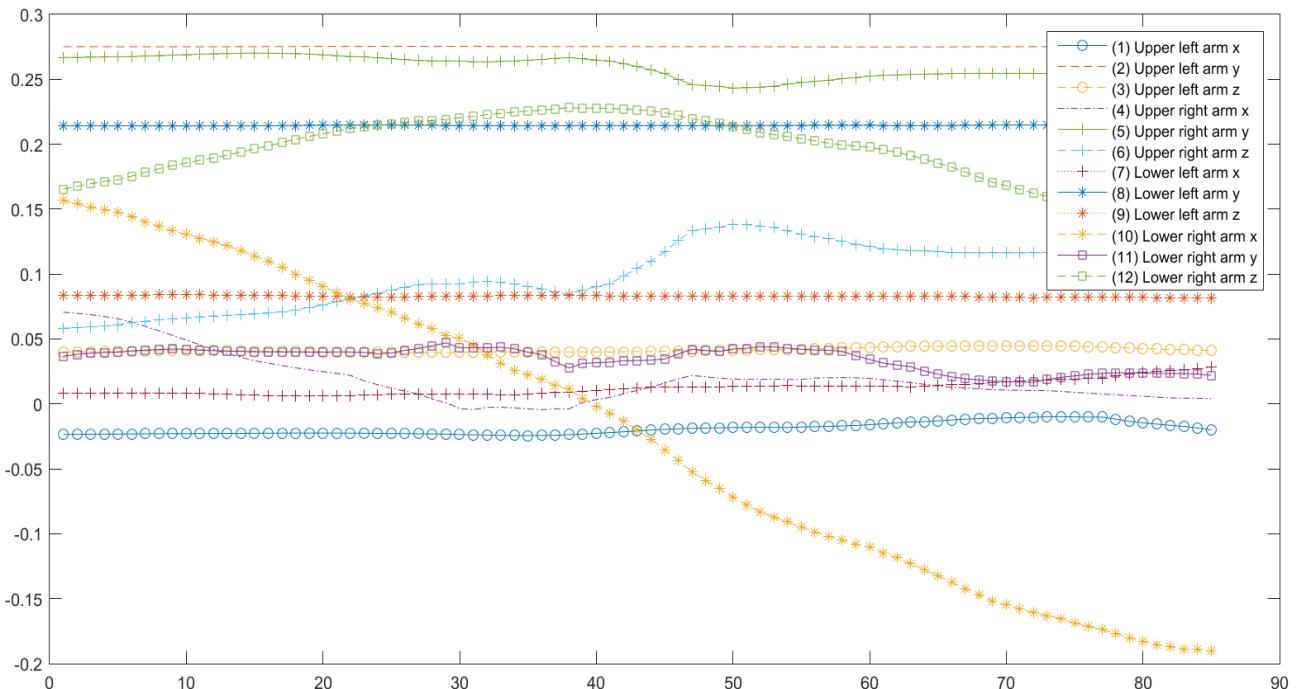


Hình 6. Thứ tự các khớp được quy định

Với quy ước về tín hiệu khung xương tại một frame như trên, ta tính khoảng cách giữa hai tín hiệu bằng công thức (3).

$$d(signal1, signal2) = \|signal2 - signal1\|_2 \quad (3)$$

Khi thu một chuỗi tín hiệu (chuỗi tín hiệu khung xương chuyển động), ta sử dụng một mảng hai chiều để lưu, với chiều thứ nhất là lưu thông tin tín hiệu từng frame, và chiều thứ hai là chiều thời gian. Ví dụ, Hình 7 minh họa tín hiệu thu được của chuyển động xoay khuỷu tay theo thời gian.



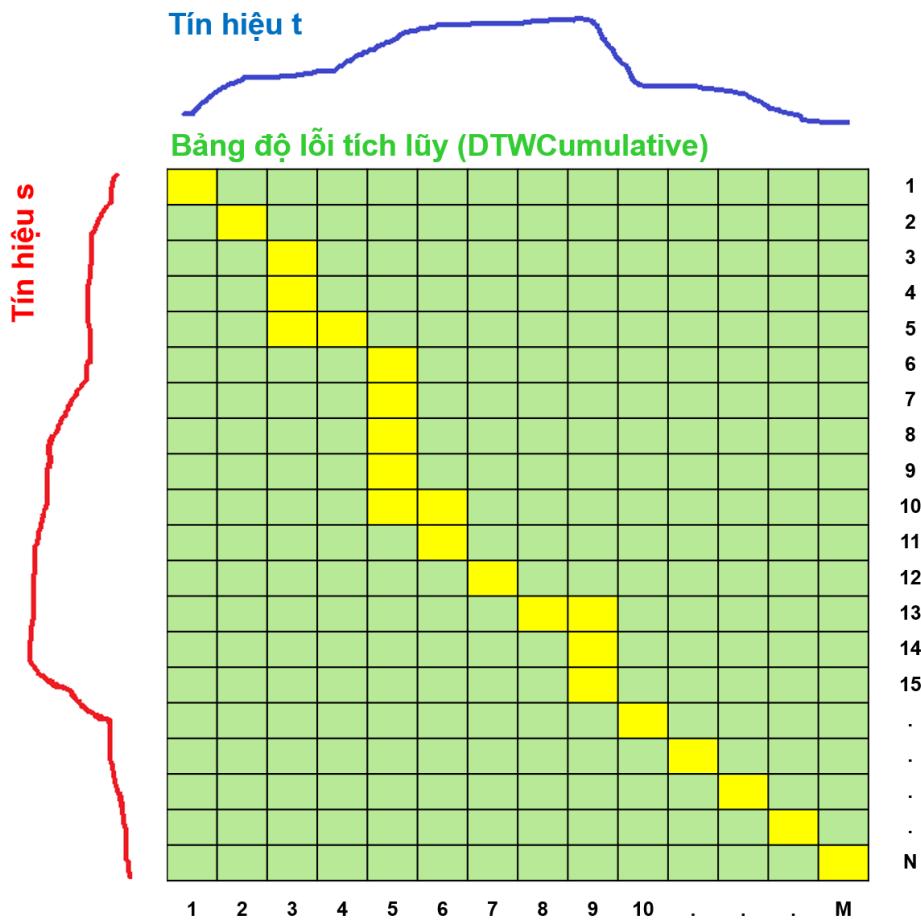
Hình 7. Tín hiệu chuyển động 1 của bài tập xoay khuỷu tay

2.4 Phương pháp Dynamic Time Warping

2.4.1 Thuật toán canh chỉnh tín hiệu

Dynamic Time Warping (DTW) là một thuật toán dùng để canh chỉnh, đo sự giống nhau giữa hai tín hiệu tương tự nhau về hình dạng nhưng có thể khác nhau về thời gian [6]. Ví dụ, có thể sử dụng DTW để đánh giá sự giống nhau giữa tín hiệu của hai dáng đi, mặc dù một tín hiệu có bước đi nhanh hơn tín hiệu bước đi của tín hiệu còn lại, hoặc trong lúc đi, người thực hiện có thể tăng tốc hoặc giảm tốc bất cứ lúc nào. DTW có thể được áp dụng trong xử lý các tín hiệu theo chuỗi thời gian như tín hiệu video, âm thanh,... Ví dụ, một trong những ứng dụng quan trọng của DTW là nhận diện giọng nói. DTW giúp ta vượt qua được khó khăn khi người nói nói ở các tốc độ khác nhau.

Để canh chỉnh và đo sự tương đồng giữa hai tín hiệu là s (có độ dài N) và t (có độ dài M). Thuật toán 1 trình bày thuật toán Dynamic Time Warping. *Hình 8* mô tả cách thức hoạt động của thuật toán.



Hình 8. Minh họa thuật toán Dynamic Time Warping.

Các ô màu vàng chỉ con đường di canh chỉnh ngắn nhất từ $[0,0]$ tới $[N,M]$

Trong bài toán của ta, ta sẽ dùng DTW để đo độ tương đồng giữa động tác mẫu và động tác mới thực hiện bởi bệnh nhân. Vì hai động tác này tuy về lý thuyết thì giống nhau, nhưng khi thực hiện, tốc độ động tác có thể khác nhau, lúc nhanh lúc chậm. Do đó, ta dùng DTW để đánh giá động tác độc lập với thời gian thực hiện. *Hình 9a* và *Hình 9b* minh họa tín hiệu thu được từ chuyển động thứ nhất của động tác đang khép nắp. *Hình 9a* là động tác mẫu của bác sĩ (khoảng 68 frame), *Hình 9b* là động tác vừa thực hiện của bệnh nhân (khoảng 96 frame). Để ý rằng tuy hình dạng của hai chuyển động giống nhau, nhưng chúng khác nhau về mặt thời gian. Sử dụng DTW, ta canh chỉnh lại hai chuyển động này. *Hình 9c* và *Hình 9d* minh họa hai chuyển động sau khi được canh chỉnh.

Thuật toán 1. Thuật toán Dynamic Time Warping

Input:

s, t : hai tín hiệu cần canh chỉnh
(s là một mảng $[1, \dots, N]$; t là một mảng $[1, \dots, M]$)

Output:

$TotalLoss$: độ lỗi sai khác nhau nhỏ nhất giữa hai tín hiệu
 $Track$: đường đi canh chỉnh gần nhất giữa hai tín hiệu

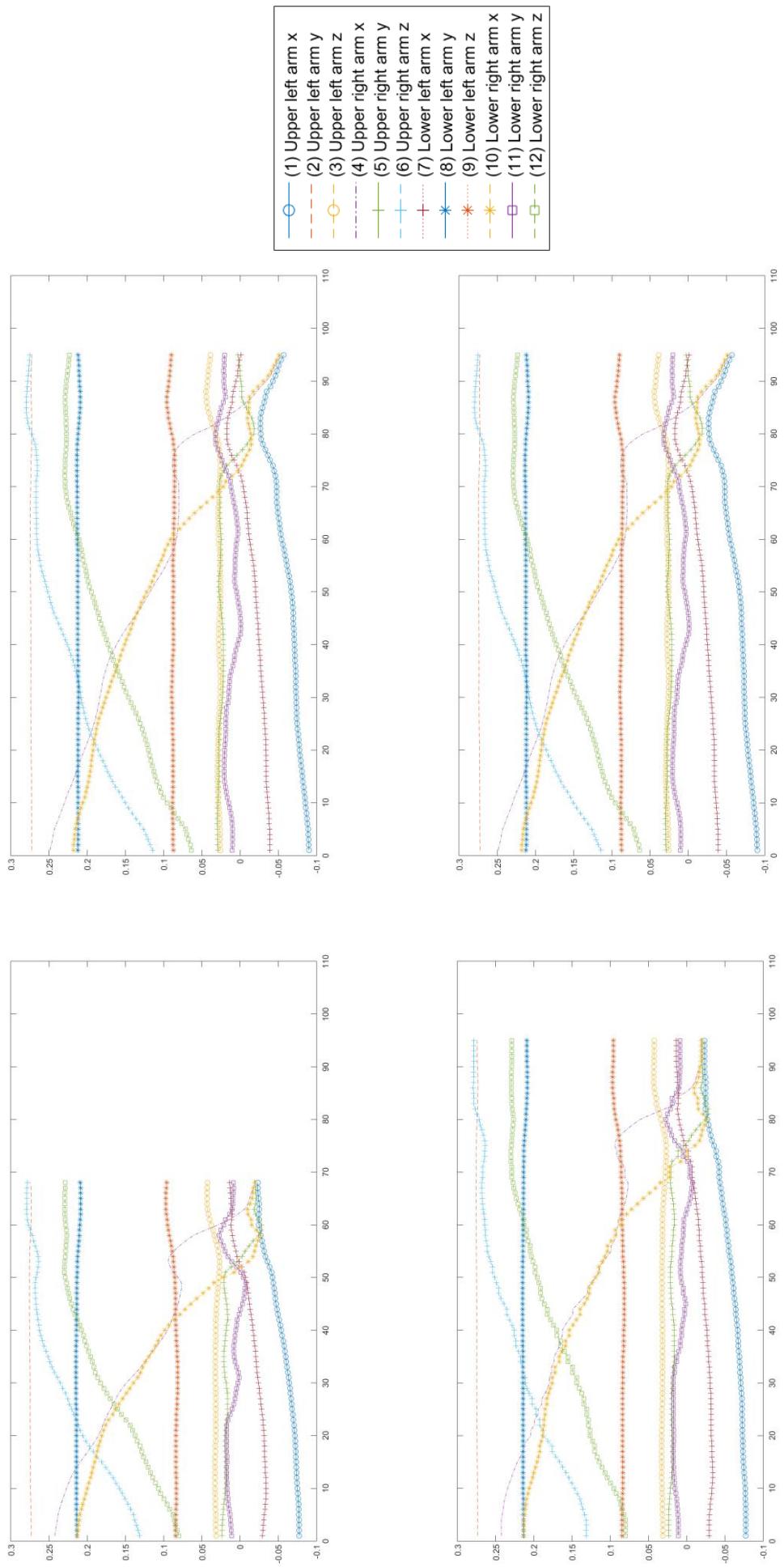
Thuật toán:

