

BỘ GIÁO DỤC  
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC  
VÀ CÔNG NGHỆ VIỆT NAM

**HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ**

---



**Nghiêm Văn Hưng**

**NGHIÊN CỨU MỘT SỐ KỸ THUẬT PHÁT HIỆN VA CHẠM  
TRONG VẬT THỂ BIẾN DẠNG VÀ CÁNH TAY COBOT**

**LUẬN ÁN TIẾN SĨ MÁY TÍNH**

*Hà Nội - 2024*

BỘ GIÁO DỤC  
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC  
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

**Nghiêm Văn Hưng**

**NGHIÊN CỨU MỘT SỐ KỸ THUẬT PHÁT HIỆN VA CHẠM  
TRONG VẬT THỂ BIẾN DẠNG VÀ CÁNҺ TAY COBOT**

**LUẬN ÁN TIẾN SĨ MÁY TÍNH**

**Ngành: Hệ thống thông tin**

**Mã số: 9 48 01 04**

**Xác nhận của Học viện  
Khoa học và Công nghệ**

**Người hướng dẫn  
(Ký, ghi rõ họ tên)**

**PGS.TS. Đặng Văn Đức**

**Hà Nội - 2024**

## LỜI CAM ĐOAN

Tôi xin cam đoan luận án: “**Nghiên cứu một số kỹ thuật phát hiện va chạm trong vật thể biến dạng và cánh tay cobot**” là công trình nghiên cứu của chính mình dưới sự hướng dẫn khoa học của PGS.TS. Đặng Văn Đức. Luận án sử dụng thông tin trích dẫn từ nhiều nguồn tham khảo khác nhau và các thông tin trích dẫn được ghi rõ nguồn gốc. Các kết quả nghiên cứu của tôi được công bố chung với các tác giả khác đã được sự nhất trí của đồng tác giả khi đưa vào luận án. Các số liệu, kết quả được trình bày trong luận án là hoàn toàn trung thực và chưa từng được công bố trong bất kỳ một công trình nào khác ngoài các công trình công bố của tác giả. Luận án được hoàn thành trong thời gian tôi làm nghiên cứu sinh tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.

*Hà Nội, ngày tháng năm 2024*

**Tác giả luận án**  
*(Ký và ghi rõ họ tên)*

**Nghiêm Văn Hưng**

## LỜI CẢM ƠN

Luận án này được nghiên cứu sinh (NCS) thực hiện trong quá trình học tập tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Để hoàn thành luận án tiến sĩ, NCS đã nhận được rất nhiều sự hướng dẫn, giúp đỡ, động viên từ thầy giáo hướng dẫn, đồng nghiệp và các nhà khoa học.

Trước tiên, NCS xin bày tỏ lòng biết ơn sâu sắc tới thầy hướng dẫn khoa học PGS.TS. Đặng Văn Đức đã định hướng, hướng dẫn về mặt chuyên môn và luôn quan tâm, chỉ bảo NCS trong quá trình học tập.

NCS xin trân trọng cảm ơn Ban Lãnh đạo Viện Công nghệ thông tin, Ban Giám đốc Học viện Khoa học và Công nghệ, Phòng Đào tạo, các Phòng Ban chức năng của Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam đã quan tâm, tạo điều kiện cho NCS trong suốt quá trình thực hiện luận án. NCS xin được gửi lời cảm ơn chân thành tới các thầy giáo, cô giáo, nhà khoa học, cán bộ nghiên cứu của Khoa Công nghệ thông tin và Viễn thông, Học viện Khoa học và Công nghệ, Phòng Công nghệ thực tại ảo, Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam đã truyền đạt kiến thức, kỹ năng và tạo điều kiện cho NCS trong quá trình học tập, nghiên cứu và thực hiện luận án.

NCS xin trân trọng cảm ơn Ban Giám hiệu, Lãnh đạo Khoa Công nghệ và An toàn thông tin, Trường Đại học Kỹ thuật - Hậu cần Công an nhân dân đã tạo điều kiện cho NCS học tập, luôn quan tâm, động viên và giúp đỡ NCS về cả chuyên môn và công tác.

NCS xin chân thành cảm ơn PGS.TS Nguyễn Thanh Phương, Trường Đại học L'Aquila, Cộng hòa Italia đã động viên, giúp đỡ NCS trong quá trình học tập.

Cuối cùng, NCS xin gửi lời cảm ơn đến gia đình, người thân, bạn bè và đồng nghiệp đã luôn luôn chia sẻ, ủng hộ NCS trong quá trình học tập, là nguồn động lực phấn đấu, khuyến khích và động viên để NCS có thể hoàn thành luận án này.

*Hà Nội, ngày tháng năm 2024*

**Tác giả luận án**  
*(Ký và ghi rõ họ tên)*

**Nghiêm Văn Hưng**

## MỤC LỤC

LỜI CAM ĐOAN .....	i
LỜI CẢM ƠN .....	ii
MỤC LỤC.....	iii
DANH MỤC CÁC KÝ HIỆU .....	v
DANH MỤC CÁC CHỮ VIẾT TẮT .....	vi
DANH MỤC CÁC BẢNG.....	viii
DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ.....	ix
MỞ ĐẦU .....	1
CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN PHÁT HIỆN VA CHẠM VÀ MỘT SỐ KIẾN THỨC NỀN TẢNG.....	11
1.1. Giới thiệu .....	11
1.2. Kiến thức nền tảng.....	13
1.2.1. Thực tại ảo .....	13
1.2.2. Cấu tạo mô hình 3D.....	18
1.2.3. Mô hình vật thể biến dạng .....	21
1.2.4. Quy tắc dấu Descartes và Định lý Vincent.....	22
1.2.5. Cobot và cấu tạo của cánh tay cobot .....	24
1.3. Tổng quan các nghiên cứu liên quan .....	29
1.3.1. Phát hiện va chạm của các mô hình vật thể 3D.....	29
1.3.2. Phát hiện va chạm của cánh tay cobot.....	30
1.4. Một số hạn chế của các phương pháp truyền thống .....	32
1.4.1. Hạn chế liên quan đến phát hiện va chạm của mô hình vật thể rắn .....	32
1.4.2. Hạn chế liên quan đến phát hiện va chạm của mô hình chất liệu vải.....	36
1.4.3. Hạn chế liên quan đến phát hiện va chạm của cánh tay cobot .....	39
1.5. Chỉ số đánh giá .....	42
1.5.1. Độ đo đánh giá hiệu suất phát hiện va chạm.....	42
1.5.2. Tiêu chí đánh giá các loại khối bao .....	42
1.6. Kết luận Chương 1 .....	43

CHƯƠNG 2. PHƯƠNG PHÁP PHÁT HIỆN VA CHẠM TRONG VẬT THỂ BIẾN DẠNG .....	45
2.1. Đề xuất phương pháp nâng cao hiệu quả phát hiện va chạm của mô hình chất liệu vải.....	45
2.1.1. Phương pháp đề xuất .....	45
2.1.2. Kết quả và thảo luận .....	53
2.2. Đề xuất phương pháp nâng cao hiệu quả phát hiện va chạm của mô hình vật thể rắn.....	58
2.2.1. Phương pháp đề xuất .....	58
2.2.2. Kết quả và thảo luận .....	65
2.3. Kết luận Chương 2.....	68
CHƯƠNG 3. PHƯƠNG PHÁP PHÁT HIỆN VA CHẠM CỦA CÁN TAY COBOT 6 BẬC TỰ DO.....	70
3.1. Phương pháp đề xuất .....	70
3.1.1. Phương pháp phát hiện va chạm.....	70
3.1.2. Bộ dữ liệu thử nghiệm .....	76
3.2. Kết quả và thảo luận .....	78
3.2.1. Kết quả với bộ dữ liệu ngưỡng 1,6 [A] .....	78
3.2.2. Kết quả với bộ dữ liệu ngưỡng 2,0 [A] .....	81
3.2.3. Thảo luận .....	84
3.3. Kết luận Chương 3 .....	85
KẾT LUẬN VÀ KIẾN NGHỊ.....	87
DANH MỤC CÔNG TRÌNH CÔNG BỐ LIÊN QUAN ĐẾN LUẬN ÁN ...	90
DANH MỤC TÀI LIỆU THAM KHẢO.....	91

## DANH MỤC CÁC KÝ HIỆU

STT	Ký hiệu	Ý nghĩa
1.	$v_1$	Đỉnh số 1 của một mặt tam giác
2.	$e_2^1$	Cạnh nối đỉnh số 1 và đỉnh số 2 của tam giác
3.	$\vec{n}_t$	Vectơ pháp tuyến tại thời điểm $t$
4.	$\vec{v}_t$	Vectơ vận tốc tại thời điểm $t$
5.	$d(t)$	Hàm tính khoảng cách tại thời điểm $t$
6.	$q_i(t)$	Vị trí của khớp $i$ trên cánh tay cobot
7.	$M(q)$	Ma trận quán tính
8.	$g(q)$	Lực hấp dẫn
9.	$TP$	True Positives: Số lượng sự kiện va chạm được phát hiện đúng
10.	$TN$	True Negatives: Số lượng sự kiện không phải là va chạm không được phát hiện
11.	$FP$	False Positives: Số lượng sự kiện không phải là va chạm bị phát hiện nhầm là va chạm
12.	$FN$	False Negatives: Số lượng sự kiện va chạm bị bỏ sót (không được phát hiện)

## DANH MỤC CÁC CHỮ VIẾT TẮT

STT	Viết tắt	Viết đầy đủ tiếng Anh	Viết đầy đủ tiếng Việt
1.	3D	Three-Dimensional	Ba chiều
2.	AABB	Axis-Aligned Bounding Box	Khối bao theo cạnh song song với các trục tọa độ
3.	AR	Augmented Reality	Thực tại tăng cường
4.	BV	Bounding Volume	Vùng bao
5.	BVH	Bounding Volume Hierarchies	Cấu trúc phân hệ vùng bao
6.	BVTT	Bounding Volume Traversal Tree	Cây phân cấp thứ tự duyệt hệ bao
7.	CD	Collision Detection	Phát hiện va chạm
8.	CMCN	Industrial Revolution	Cách mạng công nghiệp
9.	CNN	Convolutional Neural Network	Mạng nơron tích chập
10.	COBOT	Collaborative Robot	Robot cộng tác
11.	CPU	Central Processing Unit	Bộ xử lý trung tâm
12.	CSDL	Database	Cơ sở dữ liệu
13.	CURA	Cooperative Universal Robotic Assistant	Cánh tay cobot của hãng Intema (Ba Lan) sản xuất
14.	DL	Deep Learning	Học sâu
15.	DNF	Deforming Non-penetration Filters	Thuật toán lọc biến dạng không xâm nhập
16.	DoF	Degrees of Freedom	Bậc tự do
17.	EE	Edge-Edge	Phép kiểm tra cạnh - cạnh
18.	GAMMA	Geometric Algorithms for Modeling, Motion, and Animation	Bộ dữ liệu của nhóm nghiên cứu GAMMA thuộc Trường Đại học Maryland (Hoa Kỳ)



19.	GPU	Graphics Processing Unit	Bộ xử lý đồ họa
20.	HRI	Human-Robot Interaction	Tương tác người - máy
21.	ICCD	Interactive Continuous Collision Detection	Thuật toán phát hiện va chạm liên tục
22.	ISO	International Organization for Standardization	Tổ chức tiêu chuẩn hoá quốc tế
23.	k-DOP	k-Discrete Oriented Polytopes	Khối bao đa diện rời rạc có hướng
24.	LP	Linear Programming	Quy hoạch tuyến tính
25.	LSTM	Long Short-Term Memory	Bộ nhớ dài ngắn hạn
26.	M-FNN	Multi-Layer Feed-Forward Neural Network	Mạng nơron truyền thẳng đa lớp
27.	ML	Machine Learning	Học máy
28.	NCS	Ph.D. Student	Nghiên cứu sinh
29.	OC-SVM	One-Class Support Vector Machine	Máy vector hỗ trợ một lớp
30.	OOBB	Object-Oriented Bounding Box	Khối bao theo hướng của đối tượng
31.	SAT	Separating Axis Theorem	Thuật toán dựa trên Định lý về trục phân tách
32.	SBB	Sphere Bounding Box	khối bao dạng hình cầu
33.	SCD	Sturm-based Collision Detection	Thuật toán phát hiện va chạm dựa trên Định lý Sturm
34.	SVR	Support Vector Regression	Thuật toán hồi quy vector hỗ trợ
35.	VCD	Vincent-based Collision Detection	Thuật toán phát hiện va chạm dựa trên Định lý Vincent
36.	VF	Vertex-Face	Phép kiểm tra đỉnh - mặt
37.	VR	Virtual Reality	Thực tại ảo

## DANH MỤC CÁC BẢNG

	<i>Trang</i>
<i>Bảng 1.1.</i> Bảng mô tả sự tồn tại của nghiệm .....	23
<i>Bảng 1.2.</i> Bảng tham số D-H của cobot CURA6 .....	27
<i>Bảng 1.3.</i> Bảng so sánh các loại khối bao.....	43
<i>Bảng 2.1.</i> Bảng mô tả các bộ dữ liệu .....	54
<i>Bảng 2.2.</i> Bảng thời gian phát hiện va chạm trung bình của các thuật toán... 55	55
<i>Bảng 2.3.</i> Bảng so sánh tỷ lệ thời gian phát hiện va chạm của các thuật toán trong thử nghiệm thứ nhất.....	65
<i>Bảng 2.4.</i> Bảng so sánh tỷ lệ thời gian phát hiện va chạm của các thuật toán trong thử nghiệm thứ hai.....	66
<i>Bảng 3.1.</i> Ma trận nhầm lẫn, Tốc độ max V 10% - Ngưỡng 1,6 [A] .....	79
<i>Bảng 3.2.</i> Ma trận nhầm lẫn, Tốc độ max V 20% - Ngưỡng 1,6 [A] .....	79
<i>Bảng 3.3.</i> Ma trận nhầm lẫn, Tốc độ max V 30% - Ngưỡng 1,6 [A] .....	79
<i>Bảng 3.4.</i> Ma trận nhầm lẫn, Tốc độ max V 40% - Ngưỡng 1,6 [A] .....	80
<i>Bảng 3.5.</i> Ma trận nhầm lẫn, Tốc độ max V 50% - Ngưỡng 1,6 [A] .....	80
<i>Bảng 3.6.</i> Ma trận nhầm lẫn, Tốc độ max V 60% - Ngưỡng 1,6 [A] .....	81
<i>Bảng 3.7.</i> Ma trận nhầm lẫn, Tốc độ max V 10% - Ngưỡng 2,0 [A] .....	81
<i>Bảng 3.8.</i> Ma trận nhầm lẫn, Tốc độ max V 20% - Ngưỡng 2,0 [A] .....	82
<i>Bảng 3.9.</i> Ma trận nhầm lẫn, Tốc độ max V 30% - Ngưỡng 2,0 [A] .....	82
<i>Bảng 3.10.</i> Ma trận nhầm lẫn, Tốc độ max V 40% - Ngưỡng 2,0 [A] .....	83
<i>Bảng 3.11.</i> Ma trận nhầm lẫn, Tốc độ max V 50% - Ngưỡng 2,0 [A] .....	83
<i>Bảng 3.12.</i> Ma trận nhầm lẫn, Tốc độ max V 60% - Ngưỡng 2,0 [A] .....	83
<i>Bảng 3.13.</i> Đánh giá kết quả và so sánh với công bố của Czubenko [25].....	84

## DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

*Trang*

<i>Hình 1.1.</i> Hệ thống thực tại ảo của quân đội Hoa Kỳ phục vụ công tác huấn luyện tác chiến.....	13
<i>Hình 1.2.</i> Quy trình mô hình hóa thực tại ảo .....	15
<i>Hình 1.3.</i> Mô tả hình dạng đối tượng: (a) sử dụng lưới tam giác, (b) sử dụng bề mặt tham số.....	16
<i>Hình 1.4.</i> (a) Hình ảnh cobot CURA6 tại phòng thí nghiệm của Intema; (b) Mô hình cobot CURA6 .....	26
<i>Hình 1.5.</i> Một số loại khối bao được sử dụng phổ biến trong các kỹ thuật phát hiện va chạm.....	30
<i>Hình 1.6.</i> Phân hệ vùng bao sử dụng trong các kỹ thuật phát hiện va chạm..	34
<i>Hình 1.7.</i> Minh họa ý tưởng xác định mặt phẳng tách hai vật thể.....	35
<i>Hình 2.1.</i> Mô phỏng một lá cờ đang tung bay trong gió.....	45
<i>Hình 2.2.</i> Sơ đồ tổng quát thuật toán đề xuất.....	46
<i>Hình 2.3.</i> Các phép kiểm tra sơ cấp đối với lưới tam giác.....	49
<i>Hình 2.4.</i> Các bộ dữ liệu trong thư viện mở GAMMA: (a) Bộ dữ liệu Princess, (b) Bộ dữ liệu Flamenco, (c) Bộ dữ liệu Cloth-ball .....	54
<i>Hình 2.5.</i> Kết quả thực thi các thuật toán.....	56
<i>Hình 2.6.</i> Phát hiện va chạm tại các frames trong bộ dữ liệu Cloth-ball.....	56
<i>Hình 2.7.</i> Hiệu suất lọc của các thuật toán.....	57
<i>Hình 2.8.</i> So sánh chi phí thực thi các thuật toán .....	57
<i>Hình 2.9.</i> Thử nghiệm phát hiện va chạm của các mô hình thử .....	66
<i>Hình 2.10.</i> Đánh giá hiệu suất của thuật toán đề xuất.....	67
<i>Hình 3.1.</i> Vị trí của bài toán phát hiện va chạm trong hệ xử lý va chạm [29]	71
<i>Hình 3.2.</i> Sơ đồ phương pháp SVR đề xuất.....	73
<i>Hình 3.3.</i> Đầu ra của SVR và chỉ số va chạm tương ứng .....	75
<i>Hình 3.4.</i> Ví dụ về dữ liệu [25] .....	77

## MỞ ĐẦU

### 1. Tính cấp thiết của đề tài nghiên cứu

Những tiến bộ đột phá của khoa học kỹ thuật tiên tiến đã tạo động lực thúc đẩy nhiều lĩnh vực nghiên cứu phát triển, trong đó tiêu biểu là công nghệ thực tại ảo (*Virtual Reality - VR*) và công nghệ robot (*Robotics*). Đây chính là hai trong số những công nghệ chủ chốt được ưu tiên tập trung nghiên cứu, phát triển và ứng dụng. VR đã phát triển khiến việc định nghĩa VR theo các thiết bị cụ thể có thể không còn được ưa chuộng trong một hoặc hai năm là điều không mong muốn. Trong luận án này, NCS quan tâm đến các nguyên tắc cơ bản ít nhạy cảm hơn với các công nghệ cụ thể và do đó tồn tại trước thử thách của thời gian. Cần xem xét VR thực sự có ý nghĩa gì, theo cách nắm bắt được những khía cạnh quan trọng nhất bất chấp công nghệ đang thay đổi nhanh chóng. Khái niệm này cũng phải đủ tổng quát để bao gồm những gì VR được coi là hiện đại và những gì chúng ta hình dung cho tương lai của nó.