Begin

- Tạo mảng $DTWCumulative [0..N, 0..M]$ để tính độ lỗi tích lũy giữa hai tín hiệu.
- Gán $DTWCumulative [i, 0] = infinity$, với i từ $0 \rightarrow N$; gán $DTWCumulative [0, j] = infinity$, với j từ $0 \rightarrow M$.
- Tính toán toàn bộ mảng $DTWCumulative$ bằng công thức:
$$Loss = d(s[i], t[j]);$$
$$DTWCumulative [i, j] = Loss + minimum(DTWCumulative [i-1, j],$$
$$DTWCumulative [i, j-1], DTWCumulative [i-1, j-1])$$

với i từ $1 \rightarrow N$; j từ $1 \rightarrow M$
- Trả về giá trị $TotalLoss = DTWCumulative [N, M]$.
- Từ vị trí $[N, M]$, ta lần ngược về $[0, 0]$, đi qua những vị trí ta tính vào tổng $TotalLoss$.
Trả về đường đi canh chỉnh gần nhất $Track$.

End



Hình 9. Sử dụng Dynamic Time Warping để cạnh chỉnh hai tín hiệu.

a) Tín hiệu mẫu của bác sĩ. b) Tín hiệu thu mới của bệnh nhân.

c) Tín hiệu sau khi cạnh chỉnh của bác sĩ. d) Tín hiệu sau khi cạnh chỉnh của bệnh nhân.

2.4.2 Chuẩn hóa tín hiệu sau khi canh chỉnh

Một vấn đề gặp phải khi tính toán độ lỗi giữa hai tín hiệu sau khi canh chỉnh là sự không thống nhất về độ dài tín hiệu sau canh chỉnh. Cụ thể, với các chuyển động mà bệnh nhân thực hiện quá chậm, làm cho tín hiệu sau khi canh chỉnh bị kéo dài ra. Khi này, nếu ta tính tổng độ lỗi theo cách tính tổng khoảng cách Euclid tại tất cả các thời điểm, độ lỗi sẽ tăng lên rất cao. Bởi vì mặc dù hai chuyển động mới thực hiện và chuyển động mẫu rất giống nhau, độ lỗi tại mỗi thời điểm rất thấp, nhưng vì có rất nhiều thời điểm nên khi tính tổng lại độ lỗi vẫn rất cao. Ngược lại, một chuyển động tuy sai, nhưng nếu thực hiện nhanh, thì tín hiệu sau khi canh chỉnh có độ dài ngắn, thì tại mỗi thời điểm dù độ lỗi khá cao nhưng do có ít thời điểm, nên khi tổng lại độ lỗi vẫn không quá cao.

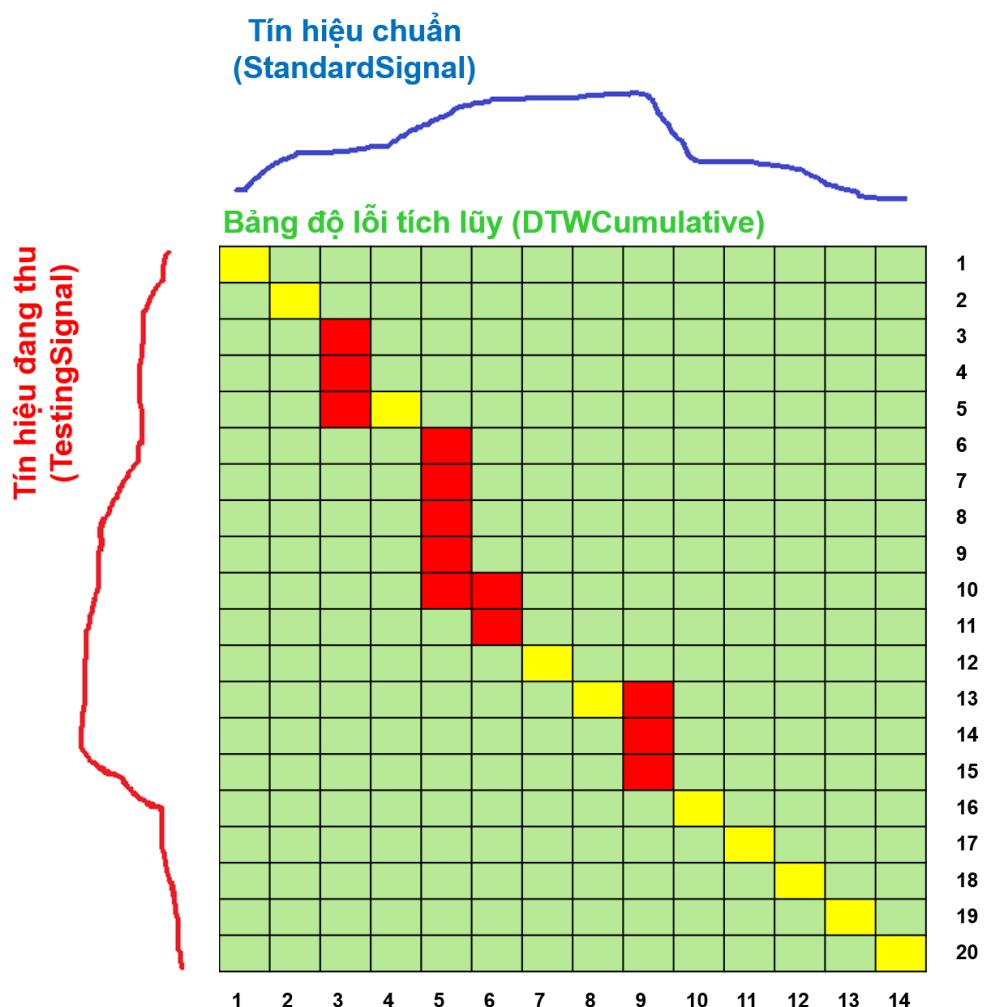
Để xử lý vấn đề này, tôi đề xuất một thuật toán để chuẩn hóa các tín hiệu sau khi đã được canh chỉnh. Nghĩa là sau khi canh chỉnh, độ dài của cả hai tín hiệu sẽ dài ra, nhưng ta sẽ đưa chúng về lại độ dài gốc của tín hiệu mẫu. Nhờ vậy, dù tín hiệu mới có quá nhanh hay quá chậm, thì sau khi canh chỉnh và đưa về độ dài của tín hiệu mẫu, ta đều có những tín hiệu có độ dài tương đương nhau. Và tới khi này ta mới tính độ lỗi của chuyển động đó với chuyển động mẫu. Điều này đảm bảo việc tính độ lỗi của ta hoàn toàn độc lập với thời gian thực hiện động tác.

Tuy vậy, ta không đơn thuần chỉ resize lại tín hiệu sau khi canh chỉnh DTW về kích cỡ của tín hiệu gốc. Vì việc này vẫn chưa đảm bảo chuẩn hóa các chuyển động mới tương ứng với chuyển động mẫu ban đầu. Ta sẽ làm như sau: quét qua toàn bộ frame trong tín hiệu mẫu, tại mỗi frame, nếu chỉ có một frame tương ứng của tín hiệu thu mới (đã được canh chỉnh), ta sẽ nhận frame này. Nhưng nếu có nhiều hơn một frame tương ứng, ta chỉ có thể chọn một frame duy nhất mà thôi. Và ta sẽ chọn frame mà khi so với frame đang xét của tín hiệu mẫu thì có độ lỗi thấp nhất. Và như vậy khi quét hết qua các frame của tín hiệu mẫu, ta cũng có được một tín hiệu tương ứng (đã được giản gọn) của tín hiệu thu mới sau khi canh chỉnh, có cùng độ dài với tín hiệu mẫu. Hình 10 minh họa quá trình này.

Thuật toán 2 mô tả thuật toán chuẩn hóa Dynamic Time Warping do tôi đề xuất. Hình 11 minh họa hai tín hiệu đê cập trong ví dụ phần 2.4.1 đã được chuẩn hóa sau bước canh chỉnh.

Trong Hình 10, ta sẽ quét qua toàn bộ các frame của *StandardSignal* (các cột). Tại mỗi cột, nếu số ô vàng là một, thì ta sẽ chọn frame tương ứng đó của *TestingSignal* (dòng).

Nếu cột có nhiều hơn một ô vàng (trong hình đánh dấu bằng các vùng đỏ), ta sẽ chọn ô mà độ lỗi với frame đang xét của *StandardSignal* nhỏ nhất. Sau các bước trên, ta sẽ có tín hiệu *TestingSignal* được chỉnh sửa, chuẩn hóa dựa trên chính tín hiệu chuẩn *StandardSignal* gốc.



Hình 10. Minh họa cách thức chuẩn hóa Dynamic Time Warping
 (các ô màu đỏ chỉ các vị trí tín hiệu mới thu dài hơn tín hiệu chuẩn, gây tăng
 độ lỗi không đáng có)

Thuật toán 2. Thuật toán chuẩn hóa Dynamic Time Warping

Input:

Track: đường đi canh chỉnh gần nhất giữa hai tín hiệu chuẩn (biến *StandardSignal*) và tín hiệu đang xét (biến *TestingSignal*)

Output:

NormalizedTrack: đường đi canh chỉnh gần nhất được chuẩn hóa

Thuật toán:

Begin

Quét qua từng frame tín hiệu của tín hiệu chuẩn *StandardSignal*

- Với từng frame của tín hiệu chuẩn, dựa vào *Track*, chọn ra những frame tương ứng gần nhất của tín hiệu đang xét *TestingSignal*:

+ Nếu *TestingSignal* chỉ có 1 frame tương ứng: chọn frame này và cho vào biến *TrackNormalized*

+ Nếu *TestingSignal* có nhiều hơn 1 frame tương ứng: chọn ra frame có độ lỗi thấp nhất so với frame đang xét của *StandardSignal*. Đưa frame này vào biến *TrackNormalized*

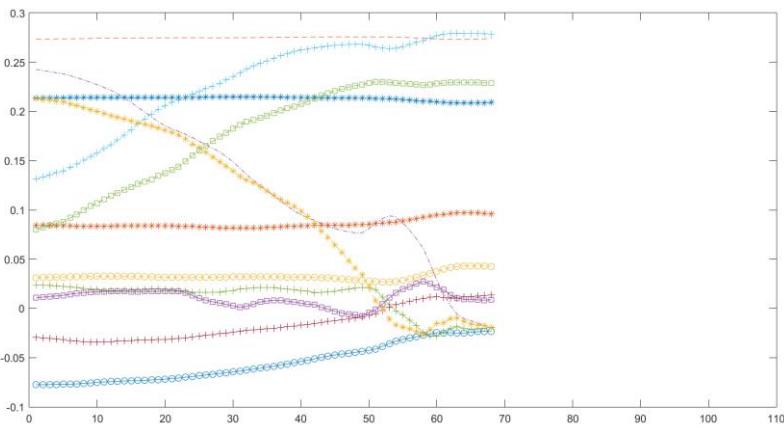
- Trả về biến *TrackNormalized*

End

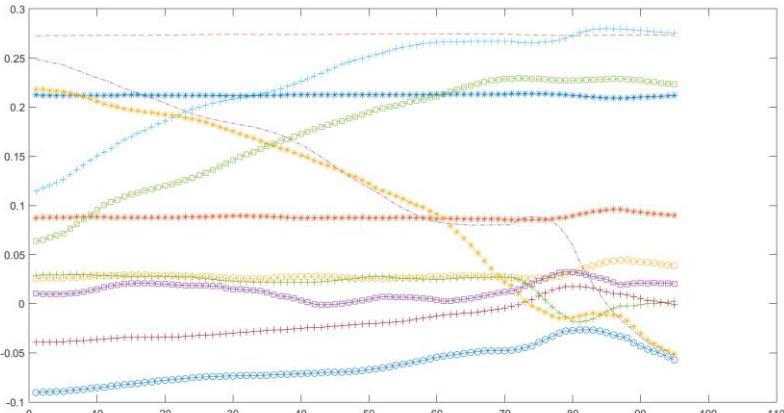
2.5 Bộ lọc trung bình

Khi Kinect thu nhận tín hiệu khung xương từ người tập, việc nhiễu xuất hiện là điều không thể tránh khỏi. Nghiều này đến từ nhiễu trắng (do bản chất không hoàn toàn ổn định của cảm biến) và nhiễu do sự che lấp các chi của bệnh nhân [4]. Hình 12a minh họa tác động của các loại nhiễu lên tín hiệu thu được từ bài tập xoay khuỷu tay.

Để giảm thiểu tác động của nhiễu, tôi đề xuất sử dụng bộ lọc trung bình. Ta xây dựng một cửa sổ có độ dài là 9 frame, quét qua toàn bộ tín hiệu. Tại mỗi vị trí, ta tính trung bình cộng của tín hiệu trong 9 frame liên tiếp, và gán kết quả cho tín hiệu ở frame trung tâm. Việc lọc tín hiệu giúp tránh ảnh hưởng của nhiễu, và cho thấy được xu hướng thay đổi chính của tín hiệu. Hình 12b biểu diễn tín hiệu sau khi áp dụng bộ lọc. Chi phí tính toán cho phép lọc là rất nhỏ cho một chuyển động, do đó không ảnh hưởng nhiều tới tốc độ của cả hệ thống. Thuật toán giảm nhiễu bằng bộ lọc trung bình được trình bày trong Thuật toán 3.

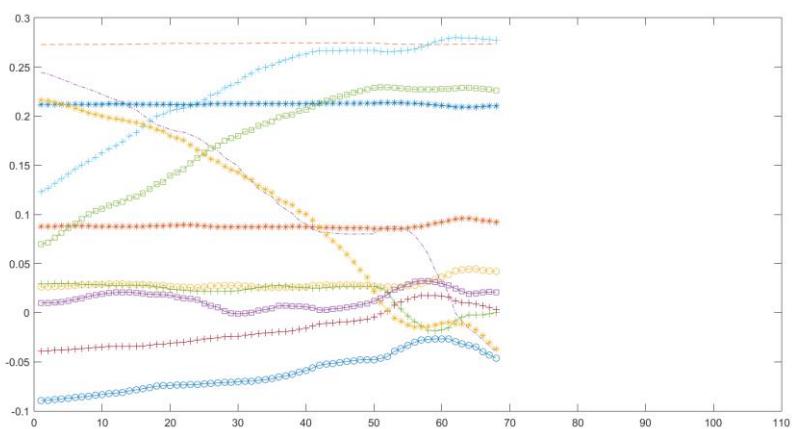


a)



b)

(1) Upper left arm x
(2) Upper left arm y
(3) Upper left arm z
(4) Upper right arm x
(5) Upper right arm y
(6) Upper right arm z
(7) Lower left arm x
(8) Lower left arm y
(9) Lower left arm z
(10) Lower right arm x
(11) Lower right arm y
(12) Lower right arm z



c)

Hình 11. Minh họa thuật toán chuẩn hóa Dynamic Time Warping.

- a) Tín hiệu động tác mẫu
- b) Tín hiệu động tác mới thực hiện sau bước dùng DTW để canh chỉnh
- c) Tín hiệu động tác mới thực hiện sau bước dùng DTW để canh chỉnh và chuẩn hóa

Thuật toán 3. Thuật toán giảm nhiễu bằng bộ lọc trung bình

Input:

OriginalSignal: tín hiệu gốc được thu thập bởi Kinect (sau khi đã chuyển thành hệ tọa độ cơ thể người)

WindowSize: kích cỡ cửa sổ bộ lọc trung bình

Output:

FilteredSignal: tín hiệu sau khi giảm nhiễu bằng bộ lọc trung bình

Thuật toán:

Begin

- Quét i từ $1 \rightarrow NumberOfFrame$ ($NumberOfFrame$ là tổng số frame của tín hiệu gốc *OriginalSignal*)

- Với mỗi i , ta tính *FilteredSignal* như sau:

- Nếu $i \leq [WindowSize / 2]$

$$FilteredSignal[i] = mean(OriginalSignal[1, \dots, i + [WindowSize / 2]])$$

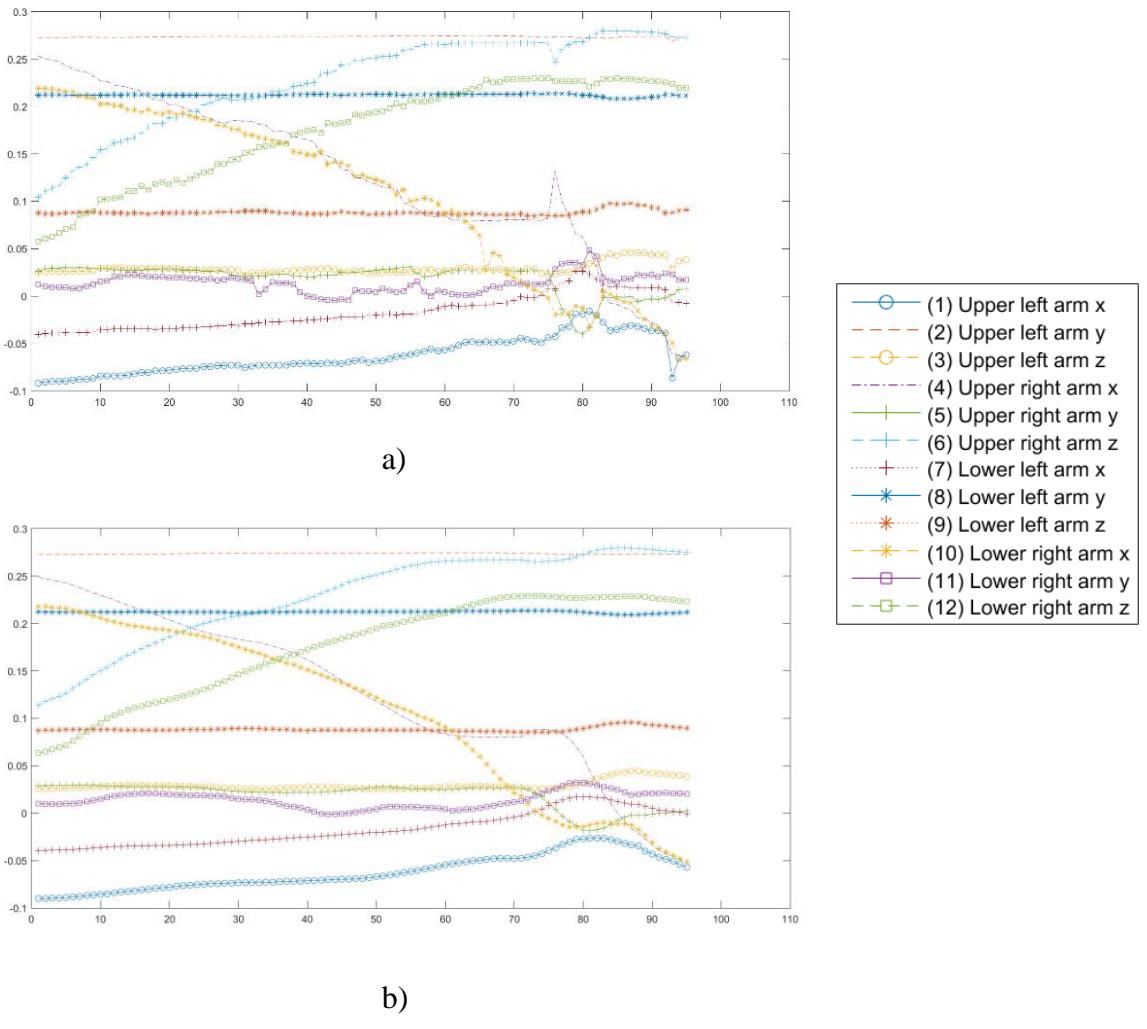
- Nếu $[WindowSize / 2] < i < NumberOfFrame - [WindowSize / 2] + 1$

$$FilteredSignal[i] = mean(OriginalSignal[i - [WindowSize / 2], \dots, i + [WindowSize / 2]])$$

- Nếu $NumberOfFrame - [WindowSize / 2] + 1 \leq i$

$$FilteredSignal[i] = mean(OriginalSignal[i - [WindowSize / 2], \dots, NumberOfFrame])$$

End



Hình 12. Minh họa lọc giảm nhiễu tín hiệu bằng bộ lọc trung bình.

a) Tín hiệu thô chưa qua xử lý. b) Tín hiệu sau khi qua bộ lọc

Chương 3

Xây dựng thuật giải quyết bài toán

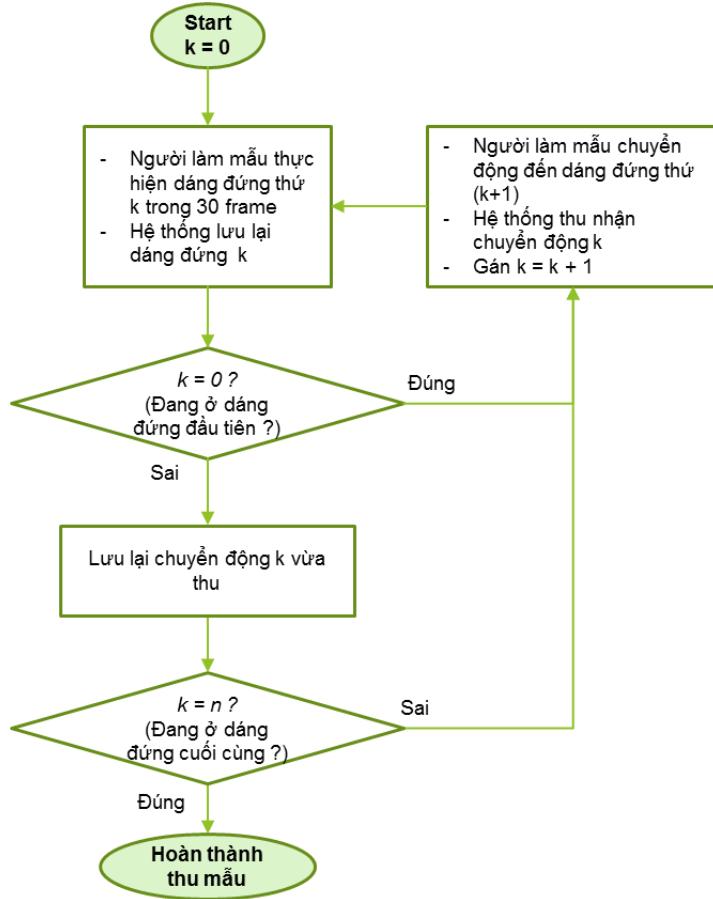
3.1 Thuật toán thu thập thông tin bài tập mẫu

Nhu cầu thu tín hiệu bài tập mẫu là rất cao, đến từ việc bệnh nhân thông thường phải tập nhiều bài tập khi điều trị chấn thương, và với mỗi bài tập, biên độ động tác cũng phải thay đổi liên tục, khó dàn lên tương ứng với tình trạng hồi phục của bệnh nhân. Do đó yêu cầu đặt ra cho hệ thống thu bài tập mẫu là phải dễ dàng sử dụng, nhanh chóng nhất có thể.