VR được định nghĩa là “công nghệ tạo ra hành vi có chủ đích trong sinh vật bằng cách sử dụng kích thích giác quan nhân tạo, trong khi sinh vật đó có rất ít hoặc không có nhận thức về sự can thiệp” [1]. Bốn thành phần chính xuất hiện trong định nghĩa: (1) Hành vi có mục tiêu: Sinh vật đang có một “trải nghiệm” do người sáng tạo thiết kế. Ví dụ bao gồm bay, đi bộ, khám phá, xem phim và giao lưu với các sinh vật khác. (2) Sinh vật: Đây có thể là con người khác hoặc thậm chí là một dạng sống khác. (3) Kích thích giác quan nhân tạo: Thông qua sức mạnh của kỹ thuật, một hoặc nhiều giác quan của sinh vật trở nên đồng lựa chọn, ít nhất một phần, và các đầu vào thông thường của chúng được thay thế hoặc tăng cường bằng kích thích nhân tạo. (4) Nhận thức: Trong khi trải nghiệm, sinh vật dường như không nhận thức được sự can thiệp, từ đó bị “đánh lừa” cảm thấy mình đang hiện diện trong một thế giới ảo. Sự không nhận thức này dẫn đến cảm giác hiện diện trong một thế giới ảo, môi trường ảo. Nó được chấp nhận như một lẽ tự nhiên.

Thế hệ hệ thống VR hiện tại được tạo ra nhờ những tiến bộ trong công nghệ hiển thị, cảm biến và điện toán của ngành công nghiệp điện thoại thông minh [2-8]. Từ Palmer Luckey's 2012 Oculus Rift thiết kế để chế tạo hộp xem cho điện thoại thông minh, thế giới đã nhanh chóng thay đổi khi tai nghe VR được sản xuất hàng loạt và đeo trên đầu hàng triệu người. Xu hướng này về nhiều mặt tương tự như cuộc cách mạng máy tính gia đình và trình duyệt web; khi nhiều người có quyền truy cập vào công nghệ hơn, số việc họ làm với nó cũng sẽ mở rộng đáng kể [9-15]. Ở đây chỉ có thể mô tả những trải nghiệm bằng lời nói và hình ảnh, còn lâu mới có thể cảm nhận được sự trân trọng khi tự mình trải nghiệm chúng. Phương tiện in ấn này (một cuốn sách) thật không đủ để truyền tải đầy đủ phương tiện VR [16, 17].

Tại Việt Nam, các nghiên cứu về cobot chủ yếu về các kỹ thuật mô hình hóa cánh tay cobot như công bố của nhóm tác giả Duyên và cộng sự [37]. Trong đó, nhóm tác giả phân tích động học ngược cobot UR và sử dụng bộ điều khiển PID thực hiện mô phỏng mô hình cánh tay cobot trên phần mềm Matlab/Simulink. Nhóm tác giả Lâm và cộng sự [38] thiết kế cánh tay cobot có đặc tính gọn nhẹ, dễ di chuyển. Nhóm tác giả Thương và cộng sự [24] khảo sát tương đối toàn diện về nghiên cứu tính an toàn của cobot.

Các kết quả nghiên cứu về cobot tại Việt Nam chủ yếu tập trung vào các kỹ thuật mô hình hóa, thiết kế phần cứng, xây dựng phần mềm hoặc phát triển các kỹ thuật tránh va chạm. Các nghiên cứu về phát hiện va chạm của bộ phận cánh tay cobot còn chưa phổ biến do hạn chế về thiết bị, công nghệ và bộ dữ liệu phục vụ nghiên cứu.

VR được định nghĩa là “công nghệ tạo ra hành vi có chủ đích trong sinh vật bằng cách sử dụng kích thích giác quan nhân tạo, trong khi sinh vật đó có rất ít hoặc không có nhận thức về sự can thiệp”. Bốn thành phần chính xuất hiện trong định nghĩa:

- Hành vi có mục tiêu: Sinh vật đang có một “trải nghiệm” do người sáng tạo thiết kế. Ví dụ bao gồm bay, đi bộ, khám phá, xem phim và giao lưu với các sinh vật khác.

- Sinh vật: Đây có thể là con người khác hoặc thậm chí là một dạng sống khác.

- Kích thích giác quan nhân tạo: Thông qua sức mạnh của kỹ thuật, một hoặc nhiều giác quan của sinh vật trở nên đồng lựa chọn, ít nhất một phần, và các đầu vào thông thường của chúng được thay thế hoặc tăng cường bằng kích thích nhân tạo.

- Nhận thức: Trong khi trải nghiệm, sinh vật dường như không nhận thức được sự can thiệp, từ đó bị “đánh lừa” cảm thấy mình đang hiện diện trong một thế giới ảo. Sự không nhận thức này dẫn đến cảm giác hiện diện trong một thế giới ảo, môi trường ảo. Nó được chấp nhận như một lẽ tự nhiên.

Thế hệ hệ thống VR hiện tại được tạo ra nhờ những tiến bộ trong công nghệ hiển thị, cảm biến và điện toán của ngành công nghiệp điện thoại thông minh [2-8]. Từ Palmer Luckey's 2012 Oculus Rift thiết kế để chế tạo hộp xem cho điện thoại thông minh, thế giới đã nhanh chóng thay đổi khi tai nghe VR được sản xuất hàng loạt và đeo trên đầu hàng triệu người. Xu hướng này về nhiều mặt tương tự như cuộc cách mạng máy tính gia đình và trình duyệt web; khi nhiều người có quyền truy cập vào công nghệ hơn, số việc họ làm với nó cũng sẽ mở rộng đáng kể. Cần cung cấp cái nhìn tổng quan nhanh về những gì mọi người đang làm với hệ thống VR và cung cấp điểm khởi đầu để tìm kiếm những trải nghiệm tương tự trên Internet [9]. Ở đây, chúng ta chỉ có thể mô tả những trải nghiệm bằng lời nói và hình ảnh, còn lâu mới có thể cảm nhận được sự trân trọng khi tự mình trải nghiệm chúng.

Thuật ngữ thực tế ảo bắt nguồn từ chuyên gia người Đức Immanuel Kant, mặc dù việc sử dụng nó không liên quan đến công nghệ. Kant đưa ra thuật ngữ này để chỉ “thực tại” tồn tại trong tâm trí ai đó, khác biệt với thế giới vật chất

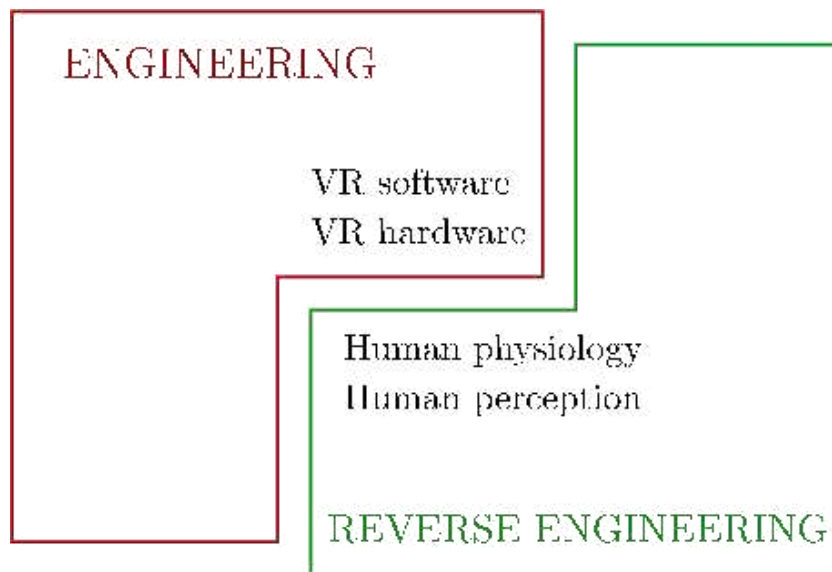
bên ngoài, cũng là một thực tại. Việc sử dụng hiện đại thuật ngữ VR đã được phổ biến rộng rãi bởi Jaron Lanier vào năm 1985. Thật không may, bản thân cái tên thực tế ảo dường như đã tự mâu thuẫn, đó là một vấn đề triết học được khắc phục bằng cách đề xuất thuật ngữ thay thế ảo. Mặc dù thừa nhận vấn đề này, tuy nhiên NCS sẽ tiếp tục sử dụng thuật ngữ thực tế ảo. Tuy nhiên, sự khác biệt sau đây sẽ trở nên quan trọng: Thế giới thực đề cập đến thế giới vật lý chứa đựng người dùng tại thời điểm trải nghiệm và thế giới ảo đề cập đến thế giới được nhận thức như một phần của trải nghiệm VR được nhắm mục tiêu.

Mặc dù thuật ngữ VR đã khá bao quát nhưng hiện nay một số thuật ngữ cạnh tranh liên quan đến VR vẫn đang được sử dụng phổ biến. Thuật ngữ môi trường ảo có trước khi VR được sử dụng rộng rãi và được hầu hết các nhà nghiên cứu ở trường đại học ưa thích. Nó thường được coi là đồng nghĩa với VR; tuy nhiên, NCS nhấn mạnh trong cuốn sách này rằng môi trường được cảm nhận có thể là một thế giới “thực” được ghi lại bằng nhiếp ảnh cũng như một thế giới hoàn toàn tổng hợp. Do đó, môi trường nhận thức được trình bày trong VR không cần phải có vẻ “ảo”. Thực tế tăng cường (AR) đề cập đến các hệ thống trong đó hầu hết các kích thích thị giác được truyền trực tiếp qua kính hoặc máy ảnh đến mắt và một số cấu trúc bổ sung, chẳng hạn như văn bản và đồ họa, dường như được đặt chồng lên thế giới của người dùng. Thuật ngữ thực tế hỗn hợp (MR) đôi khi được dùng để chỉ toàn bộ quang phổ bao gồm VR, AR và thực tế thông thường [25]. Mọi người đã nhận ra rằng những thuật ngữ và sự phân biệt hàng thập kỷ này đã bị xói mòn trong những năm gần đây, đặc biệt là khi các công nghệ thống nhất đã phát triển nhanh chóng. Do đó, gần đây đã có những nỗ lực nhằm nhanh chóng hợp nhất chúng lại với nhau dưới tiêu đề XR., X Thực tế, Thực tế ảo/AR, Thực tế tăng cường/thực tế ảo, VR/AR/MR và như thế.

Khái niệm liên quan đến Telepresence đề cập đến các hệ thống cho phép người dùng cảm thấy như họ đang ở một nơi nào khác trong thế giới thực; nếu họ có thể điều khiển bất cứ thứ gì, chẳng hạn như máy bay không người lái,

thì điều khiển từ xa là một thuật ngữ thích hợp. Vì mục đích của NCS, môi trường ảo, AR, thực tế hỗn hợp, hiện diện từ xa và vận hành từ xa đều sẽ được coi là những ví dụ hoàn hảo về VR.

Ý tưởng quan trọng nhất của VR là nhận thức của người dùng về thực tế đã được thay đổi thông qua kỹ thuật, thay vì liệu môi trường mà họ tin rằng họ đang ở có vẻ “thực” hơn hay “ảo” hơn. Một ảo ảnh nhận thức đã được thiết kế. Do đó, một thuật ngữ hợp lý khác cho lĩnh vực này, đặc biệt nếu được coi là một môn học thuật, có thể là kỹ thuật nhận thức, các phương pháp kỹ thuật đang được sử dụng để thiết kế, phát triển và mang lại ảo giác nhận thức cho người dùng. Hình 1.2 minh họa các thành phần của kỹ thuật nhận thức, điều này cũng thúc đẩy các chủ đề nghiên cứu, là sự kết hợp giữa kỹ thuật và tâm lý học và nhận thức của con người.



Hình 1.1. Các thành phần của kỹ thuật nhận thức VR

Năm 1965, kỹ sư Ivan Sutherland đã thiết kế thiết bị quan sát được gọi là Ultimate Display. Thiết bị hiển thị này được coi là chiếc kính thực tại ảo đầu tiên trên thế giới. Sang giai đoạn những năm 1990, những chiếc kính thực tại ảo đã dần phổ biến trong các trò chơi mô phỏng. Vào năm 1993, Sega giới thiệu đến công chúng chiếc kính thực tại ảo được dùng cho bảng điều khiển Sega Genesis. Tuy nhiên, giao diện điều khiển còn chưa thân thiện, khó sử dụng.





Hình 1.2. Chiếc kính thực tế ảo Oculus Rift

Chiếc kính thực tại ảo Oculus Rift được hình thành từ chiến dịch ra mắt năm 2012. Đến năm 2015, Oculus đã mua lại công ty Surreal Vision hợp tác với Samsung để cùng phát triển thiết bị kính thực tại ảo Gear VR của Samsung. Oculus đã tạo ra trải nghiệm nhập vai và mô phỏng thực tại ảo.

Ngoài việc dạy về sự đồng cảm, góc nhìn của người thứ nhất có thể cách mạng hóa nhiều lĩnh vực giáo dục. Trong kỹ thuật, toán học và khoa học, VR mang đến cơ hội hình dung các mối quan hệ hình học trong các khái niệm hoặc dữ liệu khó diễn giải. Hơn nữa, VR phù hợp một cách tự nhiên cho việc đào tạo thực tế vì các kỹ năng được phát triển trong môi trường ảo thực tế có thể chuyển sang môi trường thực một cách tự nhiên. Động lực sẽ đặc biệt cao nếu môi trường thực tế cung cấp tốn kém hoặc gây ra những rủi ro về sức khỏe. Một trong những ví dụ sớm nhất và phổ biến nhất về đào tạo trong VR là *mô phỏng chuyển bay* (Hình 1.4). Các ví dụ khác bao gồm chữa cháy, an toàn nhà máy điện hạt nhân, tìm kiếm và cứu nạn, hoạt động quân sự và các thủ tục y tế.

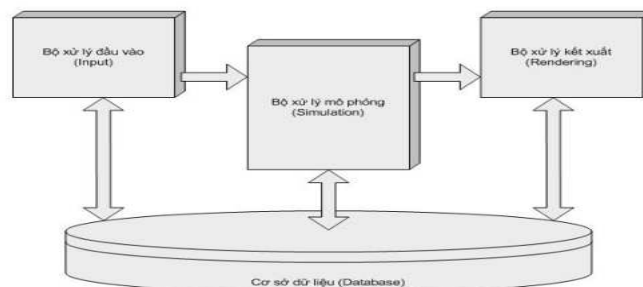


Hình 1.3. Hệ thống mô phỏng huấn luyện phi công lái máy bay

Ngoài những ứng dụng phổ biến này của VR, có lẽ cơ hội lớn nhất cho giáo dục VR nằm ở các ngành nhân văn, bao gồm lịch sử, nhân chủng học và tiếp thu ngoại ngữ. Hãy xem xét sự khác biệt giữa việc đọc một cuốn sách về thời Victoria ở Anh và việc có thể đi lang thang trên đường phố London thế kỷ 19, trong một mô phỏng đã được các nhà sử học xây dựng một cách tỉ mỉ. Chúng ta thậm chí có thể đến thăm một thành phố cổ được xây dựng lại từ đồng đồ nát (Hình 1.5). Có những khả năng hấp dẫn khi tham quan các bảo tàng vật lý thông qua giao diện VR hoặc quét và trưng bày các hiện vật trực tiếp trong bảo tàng ảo. Những ví dụ này thuộc tiêu đề *di sản kỹ thuật số*.