Từ yêu cầu này, tôi đề xuất một hệ thống thu tín hiệu được thiết kế hoạt động theo phương thức sau. Với một bài tập bao gồm n dáng đứng và $(n-1)$ chuyển động, để thu tín hiệu mẫu, người làm mẫu lần lượt thực hiện các bước:

- Bước 1: người thực hiện đứng vào tư thế chuẩn bị (dáng 0) trong khoảng 1.5 giây (tương ứng với 30 frame ảnh với tốc độ thu 20 frames/giây). Khi này Kinect và hệ thống sẽ thu nhận tín hiệu dáng 0.
- Bước 2: người thực hiện di chuyển sang tư thế tiếp theo (dáng 1). Hệ thống thu nhận chuyển động này (chuyển động 1).
- Bước 3: người thực hiện dừng lại ở tư thế tiếp theo (dáng 1). Hệ thống thu nhận dáng 1.
- Bước 4: người thực hiện di chuyển sang dáng 2. Hệ thống thu nhận chuyển động 2.
- Bước 5: người thực hiện dừng lại ở dáng 2. Hệ thống thu nhận dáng 2.
-
- Bước $2n - 2$: người thực hiện di chuyển sang dáng n . Hệ thống thu nhận chuyển động n .
- Bước $2n - 1$: người thực hiện dừng lại ở dáng n . Hệ thống thu nhận dáng n .
- Kết thúc việc thu nhận bài tập mẫu.

Lưu đồ 1 biểu diễn các bước thực hiện trên.



Lưu đồ 1. Các bước thu thập một bài tín hiệu mẫu

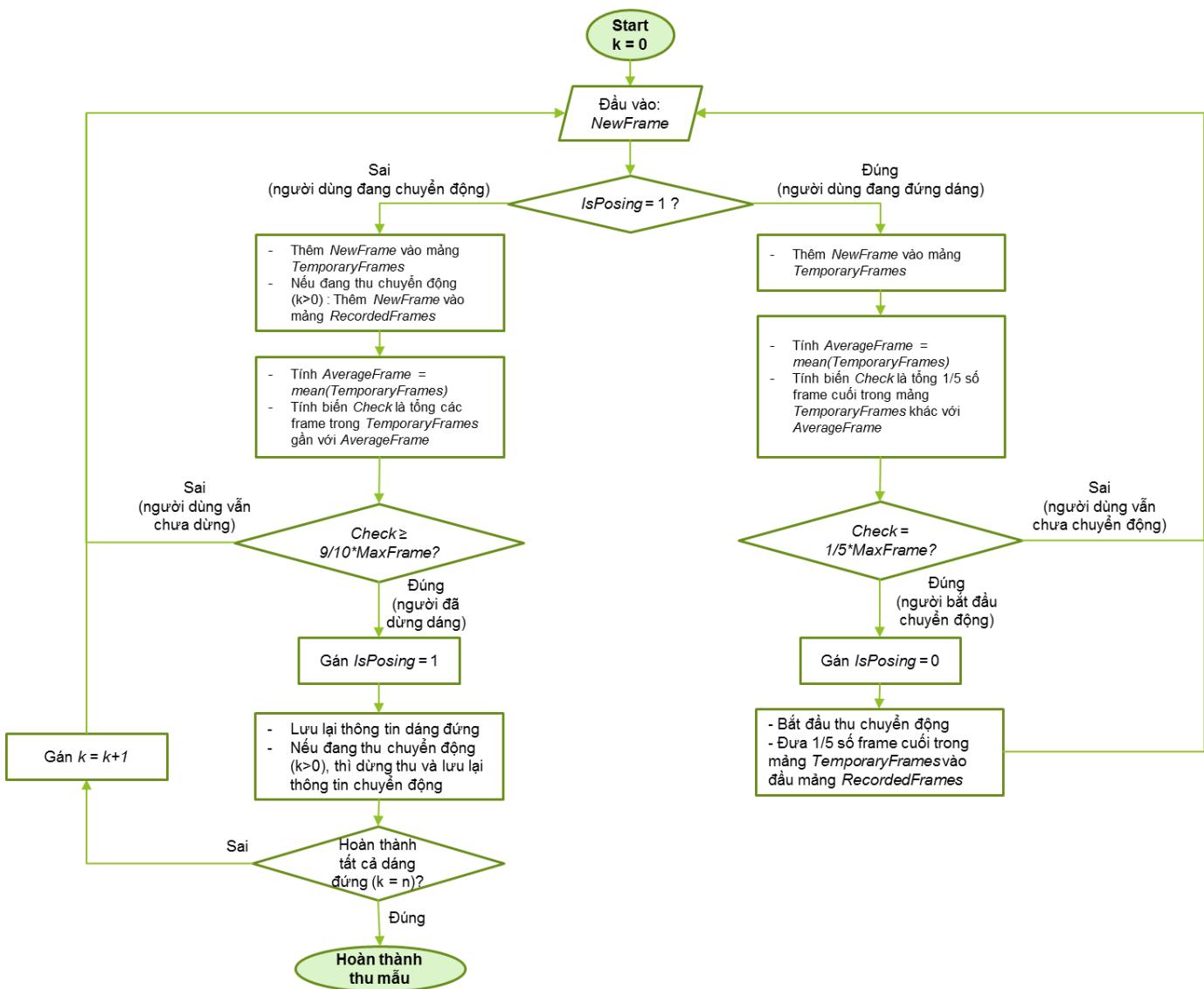
Với cách thức thu nhận tín hiệu này, người làm mẫu sẽ rất dễ dàng và thoải mái khi làm mẫu, hầu như không khác gì so với thực hiện cho bệnh nhân xem.

Để được giao diện sử dụng tiện lợi như vậy, tôi đề xuất hệ thống bên trong sẽ thực hiện như sau:

Ta dùng một mảng *TemporaryFrames* để lưu trữ lại tín hiệu khung xương người tập trong 30 frame gần nhất. Ta tính trung bình tín hiệu khung xương trong 30 frame ảnh này. Sau đó so sánh với từng frame xem có bao nhiêu frame gần với trung bình (độ sai lệch dưới ngưỡng 0.06). Nếu có khoảng 90% trong số 30 frame gần với trung bình thì ta xác định là người thực hiện đang dừng lại ở một dáng đứng nào đó. Khi này hệ thống sẽ thu nhận dáng đứng hiện tại của người thực hiện, sau đó xử lý và lưu vào cơ sở dữ liệu. Lưu ý rằng, hệ thống vẫn luôn thu nhận tín hiệu vào mảng *TemporaryFrames* tại mọi lúc để luôn xác định trạng thái của người tập mẫu.

Khi hệ thống đã thu nhận được dáng đứng của người tập, người tập có thể di chuyển sang dáng đứng tiếp theo. Ta nhận biết được khi nào người tập rời khỏi dáng đứng hiện tại để chuyển động bằng cách xem xét mảng *TemporaryFrames*. Nếu có 20% trong tổng số frame trong *TemporaryFrames* sai lệch nhiều so với frame trung bình cộng (sai nhiều hơn ngưỡng 0.06), thì ta biết rằng người tập đã rời khỏi dáng và bắt đầu chuyển động. Khi này, hệ thống bắt đầu thu chuỗi tín hiệu của chuyển động vào mảng *MotionFrames*.

Người làm mẫu tiếp tục di chuyển cho đến khi đến dáng đứng tiếp theo, tới đây, người làm mẫu lại đứng yên tại dáng đó thêm 30 frame (1.5 giây nữa). Khi này, ta lại so sánh



Lưu đồ 2. Biểu diễn sơ đồ khái thuật toán thu thập tín hiệu mẫu

các frame ảnh trong *TemporaryFrames* với giá trị trung bình, nếu 90% các frame gần với giá trị trung bình, hệ thống nhận biết người tập đã dừng lại, và thu thập tín hiệu dáng dừng hiện tại. Quá trình tiếp tục như vậy cho tới khi dáng đứng cuối cùng (dáng n) được phát hiện. Thuật toán 4 biểu diễn mã giả của hệ thống thu nhận bài tập mẫu. Lưu đồ 2 biểu diễn sơ đồ khái thuật toán này. Hình 13 minh họa cách thuật toán phát hiện các dáng đứng và chuyển động của người tập mẫu.

Thuật toán 4. Thu thập bài tập mẫu

Input:

MaxFrames: số frame xem xét động tác dừng hay chưa. Mặc định là 30 frame.

TemporaryFrames: mảng lưu các frame để xem xét động tác dừng (có độ dài là *MaxFrames*). Hoạt động như cấu trúc hàng đợi Queue. Khi đầy, các frame đầu mảng bị loại bỏ để thêm các frame mới vào cuối mảng.

IsPosing: biến chỉ trạng thái hiện tại của người làm mẫu là dừng, hay đang chuyển động. Giá trị khởi tạo là 0 (chưa đứng dáng).

NewFrame: frame tín hiệu vừa được thu nhận mới.

Output:

RecordedPose: biến lưu dáng đứng tại các điểm dừng của người làm mẫu.

RecordedMotion: mảng lưu chuỗi tín hiệu chuyển động mẫu, có kích cỡ không giới hạn.

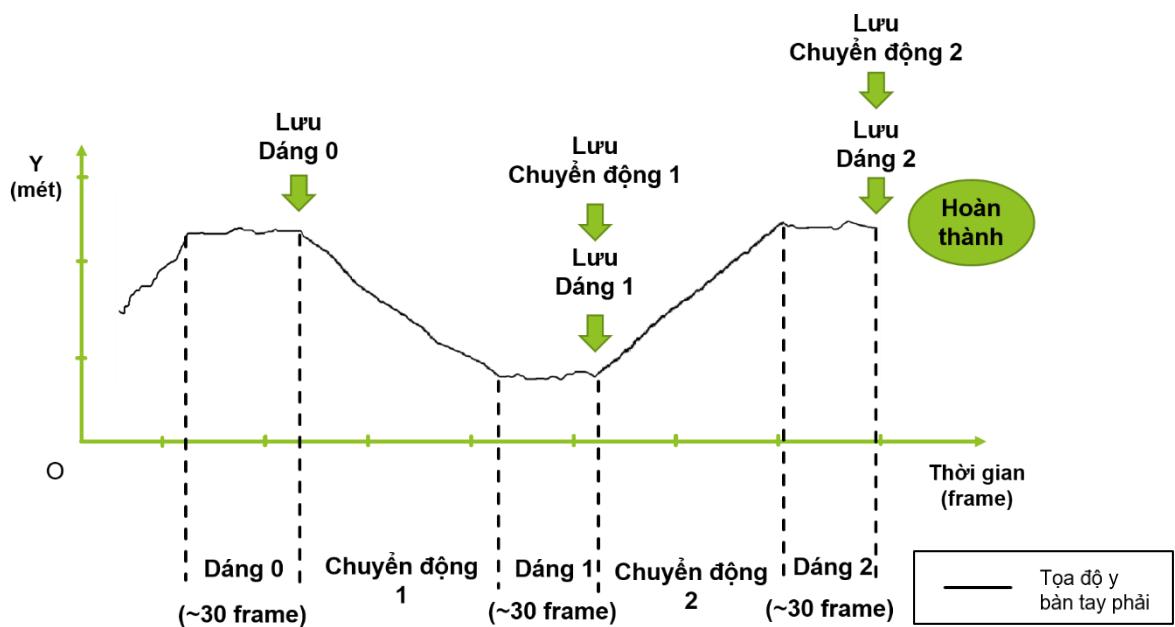
Thuật toán:

Begin

Với mỗi frame ảnh mới *NewFrame* được cung cấp, hệ thống chạy như sau:

- Nếu *IsPosing* = 0 (người tập chưa dừng dáng)
 - + Thêm *NewFrame* vào *TemporaryFrames*. Trong khi *TemporaryFrames* chưa đầy thì chỉ thêm *NewFrame* và dừng thuật toán (Quay lại bước đầu với *NewFrame* mới).
 - + Nếu đang thu chuyển động thì thêm *NewFrame* vào cuối *RecordedMotion*.
 - + Tính trung bình *AverageFrame* toàn mảng *TemporaryFrames*, kiểm tra số frame ở cuối *TemporaryFrames* xem có bao nhiêu frame giống (độ lỗi nhỏ hơn ngưỡng 0.06m) so với *AverageFrame*:
 - Nếu có 9/10 số frame giống,
 - Gán *IsPosing* = 1 (người tập đã dừng dáng)
 - Lưu lại biến *AverageFrame* thành *RecordedPose* làm đặc trưng cho dáng đứng hiện tại.

- Nếu đang thu chuyển động bằng biến *RecordedMotion*, cho dừng thu, xử lý chuỗi tín hiệu này và lưu lại làm đặc trưng cho chuyển động vừa thực hiện.
 - Nếu đã đủ số dáng đứng: kết thúc bài tập. Ngược lại, dừng thuật toán và quay lại từ đầu.
 - Ngược lại, dừng thuật toán quay lại từ đầu do người tập vẫn chưa dừng dáng.
- Nếu *IsPosing* = 1 (người tập đang dừng dáng)
- + Thêm *NewFrame* vào cuối *TemporaryFrames*.
 - + Tính trung bình *AverageFrame* toàn mảng *TemporaryFrames*, kiểm tra 1/5 số frame ở cuối *TemporaryFrames* xem có bao nhiêu frame khác (độ lỗi lớn hơn ngưỡng 0.06) so với *AverageFrame*:
 - Nếu tất cả 1/5 số frame khác,
 - Gán *IsPosing* = 0 (người tập bắt đầu chuyển động)
 - Gán 1/5 số frame cuối của *TemporaryFrames* vào đầu biến *RecordedMotion* (bắt đầu thu tín hiệu chuyển động)
 - Ngược lại, dừng thuật toán quay lại từ đầu do người tập vẫn chưa chuyển động.
- End*



Hình 13. Minh họa thuật toán thu nhận bài tập

3.2 Thuật toán xử lý thông tin bài tập mẫu

Sử dụng thuật toán thu nhận thu nhận bài mẫu như trên, ta sẽ thu được tín hiệu các dáng đứng và chuyển động bao gồm: tín hiệu các dáng đứng được thu trong 30 frame người tập đứng yên trong mảng *TemporaryFrames*; chuỗi tín hiệu chuyển động thu được trong mảng *MotionFrames*. Tôi đề xuất phương thức xử lý tín hiệu, trích đặc trưng các dáng đứng và chuyển động như sau. Thuật toán 5 mô tả mã giả của thuật toán được đề xuất.

3.2.1 Xử lý thông tin dáng đứng

Trong một lần thu bài tập, với mỗi dáng đứng, ta có 30 frame ổn định của dáng đứng thu được từ mảng *TemporaryFrames*, ta sẽ lấy trung bình của các frame này để làm dáng đứng chuẩn nhất. Lưu ý rằng, vì người tập mẫu sẽ thực hiện một bài tập nhiều lần khi thu mẫu, nên tín hiệu một dáng đứng chuẩn sẽ được lấy nhiều lần. Ta cũng sẽ lấy trung bình cộng tín hiệu dáng đứng của những lần này để làm tín hiệu chuẩn trung bình và lưu vào cơ sở dữ liệu của bài tập.

3.2.2 Xử lý thông tin chuyển động

Người tập mẫu sẽ thực hiện một bài tập nhiều lần, và hệ thống sẽ dựa trên những lần đó để chọn ra lần tập mẫu tốt nhất. Thông thường, người tập mẫu cần thực hiện mẫu một bài tập ít nhất 5 lần, và càng nhiều lần càng tốt. Như vậy với mỗi chuyển động trong bài tập, ta sẽ có ít nhất 5 mẫu. Ta sẽ xử lý thông tin các chuyển động như sau:

- Bước 1. Chọn chuyển động chuẩn

Với mỗi chuyển động, ta sẽ tính độ lỗi của nó với tất cả các chuyển động còn lại (sử dụng phép Dynamic Time Warping chuẩn hóa được giới thiệu ở phần 2.4). Sau đó ta chọn ra chuyển động mà tổng độ lỗi của nó so với các chuyển động khác gần với trung vị nhất. Ta gọi chuyển động này là chuyển động chuẩn *StandardSignal*. Lý do ta không chọn chuyển động có độ lỗi nhỏ nhất là do việc này quá tối ưu hóa độ lỗi chuyển động chuẩn, làm bệnh nhân rất khó thực hiện lại chuyển động chuẩn khi tập ở nhà. Bảng 1 minh họa việc tính độ lỗi của từng chuyển động so với các chuyển động còn lại, và chọn ra chuyển động chuẩn. Chuyển động được minh họa là chuyển động 1 của bài tập đang khép nắp ngang.

- Bước 2. Tính trung bình và độ lệch chuẩn độ lỗi của chuyển động chuẩn với các chuyển động còn lại

Khi đã tìm được chuyển động chuẩn, ta tính lại độ lỗi của chuyển động chuẩn với

các chuyển động còn lại. Khi này, những lần thực hiện có độ lỗi quá lớn so với trung vị các độ lỗi sẽ bị cho là lần thực hiện sai, sẽ bị lược bỏ. Sau đó ta tính giá trị trung bình μ và độ lệch chuẩn σ các độ lỗi của những chuyển động còn lại (không bị lược bỏ). Hai giá trị μ và σ được lưu lại vào cơ sở dữ liệu để làm đặc trưng cho chuyển động này. *Bảng 2* minh họa bước này.

Bảng 1. Bảng tính độ lỗi giữa các lần thực hiện

Kết quả cho thấy trung vị của tổng độ lỗi nằm ở lần 3, do đó lần 3 được chọn là chuyển động chuẩn

	Lần 1	Lần 2	Lần 3	Lần 4	Lần 5	Tổng lỗi
Lần 1	0	0.9274	1.1667	0.8882	2.1629	5.1451
Lần 2	1.2013	0	1.1418	1.3910	2.7119	6.4460
Lần 3	1.2292	0.9162	0	1.2885	2.3364	5.7703
Lần 4	0.8995	1.0922	1.2067	0	2.1060	5.3044
Lần 5	3.1156	3.1735	3.2772	3.0938	0	12.661

Bảng 2. Bảng tính độ lỗi giữa các lần thực hiện so với lần thực hiện chuẩn.

Và trung bình / độ lệch chuẩn các độ lỗi. Chú ý rằng lần thực hiện thứ 5 bị loại bỏ do độ lỗi quá lớn so với những lần còn lại (bị xem là chuyển động mâu sai)

Lần 3	1.2292	0.9162	0	1.2885	2.3364
μ			1.1446		
σ			0.2001		

- Bước 3. Tính thời gian trung bình của chuyển động
Bên cạnh thông tin độ lỗi, ta còn tính trung bình độ dài thời gian thực hiện chuyển động (qua đếm số frame của mỗi chuyển động) để xác định thời gian trung bình t thực hiện chuyển động đó.

Như vậy, với một chuyển động, các đặc trưng ta lưu vào cơ sở dữ liệu có dạng (4).

$$MovementFeature = \{StandardMotion, \mu, \sigma, t\} \quad (4)$$

Trong đó,

$StandardMotion$ là chuỗi tín hiệu của lần thực hiện chuẩn.

μ là trung bình độ lỗi của các lần thực hiện so với lần thực hiện chuẩn.

σ là độ lệch chuẩn độ lỗi của các lần thực hiện so với lần thực hiện chuẩn.

t là trung bình thời gian của các lần thực hiện.

Thuật toán 5. Thuật toán xử lý dữ liệu dáng đứng và chuyển động

Input:

Times: cấu trúc lưu các lần thực hiện bài tập mẫu, là một mảng $[1,.., NumberOfTimes]$. Mỗi phần tử *Times[i]* có hai thành phần, *RecordedPose* và *RecordedMotion*.

Times[i].RecordedPose: dáng đứng được thu nhận và lưu lại ở lần thực hiện thứ *i*
Times[i].RecordedMotion: chuyển động được thu nhận và lưu lại và lưu lại ở lần thực hiện thứ *i*

Output:

StandardPose: các dáng đứng chuẩn của động tác

StandardMotion: các chuyển động chuẩn của động tác

MeanLossMotion: trung bình độ lỗi của các lần thực hiện chuyển động

StdLossMotion: độ lệch chuẩn của các lần thực hiện chuyển động

TimeMotion: trung bình thời gian chuyển động

Thuật toán:

Begin

● Xử lý dữ liệu dáng đứng

- Tính trung bình dáng đứng

$$StandardPose = mean(Times[1,.., NumberOfTimes].RecordedPose)$$

- Lưu lại biến *StandardPose* làm dáng đứng chuẩn

● Xử lý dữ liệu chuyển động

- Lập bảng $Loss[(1,.., NumberOfTimes) \times (1,.., NumberOfTimes)]$

- Quét qua các lần thực hiện, i từ $1 \rightarrow NumberOfTimes$, j từ $1 \rightarrow NumberOfTimes$

➢ Canh chỉnh và chuẩn hóa DTW tín hiệu *Times[i].RecordedPose* so với tín hiệu *Times[j].RecordedPose*. Cập nhật độ lỗi giữa hai tín hiệu vào bảng *Loss* tại vị trí *Loss[i,j]*.

- Với mỗi *i*, tính tổng độ lỗi so với các lần thực hiện *j* (*j* từ $1 \rightarrow NumberOfTimes$)

- Chọn ra lần thực hiện *i* có tổng độ lỗi là trung vị, chọn nó là chuyển động chuẩn *StandardMotion*

- Tính độ lỗi DTW chuẩn hóa giữa *StandardMotion* với những lần thực hiện còn

lại. Xóa bỏ các lần thực hiện có độ lỗi quá lớn (các ngoại biên)

- Tính trung bình *MeanLossMotion* và độ lệch chuẩn *StdLossMotion* của độ lỗi các lần thực hiện (sau khi đã loại bỏ ngoại biên)

- Lưu lại các biến *StandardMotion*, *MeanLossMotion* và *StdLossMotion*.