## 2. Mục tiêu nghiên cứu

Mô hình kiến trúc hệ thống thực tại ảo gồm các thành phần: Bộ xử lý đầu vào, bộ xử lý mô phỏng, bộ xử lý kết xuất và cơ sở dữ liệu như hình 1.6.

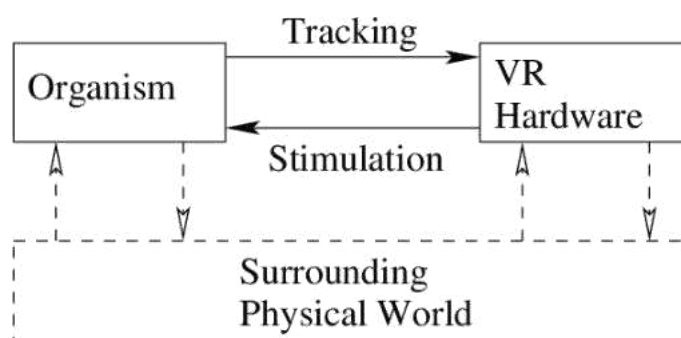


Hình 1.4. Mô hình kiến trúc hệ thống thực tại ảo

Đối với chương trình mô phỏng VR phục vụ giảng dạy thì các thiết bị cần có trong lớp học gồm: Máy tính, máy chiếu, hệ thống âm thanh, kính thực tại ảo và các thiết bị kết nối.

### 3. Đối tượng và phạm vi nghiên cứu

Bước đầu tiên để hiểu cách thức hoạt động của VR là xem xét những gì tạo nên toàn bộ *hệ thống VR*. Thật hấp dẫn khi nghĩ nó chỉ đơn thuần là các thành phần phần cứng, chẳng hạn như máy tính, tai nghe và bộ điều khiển. Điều này sẽ không đầy đủ một cách đáng tiếc. Điều quan trọng không kém là phải tính đến sinh vật, sinh vật này trong chương này sẽ chỉ đề cập đến *người dùng* là con người. Phần cứng tạo ra các kích thích lẫn át giác quan của người dùng.



Hình 1.5. Phần cứng VR

Trong màn hình gắn trên đầu từ Phần 1.3 (Hình 1.30 (b)), hãy nhớ lại rằng cần phải theo dõi để điều chỉnh kích thích dựa trên chuyển động của con người. Phần cứng VR thực hiện điều này bằng cách sử dụng các cảm biến riêng, từ đó *theo dõi* chuyển động của người dùng. Theo dõi đầu là quan trọng nhất, nhưng theo dõi cũng có thể bao gồm việc nhấn nút, chuyển động của bộ điều khiển, chuyển động của mắt hoặc chuyển động của bất kỳ bộ phận cơ thể nào khác. Cuối cùng, điều quan trọng là phải coi thế giới vật chất xung quanh là một phần của hệ thống VR. Bất chấp sự kích thích do phần cứng VR cung cấp, người dùng sẽ luôn có các giác quan khác phản ứng với các kích thích từ thế giới thực. Cô ấy cũng có khả năng thay đổi môi trường của mình thông qua chuyển động cơ thể. Phần cứng VR cũng có thể theo dõi các đối tượng không phải người dùng, đặc biệt nếu tương tác với chúng là một phần của trải nghiệm

VR. Thông qua giao diện robot, phần cứng VR cũng có thể thay đổi thế giới thực. Một ví dụ là robot điều khiển từ xa thông qua giao diện VR.

#### 4. Nội dung nghiên cứu

Thông tin được trích xuất từ thế giới vật chất như thế nào? Rõ ràng điều này rất quan trọng đối với hệ thống VR. Trong kỹ thuật, một *bộ chuyển đổi* đề cập đến một thiết bị chuyển đổi năng lượng từ dạng này sang dạng khác. Một *cảm biến* là một bộ chuyển đổi đặc biệt có chức năng chuyển đổi năng lượng mà nó nhận được thành tín hiệu cho mạch điện. Đây có thể là tín hiệu analog hoặc tín hiệu số, tùy thuộc vào loại mạch. Một cảm biến thường có một *bộ phận tiếp nhận* thu thập năng lượng để chuyển đổi. Các sinh vật hoạt động theo cách tương tự. "Cảm biến" được gọi là *cơ quan cảm giác*, với những ví dụ phổ biến là mắt và tai. Bởi vì các "mạch" của chúng ta được hình thành từ các tế bào thần kinh liên kết với nhau nên các cơ quan cảm giác sẽ chuyển đổi năng lượng thành *các xung thần kinh*. Khi ta đọc cuốn sách này, hãy ghi nhớ những điểm tương đồng giữa các cảm biến được thiết kế và các cơ quan cảm giác tự nhiên. Họ đang đo lường những thứ giống nhau và thậm chí đôi khi còn hoạt động theo cách tương tự. Điều này không có gì đáng ngạc nhiên vì chúng ta và các thiết bị được thiết kế của chúng ta có chung một thế giới vật chất: Các định luật vật lý và hóa học vẫn giữ nguyên.

Khi người dùng di chuyển trong thế giới vật chất, các cơ quan cảm giác của anh ta cũng di chuyển theo. Hơn nữa, một số cơ quan cảm giác chuyển động tương đối so với bộ xương của cơ thể, chẳng hạn như mắt chúng ta xoay trong hốc mắt. Mỗi cơ quan cảm giác có một *không gian cấu hình*, tương ứng với tất cả các cách có thể được chuyển đổi hoặc cấu hình. Khía cạnh quan trọng nhất của điều này là số *bậc tự do* hoặc *DOF* của cơ quan cảm giác. Chương 3 sẽ đề cập kỹ lưỡng vấn đề này, nhưng bây giờ hãy lưu ý rằng một vật cứng chuyển động trong không gian thông thường có sáu DOF. Ba DOF tương ứng với vị trí thay đổi của nó trong không gian: 1) chuyển động từ bên này sang bên kia, 2) chuyển động thẳng đứng và 3) chuyển động gập hơn. Ba DOF còn lại tương

ứng với các cách có thể xoay đối tượng; nói cách khác, cần có chính xác ba tham số độc lập để xác định cách định hướng đối tượng. Chúng được gọi là ngáp, cao độ và lặn và được đề cập trong Phần [3.2](#) .

## **5. Bố cục của luận án**

Chương 1: “Tổng quan về bài toán phát hiện va chạm và một số kiến thức nền tảng”.

Chương 2: “Phương pháp phát hiện va chạm trong vật thể biến dạng” trình bày giải pháp và đề xuất thuật toán phát hiện va chạm của mô hình chất liệu vải dựa trên phương pháp lọc hai pha (Collision Detection Algorithm Based on Two-Phase Filtering - AB2PF); đề xuất thuật toán phát hiện va chạm của mô hình vật thể rắn dựa trên phương pháp khử Fourier-Motzkin (Collision Detection Algorithm Based on Fourier-Motzkin Elimination - ABFME).

Chương 3: “Phương pháp phát hiện va chạm của cánh tay cobot 6 bậc tự do” trình bày phương pháp học có giám sát là SVR để phát hiện va chạm của cánh tay cobot CURA6 trên cơ sở các phép đo dòng điện cùng với mô hình động lực học của cobot.

Kết quả nghiên cứu của luận án được công bố tại 02 bài báo trên tạp chí và 04 báo cáo được liệt kê trong phần “Danh mục công trình công bố liên quan đến luận án” ở cuối của luận án này.

## CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN PHÁT HIỆN VA CHẠM VÀ MỘT SỐ KIẾN THỨC NỀN TẢNG

Chương 1 trình bày các cơ sở kiến thức về công nghệ thực tại ảo và robotics, giới thiệu bài toán phát hiện va chạm và tổng quan các công trình nghiên cứu nổi bật trong và ngoài nước có liên quan. Từ đó, NCS đã đưa ra những điểm còn hạn chế của những phương pháp đã được đề cập. Những phân tích này làm cơ sở nghiên cứu cho các chương tiếp theo của luận án.

### 1.1. Giới thiệu

Va chạm là hiện tượng xảy ra khi hai hoặc nhiều vật có sự tiếp xúc với nhau và tác động lên nhau trong một khoảng thời gian ngắn (khoảng từ  $10^{-2}$  đến  $10^{-5}$  giây) [42]. Như vậy va chạm là sự tương tác trong thời gian ngắn giữa hai hoặc nhiều vật thể đồng thời. Các mô hình vật thể trong môi trường ảo có những chuyển động riêng của nó, chuyển động đó có thể va chạm với đối tượng vật thể khác hoặc có thể va chạm với môi trường, chướng ngại vật.

Bài toán phát hiện va chạm được phát biểu như sau: Cho  $n$  đối tượng  $\{O_0, O_1, \dots, O_{n-1}\}$ , phát hiện va chạm là quá trình xác định xem các đối tượng có giao nhau hay không, tức là kiểm tra  $O_i \cap O_j, \forall i \neq j, i = 0, \dots, n-1, j = 0, \dots, n-1$  và  $n \in \mathbb{N}$ . Trong đó,  $O_i = \{p_0^i, p_1^i, \dots, p_k^i\}$  là một tập gồm  $k$  mặt cơ sở (*primitives*) và  $O_j = \{p_0^j, p_1^j, \dots, p_m^j\}$  là một tập gồm  $m$  mặt cơ sở, va chạm của  $O_i$  và  $O_j$  xảy ra khi  $p_a^i \cap p_b^j \neq \emptyset$  với  $a = 0, \dots, k, b = 0, \dots, m$  và  $k, m \in \mathbb{N}$ . Nếu xem xét trường hợp tự va chạm (*Self-Collision*) của  $O_i$  thì cần kiểm tra  $p_a^i \cap p_b^i$  với  $a, b = 0, \dots, k$ .

Các mặt cơ sở được sử dụng trong luận án này là các tam giác. Kết quả trả về của bài toán phát hiện va chạm ở dạng nhị phân Có/Không (*Boolean*), nếu có va chạm xảy ra thì cần tính toán các điểm tiếp xúc và thời điểm tiếp xúc. Việc phát hiện va chạm của các mô hình vật thể biến dạng được đánh giá

là phức tạp hơn việc phát hiện va chạm của các mô hình vật thể rắn vì còn cần phải xem xét đến quá trình biến dạng và vấn đề tự va chạm.

Phát hiện va chạm là một trong những tác vụ cơ sở của các hệ thống mô phỏng VR, đồ họa máy tính, điều khiển robotics,... Trong thời gian gần đây, các nghiên cứu về robotics đã dần chuyển hướng tập trung vào nghiên cứu các đặc điểm của sự hợp tác giữa người - máy trong các ứng dụng sản xuất công nghiệp, nhằm kết hợp độ chính xác và sức mạnh của robot với khả năng nhận thức và tính linh hoạt của con người. Với mục đích này, một thế hệ robot mới là *robot cộng tác* (được gọi là *cobot*) ra đời. Đặc điểm của cobot là làm việc cộng tác cùng con người mà không có rào chắn vật lý, do đó tính an toàn của con người luôn được đặt lên hàng đầu.

Trước đây, cobot chủ yếu được chế tạo để làm việc độc lập trong môi trường khép kín, thay thế con người thực hiện các hoạt động lặp đi lặp lại, nguy hiểm hoặc đòi hỏi độ chính xác cao. Trong bối cảnh như vậy, cobot hoạt động tách biệt khỏi con người bằng các hàng rào an toàn. Tuy nhiên, mặt hạn chế của cobot là thiếu tính linh hoạt và khả năng thích ứng giống như con người. Do đó trong thời gian gần đây, các nghiên cứu về cobot đã dần chuyển hướng tập trung vào nghiên cứu các đặc điểm của sự hợp tác giữa con người và cobot trong các ứng dụng sản xuất công nghiệp, nhằm kết hợp độ chính xác và sức mạnh của cobot với khả năng nhận thức và tính linh hoạt của con người. Trong môi trường làm việc này, con người có thể làm việc song song với các cobot trong khoảng cách gần, mà không bị giảm đi sự an toàn hoặc hiệu quả [23]. Cobot được thiết kế để tập trung vào các hoạt động lặp đi lặp lại, nhằm giúp người điều hành có thể tập trung vào các nhiệm vụ khó. Do đặc tính chia sẻ không gian làm việc chung, khi nghiên cứu và phát triển cobot, sự an toàn của con người phải là yếu tố được xem xét đầu tiên. Cobot phải có phản ứng khi có vật thể lạ, hoặc va chạm với vật thể, con người trong quá trình làm việc. Bên cạnh đó, không gian làm việc của cobot phải được giám sát. Phản ứng có thể ở các mức độ khác nhau tùy vào hệ thống và yêu

cầu làm việc. Về mặt an toàn, cần cảnh báo bằng âm thanh và ánh sáng báo động và dừng cobot khi có va chạm.

## 1.2. Kiến thức nền tảng

### 1.2.1. Thực tại ảo

#### 1.2.1.1. Khái niệm

Thực tại ảo đang ngày càng phát triển rộng rãi và ứng dụng trong nhiều lĩnh vực của cuộc sống. Do vậy có nhiều khái niệm về thực tại ảo theo các góc độ nghiên cứu khác nhau, trong đó khái niệm được đưa ra bởi Burdea [40] được sử dụng phổ biến nhất: *“Thực tại ảo là một hệ thống giao diện cấp cao giữa người sử dụng và máy tính. Hệ thống này mô phỏng các sự vật, hiện tượng theo thời gian thực và tương tác với người sử dụng qua tổng hợp các kênh cảm giác (thị giác, thính giác, xúc giác,...)”*.

Hình 1.1 mô tả một hệ thống mô phỏng tác chiến của quân đội Hoa Kỳ sử dụng công nghệ thực tại ảo để đào tạo, huấn luyện các binh sĩ thực hiện các nhiệm vụ nguy hiểm như tháo gỡ bom, mìn. Trong đội hình tác chiến giả định, mỗi người lính tham gia thực hiện nhiệm vụ tháo gỡ bom, mìn được trang bị các thiết bị thực tại ảo cung cấp đầy đủ thông tin về mục tiêu, kế hoạch tác chiến, dữ liệu vị trí bom, mìn,...



Hình 1.1. Hệ thống thực tại ảo của quân đội Hoa Kỳ phục vụ công tác huấn luyện tác chiến



Trong môi trường mô phỏng thực tại ảo, người sử dụng có thể trở thành một phần của hệ thống. Người sử dụng được nhập vai để có thể tự do chuyển động trong không gian 3D, tương tác với các vật thể 3D. Ngược lại, môi trường ảo có những tác động phản hồi tương ứng với các hành động của người sử dụng tuân theo những quy tắc tự nhiên, vật lý, toán học,... làm con người có cảm giác như đang tồn tại trong một thế giới thực. Môi trường mô phỏng nhân tạo này không tĩnh tại mà có những phản ứng tức thời, thay đổi theo tín hiệu vào của người sử dụng. Điều này xác định một đặc tính chính của thực tại ảo, đó là tương tác thời gian thực (real-time interactivity), có nghĩa là hệ thống có khả năng nhận biết được tín hiệu vào của người sử dụng và thay đổi ngay lập tức và người sử dụng cảm nhận thấy sự vật thay đổi trên giao diện mô phỏng ngay theo ý muốn của họ.

Ở nước ta hiện nay, thực tại ảo là một trong những công nghệ đang được các cơ sở giáo dục đặc biệt quan tâm. Ứng dụng công nghệ thực tại ảo vào giáo dục, đào tạo sẽ giúp các giảng viên nâng cao hiệu quả dạy học, hỗ trợ cho việc nêu các tình huống có vấn đề để kích thích tư duy sáng tạo của người học. Người học có thể dễ dàng lĩnh hội tri thức thông qua các hình ảnh, sự vật, hiện tượng khi tiếp xúc với chúng bằng cách trực quan sinh động như trong môi trường thực. Tất cả những sự mô phỏng này đóng vai trò rất quan trọng trong việc tiếp thu kiến thức của người học một cách hiệu quả.