- Tính trung bình thời gian các lần thực hiện (sau khi đã loại bỏ ngoại biên), lưu vào biến *TimeMotion*

End

3.3 Thuật toán thu thập và đánh giá bài tập do bệnh nhân thực hiện

Tôi đề xuất thuật toán thu thập và đánh giá bài tập do bệnh nhân thực hiện như sau.

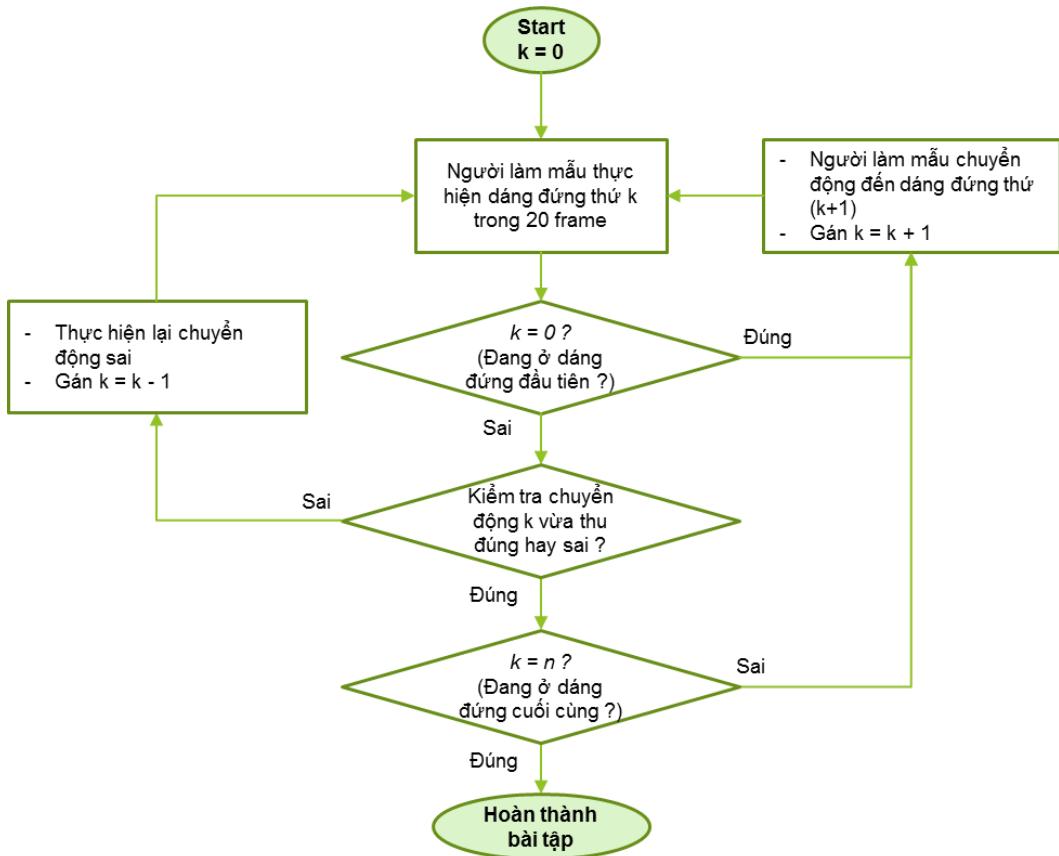
3.3.1 Thuật toán thu thập tín hiệu bài tập

Khi bệnh nhân tập ở nhà, bệnh nhân sẽ thực hiện các bước sau.

Bắt đầu, với $k = 0$:

- Bước 1: Đứng vào dáng thứ k trong 20 frame (tương ứng với hơn 1 giây). Hệ thống kiểm tra xem bệnh nhân có đứng đúng dáng hay không.
- Bước 2: Bắt đầu di chuyển tới dáng đứng thứ $k+1$. Hệ thống thu lại tín hiệu chuyển động vào mảng *MotionFrames*. Trong quá trình di chuyển, nếu thời gian vượt quá 1.5 lần thời gian trung bình của chuyển động, hệ thống sẽ báo sai (động tác quá dài) và buộc bệnh nhân thực hiện lại.
- Bước 3: Bệnh nhân dừng ở dáng thứ $k+1$ trong khoảng 1 giây. Hệ thống kiểm tra xem bệnh nhân có đứng đúng dáng hay không. Nếu đúng, dừng thu tín hiệu chuyển động, và đem chuỗi tín hiệu *MotionFrames* đi so sánh với chuyển động chuẩn *StandardSignal* được lưu trong cơ sở dữ liệu. Chi tiết về phép so sánh, đánh giá độ chính xác của chuyển động được trình bày trong phần 3.3.2.
- Bước 4: Nếu chuyển động thu được được đánh giá là đúng, quay lại bước 1 với $k \leftarrow k + 1$ (bệnh nhân tiếp tục thực hiện chuyển động kế tiếp). Nếu chuyển động được đánh giá là sai, quay lại bước 1 với k không đổi (bệnh nhân thực hiện lại chuyển động).

Lưu đồ 3 biểu diễn các bước cần thực hiện khi đánh giá bài tập mới. Lưu đồ 4 biểu diễn sơ đồ khởi thuật toán này. Thuật toán 6 mô tả mã giả thuật toán thu nhận và đánh giá bài tập mới.



Lưu đồ 3. Các bước hệ thống thực hiện khi đánh giá bài tập mới

3.3.2 Thuật toán đánh giá bài tập

- Đánh giá dáng đứng

Hệ thống so sánh dáng đứng hiện tại với dáng đứng trung bình được tính toán từ các lần thực hiện mẫu, nếu độ sai khác tính bằng khoảng cách Euclid giữa hai tín hiệu nhỏ hơn ngưỡng 0.06 (6 cm) thì ta đánh giá đây là dáng đứng đúng.

- Đánh giá chuyển động

Với chuỗi tín hiệu *MotionFrames* vừa thu được, ta so sánh tính độ lỗi với chuyển động chuẩn *StandardSignal* (sử dụng Dynamic Time Warping chuẩn hóa). Nếu độ lỗi *Loss* tính được nằm trong khoảng $(\mu - 1.5 \times \sigma, \mu + 1.5 \times \sigma)$

thì ta đánh giá là chuyển động được thực hiện đúng. Ngược lại, chuyển động được cho là sai và bệnh nhân cần thực hiện lại chuyển động.

Bên cạnh việc thông báo chuyển động vừa thực hiện là đúng hay sai, hệ thống còn chia ra các mức độ chính xác – rất tốt, tốt, tạm – để đánh giá định lượng hơn về độ chính xác của chuyển động. Công thức (5) định nghĩa các mức độ

đánh giá chất lượng bài tập. Ta đánh giá dựa vào giá trị $f(Loss)$:

✧ $f(Loss) = 3$ – Rất tốt

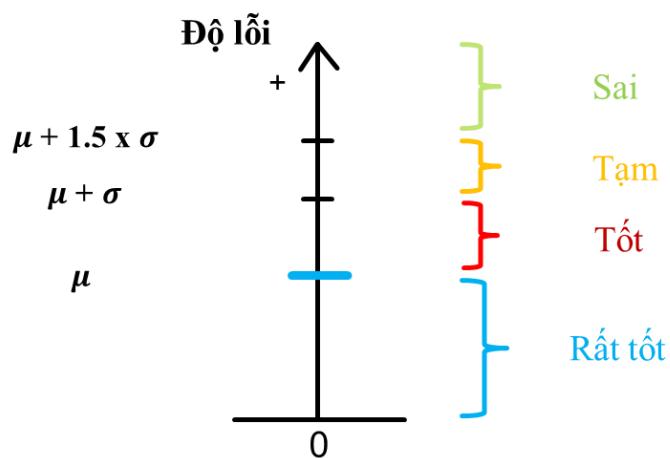
✧ $f(Loss) = 2$ – Tốt

✧ $f(Loss) = 1$ – Tạm

✧ $f(Loss) = 0$ – Sai

Hình 14 minh họa cách thức phân loại.

$$f(Loss) = \begin{cases} 3 & \text{nếu } Loss \leq \mu \\ 2 & \text{nếu } \mu < Loss \leq \mu + \sigma \\ 1 & \text{nếu } \mu + \sigma < Loss \leq \mu + 1.5\sigma \\ 0 & \text{nếu } \mu + 1.5\sigma < Loss \end{cases} \quad (5)$$



Hình 14. Cách thức phân loại chất lượng bài tập

Thuật toán 6. Thuật toán đánh giá bài tập mới

Input:

StandardPose: các dáng đứng chuẩn của động tác

StandardMotion: các chuyển động chuẩn của động tác

MeanLossMotion: trung bình độ lỗi của các lần thực hiện chuyển động

StdLossMotion: độ lệch chuẩn của các lần thực hiện chuyển động

IsPosing: biến chỉ trạng thái hiện tại của người làm mẫu là đang đứng dáng, hay đang chuyển động. Giá trị khởi tạo là 0 (chưa đứng dáng).

NewFrame: frame tín hiệu vừa được thu nhận mới.

MaxFrames: số frame xem xét bệnh nhân có vào tư thế chuẩn chưa. Mặc định là 20 frame.

TemporaryFrames: mảng lưu các frame để xem xét động tác dừng (có độ dài là *MaxFrames*). Hoạt động như cấu trúc hàng đợi Queue. Khi đầy, các frame đầu mảng bị loại bỏ để thêm các frame mới vào cuối mảng.

Output:

Result: kết quả đúng hay sai của chuyển động vừa thực hiện

Thuật toán:

Begin

Với mỗi frame ảnh mới *NewFrame* được cung cấp, hệ thống chạy như sau:

- Nếu *IsPosing* = 0 (người tập chưa dừng dáng)
 - + Thêm *NewFrame* vào *TemporaryFrames*. Nếu *TemporaryFrames* vẫn chưa đủ *MaxFrames*, dừng thuật toán (quay lại bước đầu với *NewFrame* mới)
 - + Nếu đang thu chuyển động thì thêm *NewFrame* vào cuối *RecordedMotion*.
 - + Kiểm tra số frame trong *TemporaryFrames* xem có bao nhiêu frame giống (độ lỗi nhỏ hơn ngưỡng 0.06) so với *StandardPose*:
 - Nếu có 9/10 số frame giống,
 - Gán *IsPosing* = 1 (người tập đã dừng dáng)
 - Nếu đang thu chuyển động (đang thu tín hiệu vào biến *RecordedMotion*), thì cho dừng thu.
 - Kiểm tra xem tín hiệu vừa thu *RecordedMotion* có tương đồng với động tác mẫu *StandardMotion* hay không (kiểm tra xem độ lỗi giữa hai tín hiệu có nhỏ hơn *MeanLossMotion* + $1.5 \times StdLossMotion$ hay không).
 - Nếu nhỏ hơn, thì chuyển động đúng, kết quả trả về **Result = 1**. Người tập thực hiện chuyển động tiếp theo. Nếu lớn hơn, chuyển động sai, **Result = 0**, thông báo

và cho người tập thực hiện lại chuyển động.

- Ngược lại, dùng thuật toán và quay lại từ đầu do người tập vẫn chưa dừng dáng.

➤ Nếu *IsPosing* = 1 (người tập đang dừng dáng)

+ Thêm *NewFrame* vào cuối *TemporaryFrames*.

+ Kiểm tra 1/5 số frame ở cuối *TemporaryFrames* xem có bao nhiêu frame khác (độ lỗi lớn hơn ngưỡng 0.06) so với *StandardPose*:

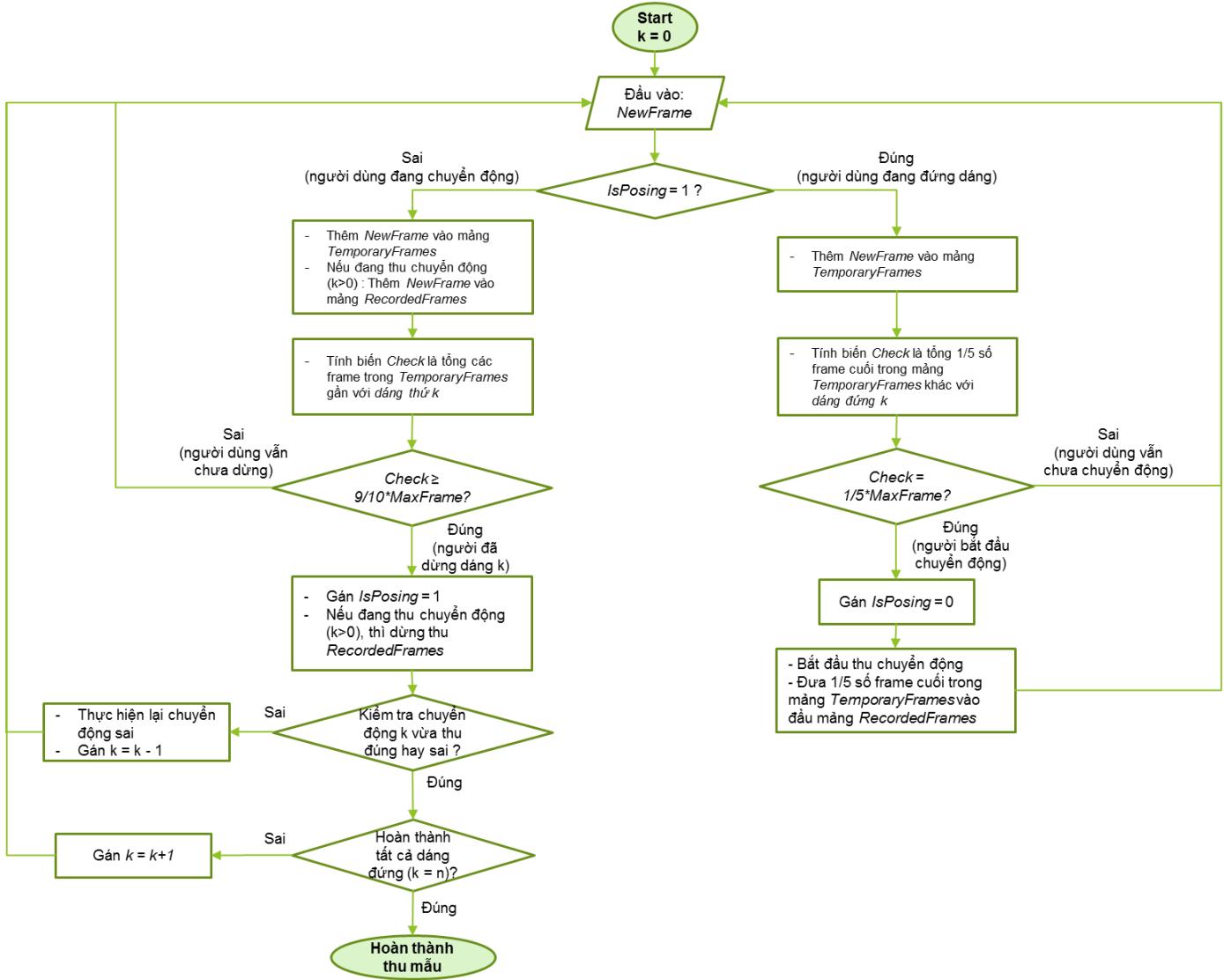
- Nếu tất cả 1/5 số frame đều khác,

- Gán *IsPosing* = 0 (người tập bắt đầu chuyển động)

- Gán 1/5 số frame cuối của *TemporaryFrames* vào biến *RecordedMotion* (bắt đầu thu tín hiệu chuyển động)

- Ngược lại, dùng thuật toán và quay lại từ đầu do người tập vẫn chưa chuyển động.

End



Lưu đồ 4. Biểu diễn sơ đồ khái thuật toán đánh giá bài tập mới

3.4 Các tính năng đặc biệt

3.4.1 Thuật toán hướng dẫn dáng đứng tiếp theo

Nhằm giúp bệnh nhân thực hiện các bài tập dễ dàng hơn. Tôi đề xuất áp dụng ý tưởng thực tế tăng cường (augmented reality) để hiện thị các dáng đứng, vị trí các chi bệnh nhân cần thực hiện để hoàn thành tốt bài tập.

Tại mỗi thời điểm, ta sẽ vẽ chòng lên màn hình dáng đứng tiếp theo bệnh nhân cần di chuyển tới. Việc này giúp cho bệnh nhân dễ dàng xác định, canh chỉnh cơ thể để thực hiện đúng động tác.

Tọa độ các chi trong dáng đứng tiếp theo đã được lưu trữ trong cơ sở dữ liệu, dưới dạng tín hiệu tọa độ với hệ trục tọa độ là cơ thể người (như trình bày trong phần 2.3.2). Hệ thống sẽ truy xuất dữ liệu này, và chuyển đổi ngược lại ra tọa độ trong không gian ba chiều với hệ tọa độ gắn trên Kinect. Từ tọa độ này, ta vẽ khung xương của dáng đứng tiếp theo lên màn hình. Hình 15 minh họa chức năng này của hệ thống.

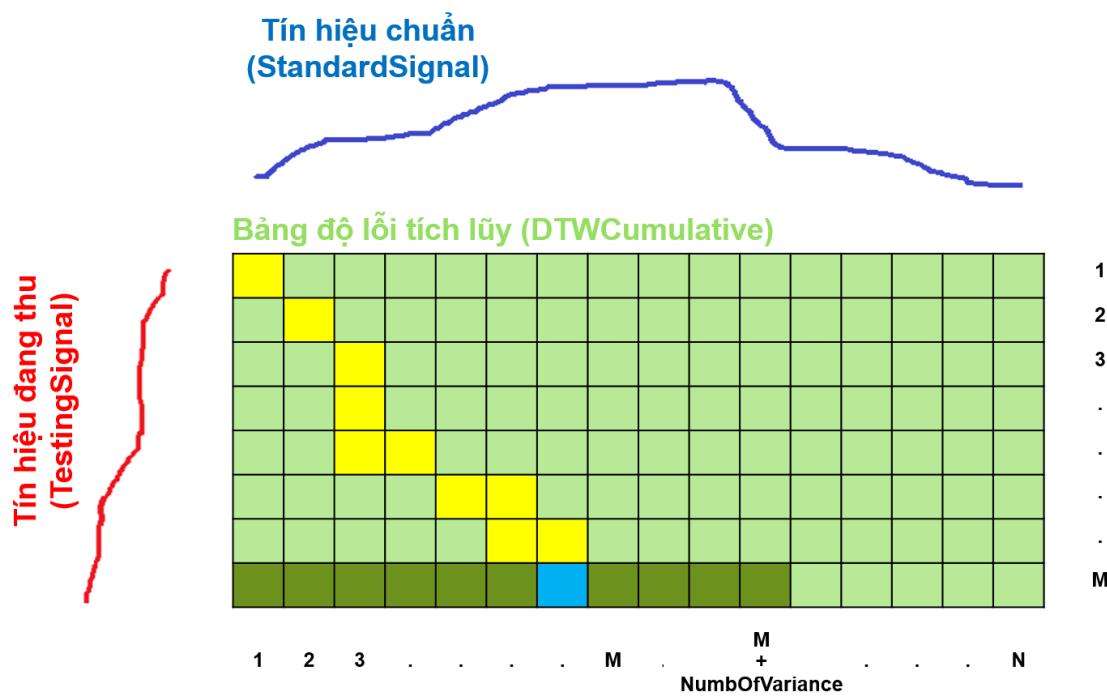


*Hình 15. Minh họa chức năng hướng dẫn dáng đứng tiếp theo
(Khung xương màu vàng chỉ dáng đứng tiếp theo)*

3.4.2 Thuật toán hướng dẫn vị trí động tác tiếp theo

Chức năng này hướng dẫn trí tiếp theo của các chi trong chuyển động. Cụ thể, dựa trên chuyển động mẫu, hệ thống sẽ xác định xem bệnh nhân đang ở thời điểm nào của chuyển động mẫu. Từ đó, xác định xem trong 10 frame nữa, vị trí của chi sẽ ở đâu. Và vẽ chi này lên màn hình nhằm giúp người tập biết được mình cần di chuyển thế nào tiếp theo. Chức năng cũng giúp người tập nhận biết được mình đang thực hiện sai chuyển động (nếu có) và ngay lập tức sửa lại cho chính xác.

Hệ thống nhận biết được bệnh nhân đang tập tới phần nào của chuyển động bằng cơ chế xét Dynamic Time Warping cục bộ. Nghĩa là thay vì đợi cho tới khi bệnh nhân hoàn thành động tác, mới dùng DTW để canh chỉnh với bài tập mẫu. Nay, với mỗi frame tín hiệu mới thu nhận được cho vào *MotionFrame*, ta sẽ ngay lập tức dùng DTW để so sánh *MotionFrame* vừa cập nhật này với bài tập mẫu. Từ đó xác định xem người tập đang đi tới thời điểm nào so với bài tập mẫu, đồng thời xác định được chuyển động đã được hoàn thành bao nhiêu phần trăm. Chúng tôi đề xuất thuật toán Dynamic Time Warping cục bộ này trong Thuật toán 7. *Hình 16* minh họa cách thức hoạt động của thuật toán DTW cục bộ.



Hình 16. Minh họa thuật toán Dynamic Time Warping cục bộ.
 Ô màu xanh lá đậm: vùng để tìm độ lỗi tích lũy nhỏ nhất giữa hai tín hiệu hiện tại
 Ô màu xanh dương: TimePoint tìm được
 Ô màu vàng: con đường canh chỉnh tốt nhất giữa hai tín hiệu tìm thấy được

Thuật toán 7. Thuật toán Dynacmic Time Warping cục bộ

Input:

NewFrame: tín hiệu khung xương tại frame mới nhất

TestingSignal: chuỗi tín hiệu đang thu

StandardSignal: chuỗi tín hiệu chuẩn (độ dài N)

DTWCumulative: bảng độ lỗi tích lũy của hai tín hiệu

NumbOfVariance: biến quy định khoảng tìm kiếm của DTW

Output:

TimePoint: thời điểm tương ứng so với tín hiệu chuẩn

DTWTrack: đường đi gần nhất canh chỉnh giữa hai tín hiệu hiện tại

Thuật toán:

Begin

Với mỗi frame mới, ta tìm đường canh chỉnh DTW cục bộ như sau:

- Cập nhật frame ảnh mới *NewFrame* vào chuỗi tín hiệu đang thu *TestingSignal*.
- Tính độ lỗi giữa *NewFrame* với tất cả các frame trong *StandardSignal*, sau đó cập nhật các độ lỗi này vào bảng *DTWCumulative*.
- Gọi độ dài hiện tại của *TestingSignal* là *M*. Ta xét mảng

$$DTWCumulative[M, 1 \dots (M + NumbOfVariance)]$$

Ta tìm vị trí trong mảng trên có độ lỗi bé nhất. Đây là vị trí canh chỉnh tốt nhất giữa hai tín hiệu đang thu và tín hiệu chuẩn. Ta lưu vị trí này vào biến *TimePoint*.