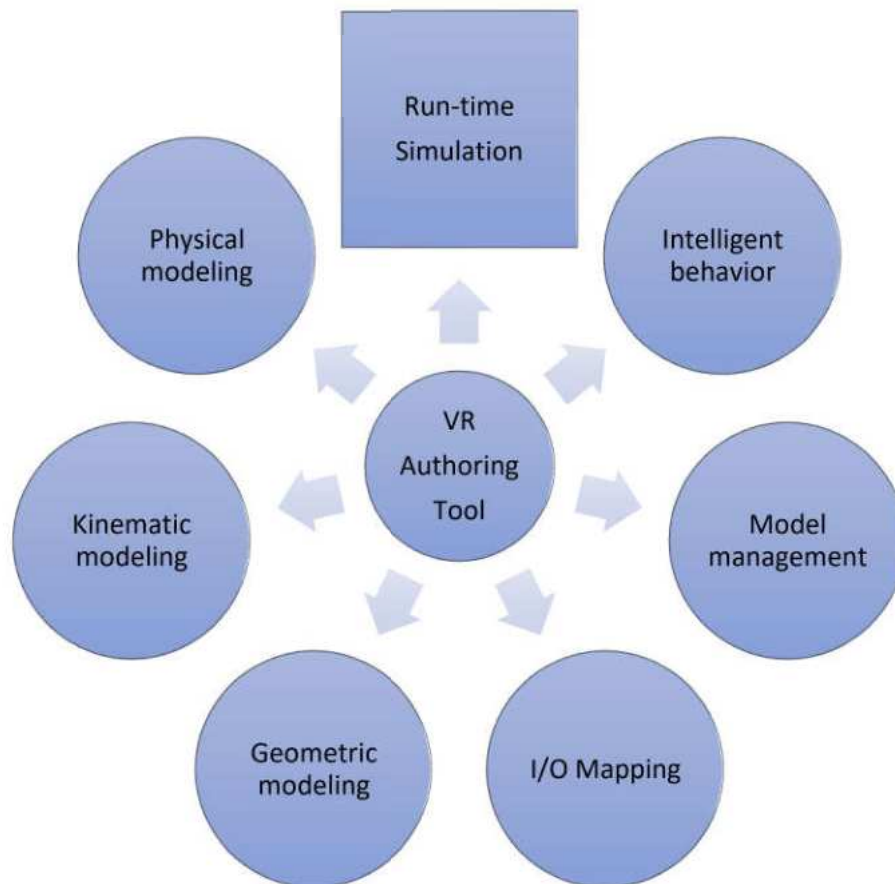
Có ba đặc tính chính của hệ thống thực tại ảo là tương tác (*Interactive*), đắm chìm (*Immersion*) và tưởng tượng (*Imagination*). Tính tương tác là khả năng người dùng giao tiếp với hệ thống, cho phép họ có thể điều khiển hoặc làm thay đổi trạng thái của môi trường ảo bằng các hành động, thao tác. Tính đắm chìm là cảm giác như đang có mặt trong hệ thống ảo hoặc là một phần của môi trường ảo do máy tính tạo ra và hòa lẫn vào thế giới đó. Tính tưởng tượng liên quan đến việc giải quyết các vấn đề trong thế giới thực.

Với những đặc tính như vậy, công nghệ thực tại ảo được nghiên cứu áp dụng và phát huy hiệu quả cho không chỉ trong lĩnh vực giáo dục và đào

tạo [41], mà còn được ứng dụng trong các lĩnh vực y tế, an ninh, quốc phòng, bảo tàng, thời trang, du lịch, vũ trụ và không gian,...

### 1.2.1.2. Mô hình hóa thực tại ảo

Quy trình mô hình hóa thực tại ảo bao gồm các bước: Mô hình hóa hình học (*Geometric modeling*), mô hình hóa chuyển động/động học (*Kinematic modeling*), mô hình hóa vật lý (*Physical modeling*), các ứng xử thông minh (*Intelligent behavior*) và quản lý mô hình (*Model management*).



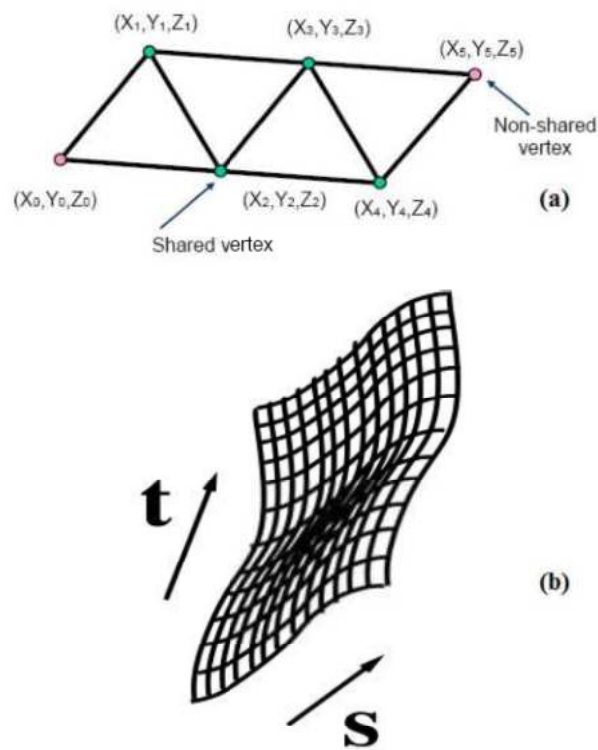
Hình 1.2. Quy trình mô hình hóa thực tại ảo

#### *Mô hình hóa hình học*

Mô hình hoá hình học là việc mô tả hình dạng của các đối tượng ảo (đa giác, tam giác, điểm, spline) cũng như bề mặt (bề mặt kết cấu, chiếu sáng bề mặt và màu sắc). Hình dạng của đối tượng ảo được mô phỏng bởi bề mặt 3D. Bề mặt 3D có thể mô tả được bằng nhiều cách. Có hai cách thường được

sử dụng là lưới đa giác và bề mặt tham số. Đa số các đối tượng ảo có bề mặt được tạo từ các lưới đa giác.

Lưới đa giác được ưu tiên vì chúng sử dụng các đỉnh. Hình 1.3.a minh họa các đỉnh của các cạnh cũng như các cạnh liền kề của các tam giác. Ví dụ, cạnh nối hai đỉnh  $(X_1, Y_1, Z_1)$  và  $(X_2, Y_2, Z_2)$  là cạnh nằm giữa tam giác thứ nhất và tam giác thứ hai. Sử dụng lưới đa giác có thể tiết kiệm được bộ nhớ. Tuy nhiên, lưới đa giác kém hiệu quả trong việc mô tả các đối tượng có độ cong lớn và các bề mặt gồ ghề.



Hình 1.3. Mô tả hình dạng đối tượng: (a) sử dụng lưới tam giác, (b) sử dụng bề mặt tham số

Các đối tượng phức tạp thường được biểu diễn bằng bề mặt tham số. Bề mặt tham số được biểu diễn bằng các điểm  $x(t)$ ,  $y(t)$  và  $z(t)$  như sau:

$$x(t) = a_x \cdot t^3 + b_x \cdot t^2 + c_x \cdot t + d_x \quad (1.1)$$

$$y(t) = a_y \cdot t^3 + b_y \cdot t^2 + c_y \cdot t + d_y \quad (1.2)$$

$$z(t) = a_z \cdot t^3 + b_z \cdot t^2 + c_z \cdot t + d_z \quad (1.3)$$

trong đó,  $t$  nhận giá trị trong khoảng  $[0, 1]$  và  $a, b, c$  là các hằng số.

Bề mặt tham số mở rộng là sự có mặt của tham số thứ hai,  $s$  cũng có giá trị trong khoảng  $[0, 1]$ , khi đó các điểm được biểu diễn bằng tọa độ  $x(s, t)$ ,  $y(s, t)$  và  $z(s, t)$ .

Mô hình hoá là bước cần thiết đầu tiên, bước tiếp theo ánh xạ kết cấu bề mặt.

Ánh xạ kết cấu bề mặt (Texture Mapping): Kết cấu hoá bề mặt là công nghệ thông dụng trên bàn quét của các đường đồ họa để thay đổi được thuộc tính bề mặt của mô hình như màu sắc hay độ phân giải,... Ánh xạ kết cấu bề mặt là phương pháp cải tiến ảnh thực thành các đa giác bề mặt. Toàn bộ điểm của đối tượng được lưu trữ thời gian mô hình hoá trong không gian tọa độ đối tượng  $(x, y, z)$ . Mô hình lưu trữ toàn bộ trong không gian giới hạn.

#### *Mô hình hóa chuyển động*

Bước tiếp theo trong mô hình hóa thế giới ảo, sau thiết lập thuộc tính, cách thể hiện các đối tượng hình học là mô hình hóa chuyển động. Nó xác định vị trí của các đối tượng 3D với các khía cạnh trong hệ tọa độ của thế giới thực cũng như chuyển động của chúng trong thế giới ảo. Chuyển động của các đối tượng có mối quan hệ phân cấp cha-con, với việc chuyển động của đối tượng cha sẽ ảnh hưởng đến chuyển động của đối tượng con.

#### *Mô hình hóa vật lý*

Mô hình hóa vật lý là quá trình tích hợp sự mô phỏng các quá trình vật lý để thiết lập các thuộc tính vật lý cho đối tượng trong thế giới ảo, mang đến sự chân thực như trong môi trường tự nhiên. Các đặc tính vật lý đó bao gồm: cơ học, trọng lượng, quán tính, đặc tính bề mặt, sự biến dạng hay độ đàn hồi, độ cứng,... Cần phải xử lý về vấn đề tính toán ảnh hưởng của các tác động cơ học lên đối tượng, để từ đó thực hiện mô hình hóa và diễn họa xúc giác (*haptic rendering*) trong thực tại ảo. Phát hiện va chạm là tác vụ quan trọng của quá trình mô hình hóa vật lý.

### 1.2.2. Cấu tạo mô hình 3D

Mô hình 3D là một cấu trúc mô tả hình thái 3D của một đối tượng. Hiện nay để tạo ra một mô hình 3D có nhiều cách khác nhau bằng các phần mềm thiết kế 3D như 3DS Max, Maya hoặc từ các máy quét 3D. Một mô hình gồm có ba thành phần cơ bản là tập danh sách các đỉnh (VertexList), tập danh sách các mặt (FaceList) và tập UV kết hợp với một ảnh chất liệu bên ngoài để tạo ra hình ảnh của mô hình với bề mặt giống với thực tế. Trong mô hình 3D, tập đỉnh là tập các vector 3D mà mỗi vector là một điểm trong không gian ba chiều. Tập đỉnh này sẽ quy định hình dạng 3D của đối tượng. Tập các mặt được sử dụng để kết nối các đỉnh với nhau từ đó tạo ra bề mặt của đối tượng. Tập các đỉnh và tập các mặt về cơ bản tạo ra một mô hình 3D hình dạng giống một lưới dựa trên quan hệ giữa các đỉnh và các mặt.

Để mô hình giống thực tế hơn, người ta xác định một texture và một tập UV để quy định việc sử dụng texture trên mỗi bề mặt của đối tượng.

Ví dụ về một đối tượng 3D cơ bản là một khối hộp được tạo ra dựa trên dạng lưới tam giác như sau:

-Tập các đỉnh

```
var size = 100;
Vector3 [] VertexList=new Vector3 []{
new Vector3(-size, -size, -size),
new Vector3(-size, size, -size),
new Vector3( size, size, -size),
new Vector3( size, -size, -size),
new Vector3( size, -size, size),
new Vector3( size, size, size),
new Vector3(-size, size, size),
new Vector3(-size, -size, size)
};
```

Khối hộp có 8 đỉnh, vị trí các đỉnh được sắp xếp trong không gian tương ứng với tập vector ba chiều VertexList được tạo ở trên.

## -Tập các mặt

```
int []FaceList = new int []{
0, 1, 3, // Mặt sau
1, 2, 3,
3, 2, 5, // Mặt trước
3, 5, 4,
5, 2, 1, // Mặt trên
5, 1, 6,
3, 4, 7, // Mặt dưới
3, 7, 0,
0, 7, 6, // Mặt trái
0, 6, 1,
4, 5, 6, // Mặt phải
4, 6, 7
};
```

Khối hộp xác định 6 mặt tương ứng là sau, trước, trên, dưới, trái, phải. Mô hình này sử dụng các mặt ở dạng tam giác, tức là một mặt được tạo ra từ ba đỉnh, giá trị và quan hệ của các mặt được mô tả trong tập mặt FaceList ở trên. Sau khi xác định được các thành phần cơ bản của một mô hình 3D mà cụ thể ở đây là một khối hộp 3D sẽ thu được kết quả là một mô hình ba chiều.

### *Một số phương pháp tạo mô hình 3D*

Phương pháp tạo mô hình 3D từ tổng thể đến chi tiết: Các vật thể hữu hình trong cuộc sống hầu hết được cấu tạo nên từ những hình khối cơ bản như khối cầu, khối trụ, khối hộp, hình chóp nhọn,... Qua việc quan sát vật thể cần mô phỏng hay thiết kế rút ra những nhận xét, cấu tạo của vật thể để sử dụng công cụ tạo trực tiếp hoặc sử dụng kết hợp nhiều hình khối, lấy phần giao, phần bù để tạo được vật thể như ý muốn.

Phương pháp tạo mô hình 3D từ chi tiết đến tổng thể: Ngược lại với phương pháp trên, để bắt tay vào xây dựng vật thể theo phương pháp này ta sẽ phải chuẩn bị ảnh hình chiếu của vật thể thường là hình chiếu mặt trước và

hình chiếu mặt trái hoặc hình chiếu mặt phải. Ưu điểm của phương pháp này là độ chính xác cao hơn phương pháp tạo mô hình 3D từ tổng thể đến chi tiết nêu trên, tuy nhiên phương pháp này cần nhiều thời gian và độ tỉ mỉ hơn.

Phương pháp thiết kế phối hợp: Tận dụng hiệu quả của cả hai phương pháp trên, một cách hiệu quả là sử dụng linh hoạt kết hợp cả hai phương pháp. Các vật thể trong cuộc sống đều tương đối phức tạp, bằng khả năng phân tích cùng với sự linh hoạt công việc thiết kế mô phỏng sẽ được thực hiện dễ dàng và nhanh chóng, hiệu quả hơn.

Phương pháp tạo mô hình bằng máy quét 3D: Có thể tạo ra các mô hình từ các thiết bị phần cứng là máy quét 3D. Các thiết bị phần cứng khác nhau hiện đang được sử dụng để quét nhanh với độ phân giải rất cao.

Như vậy, có ba nhóm phương pháp chính để tạo ra mô hình 3D. Một là sử dụng các lệnh trong ngôn ngữ lập trình để thiết kế ra các mô hình. Hai là sử dụng các phần mềm thiết kế 3D tạo ra mô hình từ các bản vẽ thiết kế. Ba là sử dụng các thiết bị máy quét 3D tạo mô hình từ vật thể thực. Với mỗi nhóm phương pháp tạo mô hình đều mang những ưu, nhược điểm khác nhau. Nhóm phương pháp đầu khá hạn chế về chất lượng mô hình và đòi hỏi phải tính toán nhiều trong quá trình lập trình để tạo ra các đối tượng 3D. Với nhóm phương pháp thứ hai là sử dụng khả năng tạo ra mô hình 3D với khả năng tối ưu về lưới, tuy nhiên độ tối ưu của mô hình phụ thuộc vào kiến thức và kinh nghiệm của từng nhà thiết kế. Nhóm phương pháp này cũng đòi hỏi chi phí lớn, trong đó thời gian để tạo ra một sản phẩm là tương đối dài, độ đồng đều và ổn định không cao do phụ thuộc vào kỹ năng của người thiết kế. Nhóm phương pháp sử dụng các thiết bị phần cứng là máy quét để tạo mô hình 3D mang nhiều ưu điểm như: Thời gian tạo ra một mô hình ngắn, độ chính xác cao và có tính ổn định. Tuy nhiên, mô hình tạo ra từ máy quét có một nhược điểm lớn chính là số lượng lưới lớn (số các đỉnh và số các mặt lớn), điều đó ảnh hưởng rất nhiều tới việc sử dụng mô hình từ máy quét trong thực tế. Với một chương trình mô

phòng thực tại ảo cần sử dụng một số lượng lớn mô hình tạo ra từ máy quét sẽ ảnh hưởng tới quá trình xử lý, lưu trữ, tính toán, hiển thị, tương tác.

### ***1.2.3. Mô hình vật thể biến dạng***

Biến dạng là một hiện tượng tồn tại phổ biến trong tự nhiên [8], được định nghĩa là sự thay đổi về vị trí, hình dạng hoặc kích thước của vật thể trong không gian theo thời gian dưới tác động của lực hoặc nhiệt độ. Phạm vi nghiên cứu của luận án được xác định tập trung vào phát hiện va chạm trong mô hình vật thể biến dạng do tác động bởi tiếp xúc.

Mô hình chất liệu vải là một dạng điển hình của mô hình vật thể biến dạng nhưng hiện chưa có nhiều công trình nghiên cứu chuyên sâu về phát hiện va chạm của các mô hình chất liệu vải. Trong chương 2, NCS trình bày về giải pháp và đề xuất thuật toán phát hiện va chạm của chất liệu vải tương tác trong môi trường thực tại ảo.

Vải là một thuật ngữ chung bao gồm các chất liệu dựa trên sợi như sợi, chỉ, filament và loại vải khác. Ban đầu, vải chỉ ám chỉ các loại vải dệt kim. Tuy nhiên, dệt không phải là phương pháp duy nhất; nhiều phương pháp khác đã được phát triển để tạo cấu trúc vải dựa trên mục đích sử dụng. Đan len và vải không dệt cũng phổ biến. Trong thế giới hiện đại, vải đáp ứng nhu cầu chất liệu cho nhiều ứng dụng, từ quần áo hàng ngày đến chăn, rèm, mũ, khăn, lá cờ,...

Vải được chia thành hai loại: vải tiêu dùng cho gia đình và vải công nghiệp. Trong vải tiêu dùng, thẩm mỹ và thoải mái quan trọng, còn trong vải kỹ thuật, tính chức năng được ưu tiên. Vải công nghiệp bao gồm vải địa kỹ thuật, vải y tế và nhiều lĩnh vực khác, còn quần áo và đồ nội thất là ví dụ về vải tiêu dùng. Mỗi thành phần của sản phẩm vải, từ sợi, chỉ, vải đến quy trình sản xuất và hoàn thiện, đều ảnh hưởng đến sản phẩm cuối cùng và được chọn dựa trên tính thích hợp cho mục đích sử dụng.



Sợi là thành phần nhỏ nhất của vải, thường được quấn thành chỉ và dùng để sản xuất vải. Sợi có hình dáng giống tóc và tỷ lệ chiều dài so với chiều rộng cao hơn. Nguồn gốc của sợi có thể là sợi tự nhiên, sợi tổng hợp, hoặc kết hợp cả hai. Các sợi được xoắn hoặc xếp để tạo thành một dây chỉ liên tục. Chỉ sau đó được sử dụng để tạo ra vải thông qua việc dệt, đan, nút thúc, kết nút, đan nơ, hoặc bện. Sau khi sản xuất, vật liệu dệt kim được xử lý và hoàn thiện để tăng giá trị, bao gồm thẩm mỹ, đặc điểm vật lý và tính hữu ích tăng cao.

Vải được định nghĩa là một loại vật liệu mỏng, linh hoạt nào được làm từ sợi. Vải là chất liệu bao phủ xung quanh và là một phần thiết yếu của cuộc sống con người. Mô phỏng chất liệu vải có nhiều ứng dụng trong các hệ thống tương tác thực tại ảo, trong đó bài toán phát hiện va chạm của mô hình chất liệu vải cần tập trung giải quyết nhằm tránh tình trạng hệ thống bị *ngheñ nút cổ chai*. Kế thừa và phát triển từ các phương pháp của Tang [10] và Zhang [18] về phát hiện va chạm trong các mô hình vật thể biến dạng, phương pháp đề xuất của NCS được trình bày chi tiết trong chương 2 cho phép tăng tốc độ phát hiện va chạm trong các mô hình vải khác nhau thuộc bộ dữ liệu GAMMA<sup>1</sup> (Geometric Algorithms for Modeling, Motion, and Animation) của Nhóm nghiên cứu UMD GAMMA của Trường Đại học Maryland, Hoa Kỳ.

#### ***1.2.4. Quy tắc dấu Descartes và Định lý Vincent***

Có thể phát hiện va chạm trên mô hình lưới tam giác không đều (TIN) dựa trên các phép toán đại số và các đặc điểm hình học liên quan đến tính đồng phẳng của các cặp đỉnh – mặt, cạnh – cạnh. Ý tưởng chính là xác định xem có tồn tại bất kỳ nghiệm nào trong khoảng thời gian  $[0,1]$  hay không bằng những cách đơn giản và chi phí thấp. Nếu sự không tồn tại của nghiệm nguyên trong khoảng thời gian  $t \in [0,1]$  được xác định sớm hơn thì có thể

---

<sup>1</sup> <https://gamma.umd.edu/>

tránh được việc phải giải một số lượng rất lớn các phương trình bậc ba đòi hỏi chi phí tính toán tốn kém.

**Quy tắc dấu Descartes [16]:** Gọi  $f(t)$  là một đa thức theo lũy thừa giảm dần của  $t$ . Biên đổi dấu xảy ra bất cứ khi nào các hệ số khác không liên tiếp là khác dấu. Ký hiệu  $VAR(f(t))$  là số cặp hệ số khác không liên tiếp có dấu ngược nhau.

Do chỉ quan tâm đến đoạn  $[0,1]$ , Định lý sau là kết quả của quy tắc Descartes về dấu khi ánh xạ  $t$  từ  $(0, \infty)$  đến  $(0,1)$  với ánh xạ  $t \rightarrow \frac{1}{t+1}$ .

**Định lý Vincent [17]:** Xét hàm  $g(t) = (t+1)^n f\left(\frac{1}{t+1}\right)$ , nếu  $VAR(g(t)) = 0$ , với  $n$  là bậc của  $f(t)$  thì  $f(t)$  không có nghiệm nào trên đoạn  $[0,1]$ .

Số nghiệm thực dương của  $g(t)$  bằng số nghiệm thực của  $f(t)$  trên đoạn  $[0,1]$ .

Áp dụng quy tắc dấu Descartes cho  $g(t)$ , ta có kết luận như trong bảng dưới đây:

*Bảng 1.1.* Bảng mô tả sự tồn tại của nghiệm

$VAR(g(t))$	Số nghiệm
0	0
1	1
2	2 hoặc 0
3	3 hoặc 1

Từ Bảng 1.1 có thể thấy rằng việc sử dụng Định lý Vincent có thể nhanh chóng xác định sự tồn tại của nghiệm của phương trình khoảng cách nhằm nhanh chóng xác định có hay không xảy ra va chạm.

## ***1.2.5. Cobot và cấu tạo của cánh tay cobot***

### ***1.2.5.1. Cobot***

Cùng với sự phát triển của công nghệ robotics nói chung, cobot cũng đang trở thành một đối tượng được quan tâm nghiên cứu, phát triển trong những năm gần đây bởi khả năng thay thế con người thực hiện các công việc mang tính lặp đi lặp lại, trong các môi trường độc hại, nguy hiểm và trong nhiều lĩnh vực. Theo Allied Market Research [70-72], năm 2018 thị trường cobot đạt 9.340 triệu đô la và dự kiến sẽ đạt 39.585 triệu đô la vào năm 2026, với tỷ suất tăng trưởng hàng năm từ năm 2019 đến năm 2026 (CAGR) là 21,5%. Trong đó, thị trường Bắc Mỹ lớn nhất toàn cầu với 3.933,1 triệu đô la vào năm 2018 và dự kiến đạt 14.492,1 triệu đô la vào năm 2026. Nghiên cứu, chế tạo cobot là lĩnh vực của khoa học và công nghệ nhằm mục tiêu tạo ra các cobot có thể di chuyển trong mọi môi trường mà không cần sự can thiệp của con người. Chúng có thể tự động di chuyển và làm việc trong nhà máy, phòng thí nghiệm,... thông qua các thiết bị như cảm biến, phần mềm và các thiết bị các được gắn kèm trên cobot. Hiện nay, các vấn đề cơ bản của cobot đang được quan tâm, tập trung nghiên cứu là phát hiện va chạm, chuyển động, cảm biến và điều hướng.

Cobot có thể làm việc đồng thời với người vận hành. Theo ISO 10218 [73], cobot là những robot có thể tiếp xúc với con người một cách trực tiếp trong một không gian làm việc cộng tác xác định. Cobot được thiết kế để bảo đảm tương tác an toàn với con người cũng như môi trường làm việc cộng tác. Cobot được chế tạo bằng vật liệu nhẹ và thiết kế với hình dạng mô phỏng tương tự cánh tay người. Cobot hoạt động phối kết hợp với con người để tăng hiệu năng quá trình sản xuất hoặc khắc phục các hạn chế kỹ thuật mà không cần đến các thiết bị vật lý bảo hộ như tấm chắn, lưới sắt, hàng rào thép bảo vệ [74, 75].

Để tương tác tự do với con người, cobot cần đáp ứng một số tiêu chuẩn an toàn (đặc biệt là: ISO 10218 và ISO/TS 15066) [88-90]. Trên cơ sở các tiêu chuẩn này, có thể phân biệt một số bước quan trọng đối với sự an toàn của cánh tay cobot:

- Dự đoán (sự kiện) va chạm;
- Các chiến lược trước va chạm (tránh va chạm);
- Phát hiện sự kiện va chạm;
- Chiến lược sau va chạm (dừng cobot).

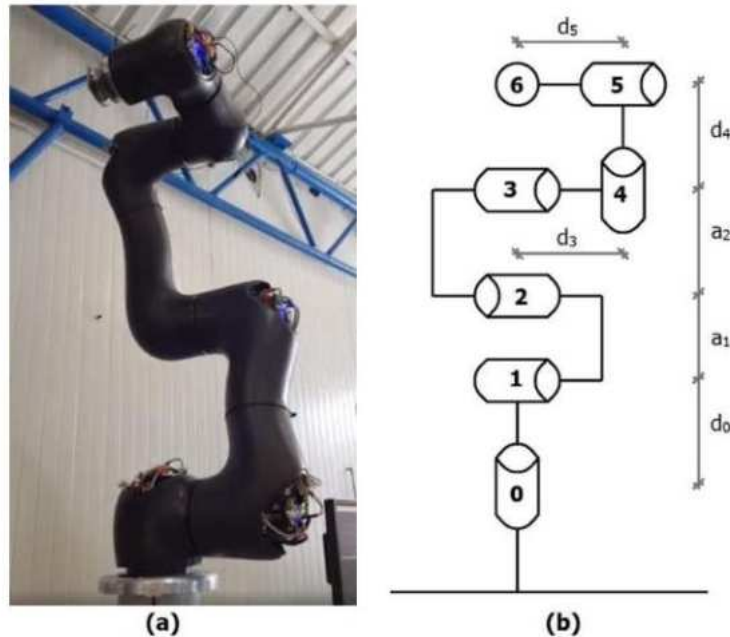
Bước 1: Việc phát hiện các sự kiện va chạm yêu cầu lắp đặt các cảm biến (thường có chi phí cao) và cài đặt thuật toán thích hợp. Các giải pháp sau đây được sử dụng để dự đoán sự kiện va chạm: Optical zone barriers; Pressure mats and strips; Safety keys; Motion and distance sensors; Vision sensors [91-97].

Bước 2: Các chiến lược trước khi va chạm có tính đến hành động sẽ xảy ra khi có đối tượng đi vào vùng không gian của cobot. Tiêu chuẩn kỹ thuật ISO/TS 15066 liên quan đến việc giảm tốc độ của cobot trong trường hợp xác định vị trí đối tượng xâm nhập trong khu vực làm việc và giám sát việc tách quỹ đạo của cobot khỏi quỹ đạo của con người.

Bước 3: Để phát hiện một sự kiện va chạm, cobot chủ yếu sử dụng cảm biến mô-men xoắn. Có thể phát hiện va chạm bằng cảm ứng (hệ thống APAS của BOSCH) hoặc các cảm biến khác, chẳng hạn như CLS (Compliant Limb Sensor) [98], cảm biến chạm [99] hoặc các loại cảm biến ảo [100-103].

Bước 4: Các chiến lược sau va chạm quy định những việc cần làm sau khi va chạm - về cơ bản nó là một điểm dừng liên quan đến an toàn. Lưu ý rằng cả hai chiến lược có thể được hiểu là một phản ứng vô thức đối với một kích thích cụ thể: xa trong trường hợp dự đoán và gần trong trường hợp va chạm trực tiếp [104].

Đã có nhiều công bố về bài toán phát hiện va chạm với đối tượng nghiên cứu là cánh tay cobot [107-118]. Điển hình là cánh tay cobot sáu bậc tự do 6-DoF (Six Degrees of Freedom) được thiết kế nhằm thực hiện các tác vụ chính xác và an toàn với con người. Hình 1.4 mô tả cánh tay cobot Cooperative Universal Robotic Assistant 6 (CURA6) của Intema [25].



Hình 1.4. (a) Hình ảnh cobot CURA6 tại phòng thí nghiệm của Intema;  
(b) Mô hình cobot CURA6

#### 1.2.5.2. Cấu tạo của cánh tay cobot

Các bộ phận chủ yếu của một cobot CURA6 bao gồm: đế, các khớp (Joint) quay hoặc tịnh tiến, các liên kết (Link) giữa các khớp và bàn tay cobot còn được gọi khác là khâu chấp hành cuối (End of Effect). Cánh tay cobot CURA6 của Intema có đặc trưng nhỏ gọn với trọng tải 5000 g, bán kính hoạt động rộng 1200 mm. Cobot CURA6 có thể thực hiện được các nhiệm vụ phức tạp với yêu cầu tính chính xác cao như thao tác xử lý, tiếp xúc đối với những vật có đặc tính dễ biến dạng hoặc dễ rách, vỡ trong những điều kiện vận hành với yêu cầu kỹ thuật rất khắt khe.

Bảng 1.2 thể hiện bảng tham số động học Denavit - Hartenberg (D-H) của cobot CURA6 [25].

Bảng 1.2. Bảng tham số D-H của cobot CURA6

$i$	$a_i$	$\alpha_i$	$d_i$	$\tau_i$	$\theta_i$
0	0	$\frac{\pi}{2}$	0,105	$\tau_0$	$\theta_0$
1	0,4	0	-	$\tau_1$	$\theta_1$
2	0,4	0	-	$\tau_2$	$\theta_2$
3	0	$\frac{\pi}{2}$	0,22	$\tau_3$	$\theta_3$
4	0	$-\frac{\pi}{2}$	0,2	$\tau_4$	$\theta_4$
5	0	0	0,14	$\tau_5$	$\theta_5$

Ký hiệu trong bảng trên:

$a_i$  là tham số khoảng cách tính từ trục  $Z_{i-1}$  tới trục  $Z_i$  được đo dọc theo trục  $X_i$

$\alpha_i$  là tham số góc quay trục  $Z_{i-1}$  xung quanh trục  $X_i$  tới song song hoặc trùng với trục  $Z_i$

$d_i$  là tham số khoảng cách giữa 2 trục  $X_{i-1}$  và trục  $X_i$  được đo dọc theo trục  $Z_{i-1}$

$\theta_i$  là tham số góc quay của trục  $X_{i-1}$  xung quanh trục  $Z_{i-1}$  tới song song với trục  $X_i$

### 1.2.5.3. Phương trình động lực học

Phương trình động lực học được định nghĩa cho cobot CURA6 là:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_F = \tau_m \quad (1.4)$$

với  $q(t) = [q_0(t), q_1(t), \dots, q_{n-1}(t)] \in \mathbb{R}^n$  bao gồm vị trí của mỗi khớp (với  $n = 6$ ),  $\dot{q}(t) \in \mathbb{R}^n$  và  $\ddot{q}(t) \in \mathbb{R}^n$  lần lượt biểu thị vectơ vận tốc và gia tốc.  $M(q) \in \mathbb{R}^{n \times n}$  là ma trận quán tính cobot,  $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$  là ma trận Coriolis,  $g(q) \in \mathbb{R}^n$  là lực hấp dẫn,  $\tau_F \in \mathbb{R}^n$  biểu thị lực ma sát và  $\tau_m \in \mathbb{R}^n$  là vectơ mômen khớp. Lưu ý  $M(q)$  là ma trận đối xứng;  $\tau_m = K_i i_m$  với  $K_i \in$

$\mathbb{R}^{n \times n}$  là ma trận khuếch đại và  $i_m \in \mathbb{R}^n$  là dòng điện động cơ. Khi va chạm xảy ra, phương trình (3.1) trở thành:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_F = \tau_m + \tau_{ext} \quad (1.5)$$

với  $\tau_{ext} \in \mathbb{R}^n$  là mômen xoắn khớp ngoài do va chạm hoặc lực bên ngoài khác tác động lên cobot.

Ma trận  $\dot{M}(q) - 2C(q, \dot{q})$  là đối xứng xiên, với  $\dot{M}(q)$  là ký hiệu đạo hàm của  $M(q)$  theo thời gian (chi tiết trong [60]).

$$\dot{M}(q) = C(q, \dot{q}) + C^T(q, \dot{q}) \quad (1.6)$$

Động lượng tổng quát của cobot xác định như sau:

$$p = M(q)\dot{q} \in \mathbb{R}^n \quad (1.7)$$

Đạo hàm của  $p$  theo thời gian được biểu diễn là:

$$\dot{p} = \dot{M}(q)\dot{q} + M(q)\ddot{q} \quad (1.8)$$

hay:

$$\dot{p} = \tau_m + \tau_{ext} - \tau_F + C^T(q, \dot{q})\dot{q} - g(q) \quad (1.9)$$

Ký hiệu  $\beta(q, \dot{q}) = g(q) - C^T(q, \dot{q})\dot{q}$  và ma trận  $K_O = \text{diag}\{k_{O,i}\} \in \mathbb{R}^{n \times n}$ , động lực học của bộ quan sát động lượng được định nghĩa như sau:

$$\dot{\hat{p}} = \tau_m - \tau_F - \hat{\beta}(q, \dot{q}) + r_m \quad (1.10)$$

và:

$$\dot{r}_m = K_O(\dot{\hat{p}} - \dot{p}) \quad (1.11)$$

Đầu ra  $r_m(t) \in \mathbb{R}^n$  của bộ quan sát động lượng được mô tả như sau [27]:

$$r_m = K_O \left( p(t) - \int_0^t (\tau_m - \tau_F - \hat{\beta} + r_m) ds - p(0) \right) \quad (1.12)$$

với  $p = \widehat{M}(q, \dot{q})$  và  $\widehat{\beta} = \widehat{g}(q) - \widehat{C}^T(q, \dot{q})\dot{q}$ , trong đó  $\widehat{M}$ ,  $\widehat{C}$ , và  $\widehat{g}$  biểu thị các tham số mô hình danh nghĩa. Trong các điều kiện lý tưởng khi  $\widehat{M} = M$  và  $\widehat{\beta} = \beta$ , mối quan hệ động lực học giữa  $r_m(t)$  và mômen xoắn khớp ngoài  $\tau_{\text{ext}}$  là:

$$\dot{r}_m = K_o(\tau_{\text{ext}} - r_m) \quad (1.13)$$

sao cho  $r_m(t)$  đóng vai trò ước tính cho  $\tau_{\text{ext}}$

Mômen xoắn khớp ngoài ước tính được so sánh theo từng thành phần với ngưỡng cố định do người dùng xác định  $\epsilon_{cd} \in \mathbb{R}^n$ , sao cho hàm xác định va chạm  $cd: r_m(t) \rightarrow \{True, False\}$  được định nghĩa là:

$$cd(r_m(t)) = \begin{cases} True & \text{if } |r_m(t)| > \epsilon_{cd} \\ False & \text{if } |r_m(t)| \leq \epsilon_{cd} \end{cases} \quad (1.14)$$

Hàm trả về kết quả dạng nhị phân cho biết có xảy ra va chạm hay không [29, 61].