- Từ vị trí *TimePoint*, ta lại truy ngược về phần tử $[0, 0]$ của *DTWCumulative* để xác định đường đi canh chỉnh giữa hai tín hiệu. Ta lưu con đường này vào biến *DTWTrack*.

End

Từ thông tin này ta xây dựng hai tính năng rất có ích cho hệ thống:

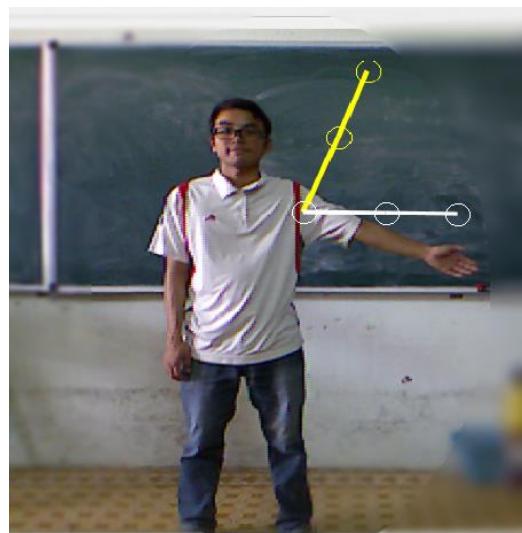
- Cho biết người tập đã hoàn thành bao nhiêu phần của bài tập, bằng công thức (6):

$$CompletedPercent = \frac{TimePoint}{\text{length of StandardSignal}} \times 100 \quad (6)$$

- Dự đoán được sau 10 frame nữa, vị trí chi ở đâu, từ đó đưa ra hướng dẫn cho người tập. Cụ thể, sau 10 frame, vị trí chi được dự đoán được tính theo công thức (7).

$$NextPositionPrediction = StandardSignal[TimePoint + 10] \quad (7)$$

Hình 17 minh họa tính năng dự đoán đặc biệt này.



*Hình 17. Minh họa chức năng hướng dẫn bệnh nhân vị trí chi tiếp theo
(Khung xương màu trắng chỉ vị trí tay tiếp theo)*

Chương 4

Kết quả thực nghiệm và hướng phát triển

4.1 Cách thức đánh giá kết quả thực nghiệm

Ta sẽ đánh giá khả năng phân biệt tính đúng sai của động tác dựa trên bốn bài tập được nêu ra ở đầu bài: dang khép đứng, dang khép nằm, xoay khuỷu tay và gấp duỗi. Đối tượng chữa trị của ta ở đây là thanh niên bình thường, không tàn tật, nhưng vì chấn thương, hoặc bệnh mà có nhu cầu chữa trị bằng vận động trị liệu. Tất cả mọi người tham gia vào thử nghiệm đều là thanh niên.

Với mỗi người tham gia thử nghiệm, đầu tiên ta yêu cầu họ tự thu lại các động tác mẫu đúng của bài tập đang xét, ít nhất là 5 lần cho mỗi bài tập. Sau đó yêu cầu họ tiến hành tập lại các bài tập đó để hệ thống đánh giá. Các “bệnh nhân” sẽ có tình thực hiện các bài tập có lúc đúng có lúc sai, nhằm kiểm tra khả năng phân biệt đúng sai của hệ thống. VỚI MỖI BÀI TẬP, BỆNH NHÂN SẼ THỰC HIỆN 20 LẦN, TRONG ĐÓ CÓ 10 LẦN CÓ TÌNH LÀM ĐÚNG VÀ 10 LẦN CÓ TÌNH LÀM SAI. 10 LẦN LÀM SAI SẼ BAO GỒM NHỮNG LỖI: SAI VỀ MẶT THỜI GIAN (QUÁ NHANH HAY QUÁ CHẬM) VÀ VỀ MẶT BIÊN ĐỘ (SAI QUỸ ĐẠO, SAU BIÊN ĐỘ,...).

Tất cả những người tham gia thí nghiệm (cả người làm mẫu và những người làm bệnh nhân), đều là thanh niên bình thường, có độ tuổi từ 20-23 tuổi, không tàn tật, có cả nam lẫn nữ với các vóc dáng khác nhau.

Cụ thể, với mỗi bài tập, ta sẽ tính các đại lượng sau:

- Dương tính đúng (True Positive) – số lượng lần người tập thực hiện đúng và hệ thống cũng phân loại là đúng, được tính bằng công thức (8).

$$TP = \frac{\sum_{i=1}^{10} P_i}{10} (\times 100\%) \quad (8)$$

Với $P_i = 1$ nếu ở lần thứ i, người tập thực hiện đúng bài tập, và hệ thống cũng đánh giá

là đúng; $P_i = 0$ nếu người tập thực hiện đúng bài tập, và hệ thống cũng đánh giá là sai.

- Âm tính sai (True Negative) – số lượng lần người tập thực hiện sai và hệ thống cũng phân loại là sai, được tính bằng công thức (9).

$$TN = \frac{\sum_{i=1}^{10} N_i}{10} (\times 100\%) \quad (9)$$

Với $N_i = 1$ nếu ở lần thứ i, người tập thực hiện đúng bài tập, và hệ thống cũng đánh giá

là đúng; $N_i = 0$ nếu người tập thực hiện đúng bài tập, và hệ thống cũng đánh giá là sai.

Trong phần thử nghiệm này, ta định nghĩa một bài tập là sai như sau. Giả sử bài tập gồm (n) dáng đứng và (n) chuyển động. Bệnh nhân luôn luôn phải đạt được các dáng đứng được yêu cầu. Trong quá trình di chuyển từ dáng đứng 0 cho tới dáng đứng tiếp (n), nếu bệnh nhân thực hiện sai một chuyển động, toàn bộ bài tập sẽ bị cho là sai, và lần tập hiện tại sẽ dừng lại, bắt đầu lại từ đầu. Ngược lại, nếu tất cả chuyển động được thực hiện đúng thì bài tập được cho là đúng.

Nếu ta quy ước M_i là kết quả đánh giá chuyển động thứ i của bệnh nhân, ta có:

- $M_i = 1$ nếu chuyển động thứ i đúng
- $M_i = 0$ nếu chuyển động thứ i sai

Thì kết quả đúng / sai R của toàn bộ bài tập sẽ là (10).

$$R = \min(M_1, M_2, \dots, M_n) \quad (10)$$

Với công thức trên, thì nếu toàn bộ chuyển động đúng, ta có $R = 1$; ngược lại, chỉ cần có một chuyển động sai, ta có $R = 0$.

Lưu ý: do đây là bước thử nghiệm, nên để đánh giá khách quan, mỗi khi có một chuyển động sai, ta quy định là toàn bộ bài tập sai và buộc bỏ qua lần tập và thực hiện lại từ đầu. Nhưng trong thực tế, khi bệnh nhân tập sai một chuyển động, ta không hủy bỏ bước bệnh nhân tập lại từ đầu, mà chỉ cho bệnh nhân tập lại động tác vừa sai mà thôi, tránh việc bệnh nhân cảm thấy mệt mỏi và chóng nản.

Hệ thống được thử nghiệm với Kinect Xbox 360 Microsoft, được lập trình bằng ngôn ngữ Matlab phiên bản R2015a, sử dụng bộ thư viện Image Acquisition Toolbox. Máy dùng để chạy thuật toán HP Envy, CPU Intel Core i7 5500U, RAM 8GB.

4.2 Kết quả thực nghiệm và nhận xét

Bảng 3 thể hiện kết quả thực nghiệm của hệ thống trên một người tham gia thử nghiệm. Kết quả trên cho thấy rằng hệ thống làm việc khá tốt trong việc đánh giá một chuyển động là đúng hay sai, đặc biệt nhận biết tốt khi động tác làm sai. Hệ thống phân biệt rõ được các loại lỗi về mặt thời gian, và biên độ.

Hệ thống chạy ổn định trong thời gian thực với tốc độ xử lý 15 frame/s cho bước thu thập bài tập mẫu; và 15 frame/s cho bước thu nhận và đánh giá bài tập mới.

Bảng 3. Kết quả thực nghiệm của hệ thống.

(Đ/Đ – Dương tính đúng (True Positive); S/S – Âm tính đúng (True Negative))

HUY								
	Tay XY		Tay XZ		Tay YZ		Khuỷu tay	
	Đ/Đ	S/S	Đ/Đ	S/S	Đ/Đ	S/S	Đ/Đ	S/S
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	0	1
4	1	1	1	1	1	1	1	1
5	1	1	0	1	1	1	1	1
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	0	1
9	1	1	0	1	1	1	1	1
10	1	1	1	0	1	1	1	0
Tổng	10	10	8	9	10	10	8	9
Phần trăm	100	100	80	90	100	100	80	90

4.3 Hướng phát triển

Trong tương lai, tác giả dự định sẽ tiến hành thử nghiệm cho trường hợp người tập mẫu và người tập thật khác nhau. Trường hợp này sẽ khó khăn hơn vì vóc dáng giữa người tập mẫu và người tập thật vẫn có những sự khác nhau nhất định, mặc dù các đặc trưng đã được rút trích sao cho bất biến với vóc dáng nhất.

Thứ hai, tác giả sẽ tiến hành các thử nghiệm trên các bộ bài tập khác, như các bài tập tay phức tạp hơn, bài tập ở chân... nhằm đánh giá khả năng làm việc của hệ thống với các dạng bài tập khác nhau.

Bên cạnh đó, tác giả cũng sẽ thử tạo ra tính năng xác định thời điểm lỗi của động tác dựa trên phép tính độ lỗi theo thời gian sử dụng DTW cục bộ. Và từ đó đề xuất cách chỉnh sửa động tác. Đây là một tính năng có ích vì nó giúp bệnh nhân xác định vị trí thực hiện sai, và dựa trên hướng dẫn có thể sửa lại ngay lần thực hiện sau.

TÀI LIỆU THAM KHẢO

- [1] “"Khoa vật lý trị liệu",” Bệnh viện đại học Y Dược, 2013. [Trực tuyến]. Available: <http://www.bvdaihocoso2.com.vn/gioi-thieu/khoa-lam-sang/khoa-vat-ly-tri-lieu>.
- [2] Nguyễn Thị Thanh Bình, Lê Quang Thanh, “Vận động trị liệu,” 2010.
- [3] Wenbing Zhao, Roanna Lun, Deborah D Espy, Ann Reinthal, “Realtime Motion Assessment For Rehabilitation Exercises: Integration Of Kinematic Modeling With Fuzzy Inference,” *Journal of Artificial Intelligence and Soft Computing Research*, tập 4, số 4, pp. 267-285, 2014.
- [4] Microsoft, “Kinect for Windows SDK,” [Trực tuyến]. Available: <https://msdn.microsoft.com/en-us/library/dn799271.aspx>.
- [5] C.-J. Su, “Personal Rehabilitation Exercise Assistant with Kinect and Dynamic Time Warping,” *International Journal of Information and Education Technology*, tập 3, số 4, pp. 448-454, 2013.
- [6] M. Müller, "Dynamic Time Warping," in *Information Retrieval for Music and Motion*, Springer, 2007, pp. 69-84.
- [7] Antón D, Goñi A, Illarramendi A, “Exercise recognition for Kinect-based telerehabilitation,” *Methods of information in medicine*, tập 54, số 2, pp. 145-155, 2015.