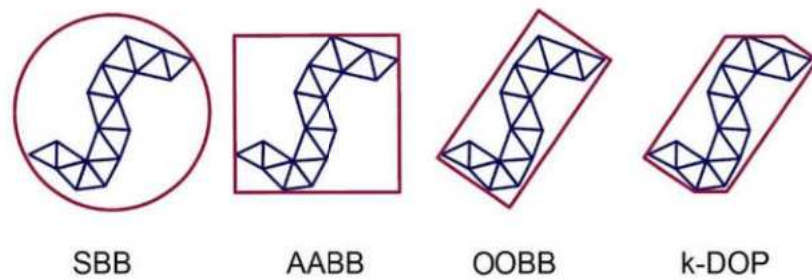
### 1.3. Tổng quan các nghiên cứu liên quan

#### 1.3.1. Phát hiện va chạm của các mô hình vật thể 3D

Mỗi hệ thống thực tại ảo đều phải có khả năng phát hiện khi nào có va chạm xảy ra và có những đối tượng nào va chạm để có những xử lý va chạm thích hợp. Vấn đề đặt ra là xem xét các đối tượng trong không gian 3D nên khi nói đến va chạm có nghĩa là va chạm trong không gian 3D. Một cách đơn giản nhất để phát hiện va chạm giữa hai đối tượng đó là kiểm tra từng mặt của đối tượng này có cắt một mặt nào đó của đối tượng kia, cách này có ưu điểm là trả về chính xác điểm va chạm.

Mỗi vật thể 3D được tạo thành từ nhiều mặt, do đó chi phí để kiểm tra va chạm là rất lớn. Hầu hết các hệ thống thực tại ảo sử dụng phương pháp phát hiện va chạm dựa trên các khối bao. Khối bao là một không gian hình học khép kín để bao bọc hoàn toàn đối tượng. Hình 1.5 biểu diễn một số loại khối bao được sử dụng trong các kỹ thuật phát hiện va chạm.





Hình 1.5. Một số loại khối bao được sử dụng phổ biến trong các kỹ thuật phát hiện va chạm

Khối bao căn chỉnh theo trục AABB (Axis Aligned Bounding Box) [43], khối bao hình cầu SBB (Sphere Bounding Box) [44], khối bao xác định theo hướng đối tượng OBB (Object-Oriented Bounding Box) [45] và khối bao đa diện rời rạc có hướng k-DOP (k-Discrete Oriented Polytopes) [46] được mô tả trong Hình 1.5 theo thứ tự chiều từ trái sang phải thì các khối bao có độ khít tăng dần, còn theo chiều từ phải sang trái thì độ phức tạp tính toán giảm dần.

### 1.3.2. Phát hiện va chạm của cánh tay cobot

Tiếp xúc với đối tượng khác là một trong những tác vụ bắt buộc của cánh tay cobot. Các phương pháp phát hiện va chạm dựa trên học máy và dựa trên cảm biến được phát triển để đáp ứng nhu cầu này.

Phương pháp phát hiện va chạm dựa trên học máy: Y. J. Heo và cộng sự [36] đã đề xuất một phương pháp dựa trên cách tiếp cận học sâu, có khả năng giám sát và nhận ra bất kỳ va chạm nào. Một mạng nơron phức hợp được sử dụng để đạt được cả độ nhạy cao đối với va chạm và độ nhạy thấp đối với cảnh báo sai. A. N. Sharkawy và các cộng sự [32] đã sử dụng Multilayer Feedforward Neural Network (M-FNN) với sai số vị trí khớp và vận tốc làm đầu vào, đồng thời ước tính mômen xoắn khớp bên ngoài mà không cần sử dụng động lực học cobot (nhưng cần điều chỉnh ngưỡng va chạm thủ công cho từng khớp), trong khi [33] xem xét các mômen khớp đo được và vector lực hấp dẫn của tay máy. Tuy nhiên, các phương pháp này chỉ

được thực nghiệm với các chuyển động hai và ba khớp của một cánh tay máy cobot 7-DoF.

Phương pháp phát hiện va chạm sử dụng cảm biến: E. Mariotti và cộng sự [47] đã đề xuất sử dụng một cảm biến lực/mo-men xoắn (F/T) 6D ở bộ hiệu ứng cuối cùng với thông tin động học đầu vào. Tính năng phát hiện lực của cảm biến này giúp ngăn ngừa va chạm ngẫu nhiên và tránh các tiếp xúc ngoài ý muốn trong quá trình chuyển động của cobot. Để phát hiện các điểm tiếp xúc trên cobot và đánh giá các lực tiếp xúc được trao đổi, E. Magrini và cộng sự [48] đã triển khai một thuật toán xử lý đồ họa GPU theo dõi trong thời gian thực khoảng cách động giữa cobot và các chướng ngại vật di chuyển trong môi trường.

Điểm mạnh của cobot chính là tính an toàn, với khả năng vận hành mà không cần rào chắn bảo vệ. Các sản phẩm cobot của hãng Universal Robots đều được chế tạo và cấu hình an toàn phù hợp tiêu chuẩn Châu Âu ENG ISO 13849:2008 PL D. Hệ thống an toàn đã được cấp bằng sáng chế của Universal Robots với 8 chức năng tùy chỉnh cho phép có thể làm việc một cách linh hoạt với cobot:

- Vị trí khớp nối
- Tốc độ khớp nối
- Vị trí TCP
- Định hướng công cụ
- Tốc độ
- Lực
- Động lượng
- Điện năng

Các cobot của Universal Robots đều được tích hợp cảm biến, ngay khi phát hiện tác động từ bên ngoài, cobot sẽ tự động dừng lại

Ngoài ra, một số đặc điểm khác của cobot là:

- Di chuyển linh hoạt và cài đặt nhanh chóng
- Có thể lắp đặt mà không làm thay đổi không gian của nhà máy
- Dễ dàng bảo trì, vận hành đơn giản đặc biệt phù hợp với các nhà máy sản xuất có quy mô vừa và nhỏ.

Trong nền sản xuất công nghệ cao 4.0, mô hình cộng tác người - máy được sử dụng để nâng cao năng suất, tính linh hoạt và tăng lợi nhuận doanh thu [24]. Do đó, cần thiết để phát triển những nghiên cứu phát hiện va chạm nhằm không chỉ đảm bảo sự an toàn đối với con người mà còn nâng cao hiệu suất làm việc cho cobot.

#### **1.4. Một số hạn chế của các phương pháp truyền thống**

##### ***1.4.1. Hạn chế liên quan đến phát hiện va chạm của mô hình vật thể rắn***

Một số dạng khối bao thường dùng trong các kỹ thuật phát hiện va chạm [20, 51]: AABB, SBB, OOB và k-DOP. Việc phát hiện va chạm giữa các vật thể được quy về bài toán phát hiện va chạm giữa các hình bao của chúng. Hai đối tượng được coi là va chạm nhau về tọa độ khi hình bao của chúng che phủ lên nhau. Phát hiện va chạm là quá trình chiếu các hình bao của các vật thể theo các trục tọa độ và xác định xem các hình chiếu này có trùng lên nhau hay không. Bởi vì hình bao dạng AABB đã được định hướng theo các trục của hệ tọa độ không gian nên việc chiếu chúng lên các mặt phẳng tọa độ được thực hiện nhanh hơn các hình bao OOB. Tuy nhiên, các hình bao dạng OOB lại cho kết quả chân thực hơn vì nó mô phỏng chính xác hơn thể tích của vật thể.

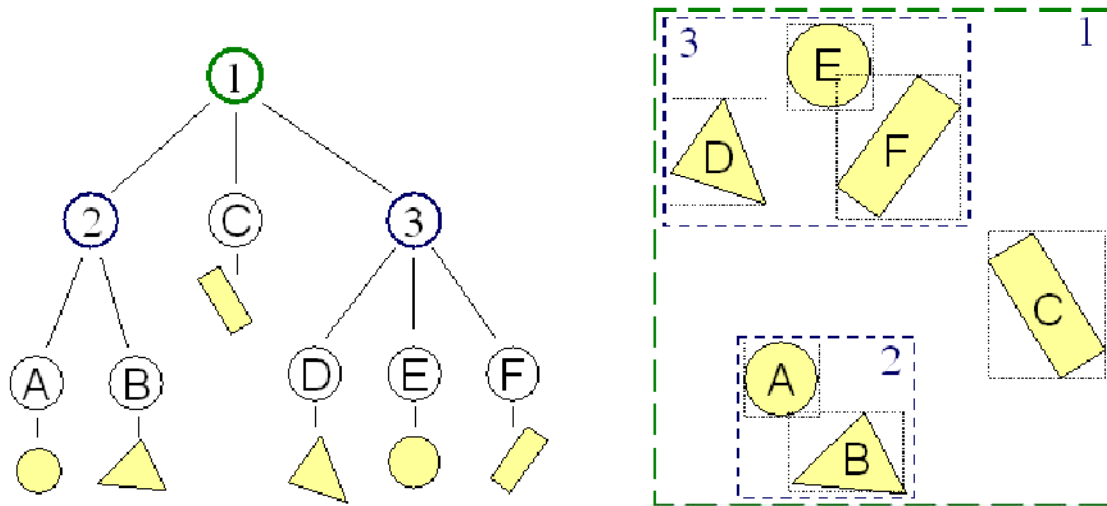
Khi xuất hiện va chạm, việc tiếp theo cần thể hiện trong môi trường là biểu diễn các hành vi tương tác của các vật thể khi va chạm, tức là các đáp ứng va chạm (collision response), chúng phụ thuộc vào các đặc tính của các đối tượng ảo. Nếu các đối tượng tương tác là các đối tượng mềm, thì phải tính

đến một dạng khác của đáp ứng va chạm là biến dạng bề mặt. Biến dạng bề mặt thay đổi dạng hình học của đối tượng 3D.

Nếu đối tượng được mô hình hóa bởi các bề mặt tham số, sự biến dạng bề mặt được tạo ra một cách gián tiếp bởi sự thay đổi vị trí của các điểm điều khiển của bề mặt. Việc thay đổi bề mặt gián tiếp thường khó khăn bởi các đường nội suy kiểu spline. Hơn nữa, việc điều chỉnh các điểm điều khiển phức tạp khi chúng bị che lấp bởi các đối tượng khác. Cách tiếp cận direct free-form deformation (DFFD) cho phép người sử dụng lựa chọn một điểm trên bề mặt tham số của đối tượng, sau đó di chuyển con trỏ đến vị trí mong muốn. Thuật toán sẽ tính toán các thay đổi cần thiết đối với lưới điểm điều khiển, từ đó tạo ra các thay đổi hình dạng bề mặt của đối tượng. Nếu đối tượng được mô hình hóa bởi các đa giác, biến dạng bề mặt được tạo ra một cách trực tiếp thông qua việc điều khiển các đỉnh của đa giác.

Khi người sử dụng tương tác với các bề mặt của đối tượng 3D thì sẽ cảm nhận được các lực phản hồi từ các đối tượng này thông qua găng tay thực tại ảo. Các lực được tính toán bởi các kênh diễn họa xúc giác và được gửi tới bộ phận phản hồi xúc giác để tác động ngược trở lại người sử dụng. Các tính toán lực được xem như là một kiểu của tương tác bề mặt và biến dạng bề mặt, cũng như các đặc tính vật lý và động học của đối tượng. Một tương tác bề mặt đơn giản nhất là tương tác đơn điểm (single-point contact) được thực thi bởi các giao diện xúc giác (haptic interfaces).

Hình 1.6 mô tả phân hệ vùng bao BVHs là một cấu trúc dữ liệu dạng cây được tạo thành trên cơ sở phân tích các đối tượng cần được kiểm tra va chạm. BVHs đóng vai trò quan trọng trong việc biểu diễn các vật thể và cho phép giải quyết nhiều vấn đề trong phát hiện va chạm [21].



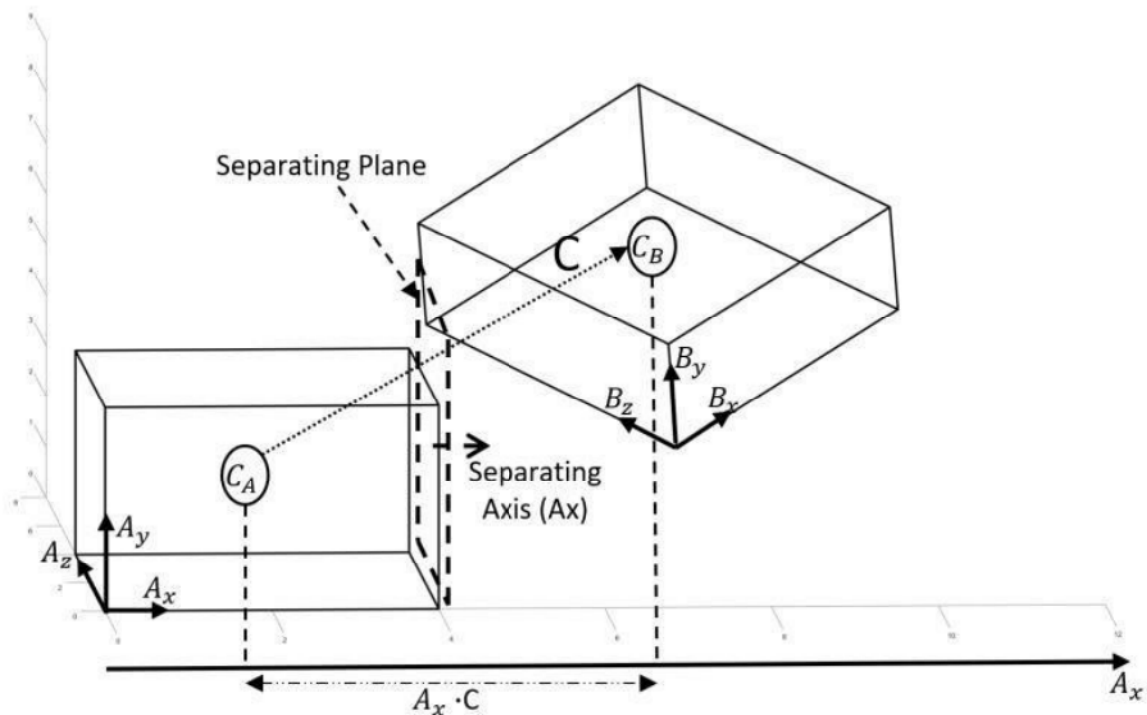
Hình 1.6. Phân hệ vùng bao sử dụng trong các kỹ thuật phát hiện va chạm

Các đối tượng hình học được bao trong các khối, các hình bao là các nút lá của cây. Các nút sau đó được gom thành các nhóm nhỏ và được đóng kín trong các hình bao lớn hơn.

Tuần tự như vậy, chúng lại được gom lại và đóng kín trong các hình bao lớn hơn nữa theo phương pháp lặp. Cuối cùng sẽ tạo nên một cấu trúc cây với một hình bao đơn tại các đỉnh của cây.

Thuật toán Separating Axis Theorem (SAT): Mô tả trực quan thì có thể xem thuật toán SAT tương đương với việc tìm một mặt phẳng có thể tách hai vật thể. Mặt phẳng này được gọi là mặt phẳng phân cách như được mô tả trong Hình 1.7. Do đó, nếu tồn tại một mặt phẳng phân tách thì các vật thể không va chạm và do đó khi không thể tìm thấy mặt phẳng phân tách thì các vật thể có thể xảy ra va chạm. Để xác định xem một trục có phải là trục phân tách hay không, hai đối tượng trước tiên được chiếu lên trục tọa độ. Hình chiếu kết quả của cả hai đối tượng trên trục đó sẽ trùng nhau hoặc không trùng nhau. Nếu hình chiếu không chồng lên nhau, thì trục là trục ngăn cách và xác nhận các đối tượng không có va chạm. Về mặt lý thuyết, có thể có vô số trục cần kiểm tra để xác định xem trục phân tách có tồn tại hay không. Tuy nhiên, hai đa giác lồi rời rạc 3D có thể được ngăn cách bởi các mặt phẳng song song với các mặt của vật thể hoặc bởi một mặt phẳng song song với bất

kỳ cạnh nào. Do đó, có thể giảm được số lượng các phép kiểm tra trong thực tế. Chẳng hạn với hai khối bao OOBs, các tình huống mà chúng tiếp xúc với nhau chỉ có thể là một trong 6 trường hợp sau đây: {Mặt - mặt; mặt - cạnh; mặt - đỉnh; cạnh - cạnh; cạnh - đỉnh; đỉnh - đỉnh}. Do vậy, có 15 trục phân cách tiềm năng như sau: {3 trục chỉ hướng của hộp bao thứ nhất; 3 trục chỉ hướng của hộp bao thứ hai; 9 trục tạo bởi tích có hướng của một trục thuộc hộp bao thứ nhất và một trục thuộc hộp bao thứ hai}. Thuật toán SAT có một số hạn chế về tốc độ tính toán và kém hiệu quả khi ngữ cảnh có nhiều va chạm cần được phát hiện.



Hình 1.7. Minh họa ý tưởng xác định mặt phẳng tách hai vật thể

Ganovelli và đồng sự [52] đã cải tiến thuật toán SAT ban đầu dựa trên kỹ thuật giảm chiều để nâng cao hiệu quả phát hiện va chạm, kết quả cho thấy một sự cải thiện đáng kể về hiệu suất đối với trường hợp va chạm của các khối tứ diện. Một thuật toán khác cho bài toán phát hiện va chạm là thuật toán GJK (Gilbert-Johnson-Keerthi) [53] cho phép tính toán khoảng cách giữa hai tập lồi nhưng yêu cầu phải có sẵn các hàm hỗ trợ hiệu quả cho các tập này.

Sau đó, thuật toán GJK đã được Ferreira cải tiến để nâng cao hiệu quả tính toán khoảng cách giữa các đối tượng [54].

Phương pháp khử Fourier-Motzkin được giới thiệu để áp dụng vào bài toán phát hiện va chạm lần đầu tiên bởi Ericson [51] và được Hornus đề cập đến trong bài báo nghiên cứu khảo sát về các kỹ thuật phát hiện va chạm [55]. Gần đây, Baillehache [56] đã đề xuất thuật toán FMB cài đặt bằng ngôn ngữ C để thực hiện phát hiện giao điểm của các cặp hình khối 2D/3D. Sau mỗi bước thực hiện phép khử, thuật toán FMB làm cho số biến giảm đi một, tuy nhiên số ràng buộc lại tăng lên khá lớn.

#### ***1.4.2. Hạn chế liên quan đến phát hiện va chạm của mô hình chất liệu vải***

Việc phát hiện va chạm của các mô hình vật thể dễ biến dạng (điển hình là mô hình chất liệu vải) được đánh giá là phức tạp hơn việc phát hiện va chạm của các mô hình vật thể rắn vì còn cần phải xem xét đến quá trình biến dạng và vấn đề tự va chạm. Đã có nhiều công bố cải tiến kỹ thuật phát hiện va chạm, hầu hết đều dựa trên cấu trúc BVH [3, 5, 6] hoặc dựa trên kỹ thuật tính toán hàm khoảng cách SDF [7]. Cấu trúc BVH và hàm SDF đã được áp dụng hiệu quả với các mô hình vật thể rắn. Đối với các mô hình vật thể biến dạng (như chất liệu vải) thì chi phí tính toán SDF cũng như duyệt và tái cấu trúc BVH là rất lớn, vấn đề đặt ra là cần thiết kế thuật toán có thể hoạt động tốt trên các cấu hình biến dạng khác nhau [8].

Curtis và cộng sự [9] đề xuất sử dụng tam giác đại diện để loại bỏ các phép kiểm tra lặp, mặc dù có tăng hiệu quả phát hiện va chạm, nhưng do tính ngẫu nhiên của thuật toán phân bố của nó, nên khó tích hợp với các thuật toán lọc khác. Sau đó, Tang và cộng sự [10] đã thực hiện một số cải tiến trên phương pháp của Curtis và cộng sự [9] để giảm hơn nữa số lượng các phép kiểm tra cơ bản, có khả năng mở rộng tốt, chi phí lưu trữ thấp. Trong khi đó, các phương pháp của Brochu [11], Tang [12] và Wang [13] có chi phí tính toán cao, yêu cầu rất nhiều tài nguyên của hệ thống.

Các hệ thống mô phỏng VR thường xuyên phải thực thi các tác vụ phát hiện va chạm. Quá trình phát hiện va chạm chiếm rất nhiều thời gian trong các mô phỏng có sự chuyển động, đồng thời cũng là phần khó và phức tạp về tính toán, thậm chí có thể gây ra tình trạng *ngheñ nút cổ chai* [8, 14]. Nhìn chung, có hai hướng giải pháp để tăng tốc phát hiện va chạm: hướng thứ nhất là sử dụng cấu trúc phân hệ vùng bao động [6, 8] và hướng thứ hai là sử dụng thuật toán lọc [15-18]. Tuy nhiên, trong những sự kiện va chạm mà các mô hình có mức độ biến dạng cao thì phương pháp sử dụng thuật toán lọc trong công bố của Du [15] và Tang [16] kém hiệu quả vì không xử lý được trường hợp lưới tam giác bị lật.

BVHs là một cấu trúc dữ liệu dạng cây được tạo thành trên cơ sở phân tích các đối tượng cần được kiểm tra va chạm. BVHs đóng vai trò quan trọng trong việc biểu diễn các vật thể và cho phép giải quyết nhiều vấn đề trong phát hiện va chạm. Với các mô hình vật thể biến dạng như vải thì BVHs cần phải được dựng lại sau mỗi biến dạng, giải pháp được đề xuất là áp dụng kỹ thuật tái cấu trúc, tuy nhiên chi phí để thực hiện tái cấu trúc là rất lớn.

Kỹ thuật phát hiện va chạm liên tục và phát hiện va chạm tăng cường: Nhiều thuật toán hiệu quả đã được cải tiến để phát hiện va chạm liên tục giữa các mô hình vật thể rắn, mô hình khớp nối và các mô hình vật thể biến dạng. Kỹ thuật phát hiện va chạm liên tục áp dụng phương pháp nội suy tuyến tính giữa các đỉnh của mô hình, thực hiện các phép kiểm tra cơ sở giữa các cặp tam giác và tính toán thời điểm tiếp xúc. Kỹ thuật phát hiện va chạm tăng cường: Kỹ thuật này phát hiện va chạm nhanh trong các ứng dụng tương tác, cách tiếp cận chính là thực hiện các tính toán va chạm tăng cường giữa các khung (frames), áp dụng cho các đa giác lồi. Một số thuật toán duy trì cấu trúc phân cấp để truy vấn nhanh hơn. Hầu hết các kỹ thuật này đã được sử dụng để tăng tốc kiểm tra va chạm giữa các mô hình vật thể rắn. Các thuật toán dựa trên sự kiện và động học được đề xuất cho các mô hình vật thể biến dạng.



Kỹ thuật phát hiện va chạm song song: Kỹ thuật này khai thác khả năng xử lý song song của CPU hoặc GPU để tăng tốc phát hiện va chạm. Thao tác chủ yếu để kiểm tra va chạm là phép duyệt trên cấu trúc cây BVTT (Bounding Volume Traversal Tree). Rao và cộng sự [19] đã chỉ ra rằng hiệu quả của phép duyệt theo chiều sâu song song phụ thuộc vào sơ đồ phân phối các tiến trình và cấu trúc hệ bao. Các kỹ thuật cân bằng tải khác nhau cũng được đề xuất trong [13]. Reinefeld [21] đã so sánh các chiến lược cân bằng tải của các phương pháp duyệt theo chiều sâu để tăng tốc phát hiện va chạm. Assarsson đã tăng tốc phát hiện va chạm giữa các mô hình vật thể rắn CAD nhanh hơn ba lần trên CPU 8 nhân. Du và cộng sự [66] đã đề xuất một phương pháp song song phân vùng tác vụ tĩnh để kiểm tra va chạm. Chen và cộng sự [6] đã chứng minh ưu điểm của việc sử dụng các kiến trúc CPU đa nhân để phát hiện va chạm và mô phỏng vật lý. Các thuật toán phát hiện va chạm trong mô phỏng chất liệu vải đã được đề xuất cho các kiến trúc bộ nhớ phân tán [27]. Các kỹ thuật đa luồng được sử dụng trong [26] để tăng tốc phát hiện va chạm cho mô phỏng chất liệu vải. Một kỹ thuật phân tách tác vụ được sử dụng trong [11] để tính toán trên kiến trúc CPU đa nhân.

Cấu trúc cây BVTT biểu diễn hệ thống phân cấp của các phép kiểm tra chồng lấp (testing for overlap) được thực hiện trong quá trình duyệt. BVTT được sử dụng trong thuật toán phát hiện va chạm. Mỗi nút trong BVTT đại diện cho một phép kiểm tra chồng lấp duy nhất giữa một cặp khối bao (BV) hoặc kiểm tra tự va chạm trên một trong các BV. BVTT không còn là cây nhị phân và một số nút bên trong của cây BVTT có thể có ba nút con.

Vùng kiểm tra BVTT: Vùng kiểm tra của BVTT là một tập hợp các nút bên trong và nút lá (nơi phép duyệt kết thúc trong khi thực hiện truy vấn va chạm trong một thời gian nhất định). Vùng kiểm tra BVTT phản ánh phép duyệt truy vấn va chạm.

Việc phát hiện va chạm trên các mô hình vải tập trung vào hai vấn đề chủ yếu là các phép kiểm tra cơ sở và tự va chạm. Các phép kiểm tra cơ sở:

Thực hiện 15 phép kiểm tra cơ sở giữa các mặt (Face), cạnh (Edge) và đỉnh (Vertex) của hai tam giác: Gồm 6 phép kiểm tra đỉnh-mặt (Vertex-Face) và 9 phép kiểm tra cạnh-cạnh (Edge-Edge) [18]. Tụ va chạm: Do chuyển động biến dạng và thay đổi hình dạng có thể dẫn đến tụ va chạm, khi đó đặc điểm của các tam giác liền kề trong lưới cần phải được xem xét.

Kỹ thuật tính toán song song: Tang và cộng sự [25] đã mô tả phương pháp phát hiện va chạm dựa trên tính toán phân cấp tuần tự bao gồm bốn pha: Cập nhật BVH; Tính toán và duyệt BVTT; Kiểm tra các cặp tam giác không liền kề; Kiểm tra các cặp tam giác liền kề. Một tập rút gọn là tập con của tập các mặt, cạnh và đỉnh thuộc các tam giác liền kề không được kiểm tra trong quá trình xử lý các cặp tam giác không liền kề. Bằng cách chỉ tiến hành kiểm tra trên tập rút gọn, số lượng phép kiểm tra cơ sở cho các tam giác liền kề có thể giảm xuống.

Việc cập nhật và xử lý cặp không liền kề của BVH có thể thực hiện song song trên nhiều nhân. Nhưng việc song song hóa tính toán BVTT tương đối khó khăn. Do đó, khó khăn đối với việc phát hiện va chạm song song là đưa ra các kỹ thuật song song thích hợp để duyệt BVTT và tính toán các cặp có khả năng chồng lấp [11,26,27]. Trong pha thứ hai cần duyệt qua BVTT theo chiều rộng, thu thập tất cả thông tin các tác vụ phụ truy vấn va chạm và đưa chúng vào một ngăn xếp. Sau đó, tất cả các tác vụ phụ trong ngăn xếp được thực thi song song trên các nhân của CPU. Tuy nhiên, cách tiếp cận PHC có điểm hạn chế là do tính chất động của các mô hình vải gây khó khăn trong việc phân phối tải của truy vấn va chạm giữa hai nút BVH.

#### ***1.4.3. Hạn chế liên quan đến phát hiện va chạm của cánh tay cobot***

Các phương pháp phát hiện va chạm trên cánh tay cobot được phân loại thành các nhóm: (1) nhóm phương pháp sử dụng học máy và (2) nhóm phương pháp không sử dụng học máy. Nhóm phương pháp (2) thường dựa vào các cảm biến bề mặt xúc giác bao phủ bên ngoài cobot [30, 31]. Tuy

nhiên, các cảm biến bề mặt xúc giác thường có chi phí cao, không bền và có thể bị hỏng khi bị va chạm mạnh hay va chạm nhiều lần. Các phương pháp dựa trên học máy [25, 27, 29] thường ước tính các mômen xoắn khớp bên ngoài do va chạm, sau đó so sánh với một tập giá trị ngưỡng cho trước để kiểm tra xem va chạm có xảy ra hay không. Đo mômen xoắn trực tiếp sẽ là lý tưởng, nhưng cảm biến mômen xoắn có thể tốn kém, giải pháp thay thế là ước tính mômen xoắn khớp từ các phép đo dòng điện tại các bộ truyền động khớp.

Hiện nay, các phương pháp phát hiện va chạm sử dụng học máy đã trở nên phổ biến. Với điều kiện thu thập đủ lượng dữ liệu đa dạng, chất lượng tốt, các phương pháp học máy có thể khắc phục được sự không chắc chắn (uncertainty) của các tham số trong mô hình động lực học cũng như các hiệu ứng không được mô hình hóa như ma sát và nhiễu đo lường.

A. N. Sharkawy và cộng sự [32] đã sử dụng M-FNN với sai số vị trí khớp và vận tốc làm đầu vào (input), đồng thời ước tính mômen xoắn khớp bên ngoài mà không cần sử dụng động lực học cobot (nhưng cũng yêu cầu điều chỉnh ngưỡng va chạm thủ công cho từng khớp), trong khi [33] xem xét các mômen khớp đo được và vectơ lực hấp dẫn của tay máy. Tuy nhiên, các phương pháp này chỉ được thực nghiệm với các chuyển động hai và ba khớp của một tay máy cobot 7-DoF. Tác giả Z. Zhang và các cộng sự [34] huấn luyện một FNN với các tín hiệu đầu vào 36 chiều được trích xuất từ cả các đặc điểm miền thời gian và tần số của các phép đo cảm biến mômen xoắn khớp. Đối với các cobot có hình dạng đầy đủ giống như con người, nhóm nghiên cứu của K. Narukawa [35] đề xuất cải tiến kỹ thuật máy vectơ hỗ trợ một lớp (OC-SVM), trong đó đầu vào được coi là vectơ điểm mômen của cobot hình người và các phép đo lực và mômen xoắn cho từng chi.

Y. J. Heo và các cộng sự [36] đề xuất phương pháp phát hiện nhanh va chạm của cánh tay cobot 6-DoF sử dụng mạng nơron tích chập một chiều (1D CNN) với kích thước là  $W \times 66$  (với  $W$  biểu thị kích thước cửa sổ thời gian) trong đó có các mômen xoắn khớp bên ngoài ước tính làm đầu vào; đầu ra là

kết quả phát hiện va chạm dạng nhị phân (*True, False*). Y. J. Heo và cộng sự đã thực hiện các thử nghiệm để đánh giá định lượng phương pháp phát hiện va chạm được đề xuất dựa trên học sâu. Đầu tiên, so sánh độ chính xác phát hiện của phương pháp được đề xuất với phương pháp tiêu chuẩn sử dụng bộ quan sát động lượng với các ngưỡng quyết định. Thứ hai, xác minh khả năng khái quát hóa của phương pháp bằng cách thực hiện các thử nghiệm trong đó áp dụng mạng được huấn luyện cho các khác chuyển động và cobot không được xem xét trong quá trình huấn luyện. Y. J. Heo và cộng sự gắn cảm biến FSR vào một liên kết cobot tùy ý và đánh ngẫu nhiên vào vị trí được gắn trong quá trình cobot vận hành.

Để xác thực, Y. J. Heo đã so sánh độ chính xác của phương pháp được đề xuất (CollisionNet) với phương pháp dựa trên động lượng (Momentum-based Observer Detector - MOD). Độ chính xác phát hiện của MOD phụ thuộc rất nhiều vào ngưỡng quyết định. Nếu ngưỡng được đặt là nhạy với va chạm trong một chuyển động cụ thể thì có thể tạo ra nhiều *dương tính giả* (False Positives - FP). Ngược lại, nếu ngưỡng không nhạy, độ trễ phát hiện tăng lên và thậm chí có thể phát sinh lỗi. CollisionNet được huấn luyện dựa trên dữ liệu thu được từ Indy-5 và Indy-7. Các tín hiệu khớp của Indy-5 kém ổn định hơn so với Indy-7.

K. M. Park và các cộng sự [27] tập trung nghiên cứu giải quyết vấn đề phát hiện va chạm của cánh tay cobot Doosan M0609. Các thuật toán phát hiện dựa trên học máy của Park (MCD-SVMR, MCD-CNN) được cài đặt bằng C++ trên bộ điều khiển cobot và thời gian tính toán trung bình được kiểm tra qua 10.000 lần lặp. Do cùng một tập tải trọng được sử dụng trong cả giai đoạn huấn luyện và thử nghiệm của các thuật toán nên kết quả thu được đối với trường hợp tải trọng ngẫu nhiên và không xác định có thể không khớp với kết quả được báo cáo trong kết quả kiểm tra thử nghiệm. Tuy nhiên, việc xác thực cho các tải trọng ngẫu nhiên vẫn còn hạn chế và là một trong những công việc rất khó khăn.

Trong khi đó, M. Czubenko và nhóm nghiên cứu [25] đề xuất kết hợp một số mạng nơron khác nhau như AR, RNN, CNN-LSTM và MC-LSTM nhằm phát hiện chính xác va chạm của cánh tay cobot CURA6. Czubenko phát hiện va chạm bằng cách sử dụng cảm biến lực ảo xử lý thông tin về dòng điện của động cơ với sự hỗ trợ của mạng nơron nhân tạo với bốn kiến trúc khác nhau. M. Czubenko đã chứng minh ConvLSTM là cấu trúc mạng nơron phù hợp hơn cả để mô hình hóa động lực học của cobot (sai số tuyệt đối trung bình của dự đoán khoảng 22,5 Nm). Tuy nhiên, hạn chế chung mà các phương pháp của K. M. Park và M. Czubenko gặp phải là sự phụ thuộc vào ma sát của động cơ, đặc biệt là ma sát Coulomb.

## 1.5. Chỉ số đánh giá

### 1.5.1. Độ đo đánh giá hiệu suất phát hiện va chạm

Độ đo để đánh giá hiệu suất phát hiện va chạm là *Precision* và *Recall* được định nghĩa như sau:

$$Precision = \frac{TP}{TP+FP} \quad (1.15)$$

và:

$$Recall = \frac{TP}{TP+FN} \quad (1.16)$$

với *TP* biểu thị số lượng dương tính thật, *FP* biểu thị số lượng dương tính giả và *FN* biểu thị số lượng âm tính giả (va chạm được coi là dương tính). *Precision* cao thể hiện tỉ lệ dương tính giả thấp trong khi *Recall* cao thể hiện tỉ lệ âm tính giả thấp. PRAUC (Area Under the Precision-Recall Curve) cao thể hiện cả *Precision* và *Recall* cao.

### 1.5.2. Tiêu chí đánh giá các loại khối bao

Mỗi vật thể 3D được tạo thành từ nhiều mặt, do đó chi phí để kiểm tra va chạm là rất cao. Hầu hết các hệ thống thực tại ảo sử dụng phương pháp phát hiện va chạm dựa trên các khối bao.

Bảng 1.3. thể hiện kết quả so sánh các loại khối bao phổ biến theo các tiêu chí đánh giá.

*Bảng 1.3.* Bảng so sánh các loại khối bao

<b>Loại khối bao</b>	<b>Tính đơn giản</b>	<b>Sử dụng bộ nhớ hiệu quả</b>	<b>Mức độ bó sát (tight-fitting)</b>	<b>Phù hợp cho mô hình vải</b>
SBB	2	1	4	3
AABB	1	2	3	1
OOBB	4	3	2	4
k-DOP	3	4	1	2

Có bốn tiêu chí đánh giá trong bảng trên và mỗi tiêu chí đánh giá được chia làm bốn mức độ được đánh số từ 1 đến 4 (quy ước giá trị số 1 là tốt nhất, giá trị số càng cao thì mức độ đáp ứng càng thấp). Việc lựa chọn loại khối bao được xác định bởi nhiều yếu tố như: chi phí xây dựng khối bao, chi phí cập nhật lại khối bao khi đối tượng di chuyển hoặc thay đổi hình dạng/kích thước, chi phí xác định điểm va chạm và mức độ chính xác của phương pháp phát hiện va chạm,... NCS lựa chọn sử dụng khối bao k-DOP để thỏa hiệp giữa các tiêu chí đánh giá và thuận lợi cho việc biểu diễn cũng như hiệu quả tính toán phát hiện va chạm của các mô hình vải. Khối bao k-DOP cũng được sử dụng trong các nghiên cứu của Curtis [9], Du [15, 66], Tang [10, 12, 16, 67] và Zhang [17, 18].

## 1.6. Kết luận Chương 1

Trong Chương 1, NCS trình bày tổng quan chung về công nghệ thực tại ảo và bài toán phát hiện va chạm trong môi trường ảo; tổng quan về cobot, phát hiện va chạm của cánh tay cobot và các nghiên cứu liên quan. Việc nghiên cứu kỹ thuật tăng tốc độ xử lý của hệ thống thực tại ảo là rất cần thiết, trong đó tập trung vào cải tiến các kỹ thuật tính toán phát hiện va chạm vì nó đóng một vai trò quan trọng trong hệ thống mô phỏng tương tác. Chương 1

cũng giới thiệu khái quát hai hướng tiếp cận phát hiện va chạm của cánh tay cobot: (1) dựa trên học máy và (2) sử dụng cảm biến.

Trong quá trình tìm hiểu và viết luận án, NCS đã trình bày khái quát về hệ thống thực tại ảo và kỹ thuật phát hiện va chạm trong môi trường ảo với mục đích phát triển phương pháp phát hiện va chạm trong môi trường ảo. Thông qua đó, NCS đã tìm hiểu, cập nhật được một số kiến thức mới liên quan trực tiếp đến đề tài phục vụ cho quá trình thực hiện luận án.

Hiện nay, các nghiên cứu về bài toán phát hiện va chạm của cánh tay cobot 6 bậc tự do được nhiều nhà khoa học quan tâm nhằm cải tiến, nâng cao chất lượng và hiệu quả phát hiện va chạm dựa trên học máy. Đây cũng chính là các hướng nghiên cứu và phát triển đề tài mà NCS quan tâm.

Trên cơ sở nền tảng của chương 1, trong chương 2, NCS tập trung vào trình bày về giải pháp và đề xuất cải tiến các thuật toán lọc trong việc phát hiện va chạm của chất liệu vải tương tác trong môi trường thực tại ảo.

Phát triển từ kiến thức nền tảng của chương 1, NCS trình bày trong chương 3 giải pháp và đề xuất cải tiến kỹ thuật phát hiện va chạm của cánh tay cobot CURA6.

Một phần kết quả nghiên cứu trình bày tại chương 1 đã được NCS công bố tại [CT1, CT2, CT3] trong Danh mục công trình công bố liên quan đến luận án.

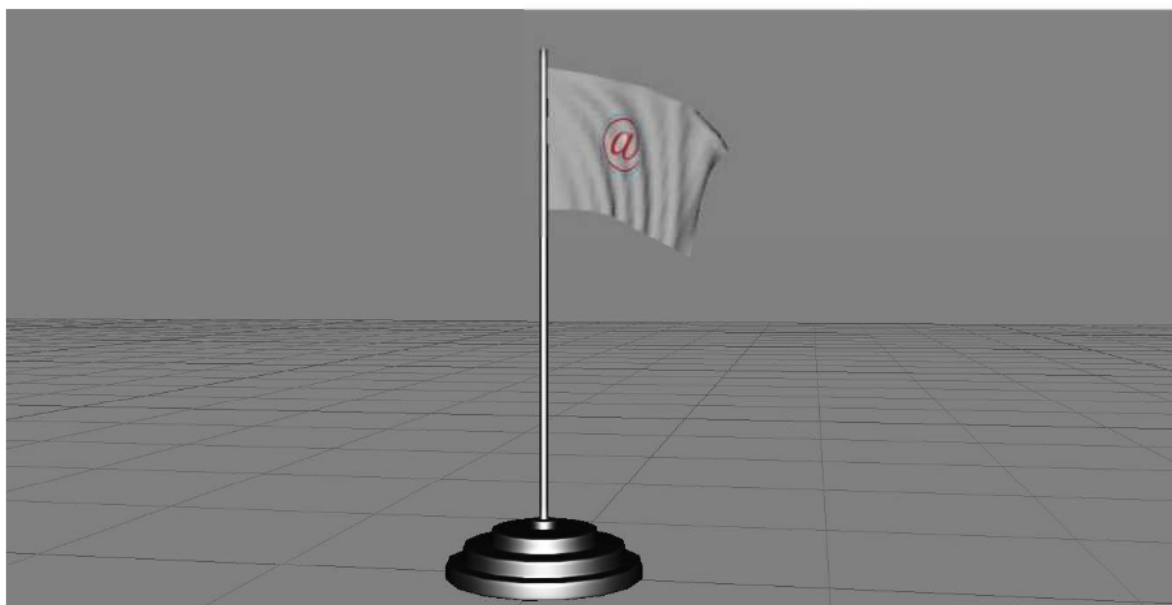
## CHƯƠNG 2. PHƯƠNG PHÁP PHÁT HIỆN VA CHẠM TRONG VẬT THỂ BIẾN DẠNG

Va chạm là một trong những vấn đề cơ bản cần phải xử lý trong các hệ thống đồ họa máy tính và tương tác thực tại ảo. Phát hiện va chạm là khâu mở đầu, mang tính chất quyết định đến các tác vụ xử lý va chạm của hệ thống. Trong chương 2, NCS trình bày kỹ thuật cải tiến để phát hiện nhanh va chạm của mô hình chất liệu vải và mô hình vật thể rắn.

### 2.1. Đề xuất phương pháp nâng cao hiệu quả phát hiện va chạm của mô hình chất liệu vải

#### 2.1.1. Phương pháp đề xuất

Các đối tượng 3D trong hệ thống mô phỏng có thể là các mô hình vật thể rắn hoặc mô hình vật thể biến dạng, trong đó vải là một trong những mô hình vật thể biến dạng điển hình. NCS tập trung vào vấn đề phát hiện va chạm của mô hình vải được biểu diễn dưới dạng mặt tam giác. Hình 2.1 minh họa quá trình va chạm và tự va chạm của vải trên một lá cờ có hình biểu trưng của Hội thảo khoa học Quốc gia “Một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông” (VNICT) đang tung bay trong gió.



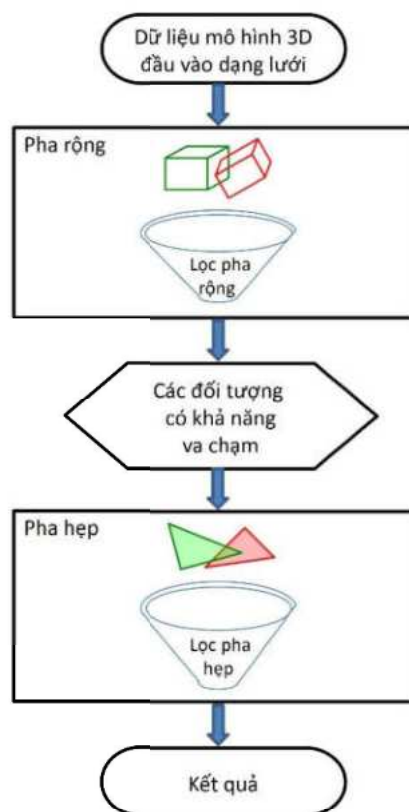
Hình 2.1. Mô phỏng một lá cờ đang tung bay trong gió



### 2.1.1.1. Ý tưởng thuật toán

#### Mô tả ý tưởng

NCS đề xuất thuật toán phát hiện va chạm của mô hình chất liệu vải dựa trên phương pháp lọc hai pha (*Collision Detection Algorithm Based on Two-Phase Filtering - AB2PF*) [CT4] trên cơ sở kế thừa và phát triển giải pháp của Tang [10] công bố năm 2009 và Zhang [18] công bố năm 2015 vì tính khả thi và phù hợp đối với vật thể chất liệu vải. Kết quả thử nghiệm cho thấy thuật toán đề xuất hiệu quả ngay cả khi mô hình vải bị biến dạng rất lớn. Quá trình phát hiện va chạm được chia thành hai pha là *pha rộng* và *pha hẹp* như minh họa trong Hình 2.2.



Hình 2.2. Sơ đồ tổng quát thuật toán đề xuất

Trong đó, pha rộng là quá trình kiểm tra va chạm giữa các hộp bao, pha hẹp là quá trình kiểm tra va chạm giữa các mắt lưới tam giác. Pha rộng loại trừ các cặp đối tượng không thể giao nhau, kết quả của pha này là chọn ra được các đối tượng có khả năng va chạm. Đầu ra của pha rộng chính là đầu

vào của pha hẹp. Pha hẹp tính toán để xác định chính xác những đối tượng thực sự va chạm.

NCS đề xuất một kỹ thuật tăng tốc phát hiện va chạm cho mô hình vải dựa trên các thuật toán *lọc pha rộng* và *lọc pha hẹp* minh họa trong Hình 2.2. Thuật toán đề xuất loại bỏ các cặp đối tượng không có khả năng xảy ra va chạm và tăng tốc phát hiện va chạm.

Phân hệ vùng bao BVH là một cấu trúc dữ liệu dạng cây được tạo thành trên cơ sở phân tích các đối tượng cần được kiểm tra va chạm. BVH đóng vai trò quan trọng trong việc biểu diễn các vật thể và phát hiện va chạm.

Cấu trúc BVTT (Bounding Volume Traversal Tree) biểu diễn hệ thống phân cấp của các phép kiểm tra chồng lấp (overlap tests) được thực hiện trong quá trình duyệt BVH. Mỗi nút trong BVTT đại diện cho một phép kiểm tra chồng lấp duy nhất giữa một cặp khối bao.

Công đoạn đầu tiên là khởi tạo mô hình 3D đầu vào và sử dụng phương pháp ICCD (Interactive Continuous Collision Detection) của Tang [10] để đánh dấu nhằm đảm bảo rằng việc kiểm tra cặp đối tượng cơ sở chỉ được thực hiện một lần. Sau đó, các BV được xây dựng. Tùy theo các BV có chồng lấp nhau hay không, có thể loại trừ một số cặp tam giác và các cặp đối tượng cơ sở không có khả năng va chạm.

Chuyển sang công đoạn kế tiếp, phương pháp lọc dựa trên tính chất hình học của Tang [16] được sử dụng để xác định tính đồng phẳng, và sau đó phương pháp lọc dựa vào phép toán đại số của Zhang [18] được sử dụng để xác định sự tồn tại nghiệm của phương trình khoảng cách.

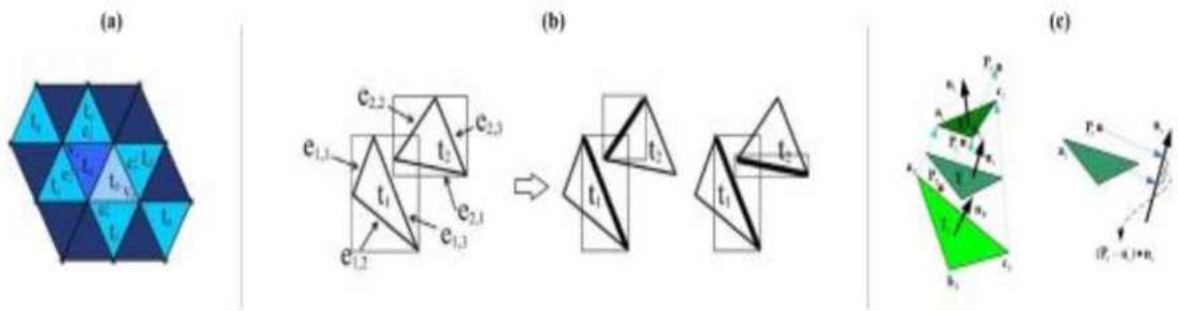
### *Lọc pha rộng*

Việc phát hiện va chạm trên các mô hình vải tập trung vào hai vấn đề chủ yếu là các phép kiểm tra cơ sở và tự va chạm.

*Các phép kiểm tra cơ sở:* Thực hiện 15 phép kiểm tra cơ sở giữa các mặt (Face), cạnh (Edge) và đỉnh (Vertex) của hai tam giác, gồm 6 phép kiểm tra đỉnh-mặt (Vertex-Face ghi tắt là VF) và 9 phép kiểm tra cạnh-cạnh (Edge-Edge ghi tắt là EE).

*Tự va chạm:* Do chuyển động biến dạng và thay đổi hình dạng có thể dẫn đến tự va chạm.

Mỗi nút trong BVTT đại diện cho một phép kiểm tra chồng lấp giữa một cặp BV hoặc đại diện cho một phép kiểm tra tự va chạm. Các đối tượng được biểu diễn bằng lưới tam giác. Bề mặt lưới cung cấp rất nhiều thông tin hữu ích có thể được sử dụng để tăng tốc phát hiện va chạm. Trong Hình 2.3a, nếu muốn kiểm tra xem hai tam giác  $t_a$  và  $t_b$  có cắt nhau trong thời gian  $\Delta T$  hay không, các phương pháp phát hiện va chạm truyền thống cần tiến hành giải 15 phương trình bậc ba, trong đó gồm 6 phép kiểm tra VF và 9 phép kiểm tra EE. Phương pháp như vậy không hiệu quả với các mô hình có mức độ biến dạng lớn và cần được cải tiến bằng một phương pháp mới. Do  $t_a$  và  $t_b$  là kề nhau nên chỉ cần đến 4 phép kiểm tra là:  $(v_a, t_b)$ ,  $(v_b, t_a)$ ,  $(e_a^1, e_b^1)$  và  $(e_a^2, e_b^2)$ . Bởi vì việc phát hiện va chạm giữa các tam giác liên kề dễ dàng hơn so với các tam giác không liên kề, toàn bộ quá trình phát hiện va chạm được chia thành hai bước. Bước đầu tiên kiểm tra các tam giác không liên kề và bước thứ hai kiểm tra các tam giác liên kề. Vì vậy, bước thứ hai sẽ tận dụng lợi thế này. Trong quá trình kiểm tra không liên kề,  $(t_g, t_b)$  và  $(v_a, t_b)$  được kiểm tra đồng thời. Tương tự,  $(t_a, t_h)$  và  $(v_b, t_a)$  được kiểm tra đồng thời,  $(t_e, t_f)$  và  $(e_a^1, e_b^1)$  được kiểm tra đồng thời,  $(t_c, t_d)$  và  $(e_a^2, e_b^2)$  được kiểm tra đồng thời. Vì vậy, trong tình huống này nếu phép kiểm tra các tam giác không liên kề được thực hiện trước thì có thể tránh được tất cả các phép kiểm tra các tam giác liên kề.



Hình 2.3. Các phép kiểm tra sơ cấp đối với lưới tam giác

Để cải thiện hiệu quả lọc, BVH được xây dựng từ ba dạng BV: BV đỉnh (V-BV), BV cạnh (E-BV) và BV mặt (F-BV) [9]. Trong hình 2.3b, BV của hai tam giác chồng lên nhau, về mặt lý thuyết thì cần tiến hành 15 phép kiểm tra sơ cấp. Nhưng trên thực tế, nếu sử dụng các E-BV thì chỉ cần 2 phép kiểm tra sơ cấp.

Trong Hình 2.3c, tại thời điểm  $t_0$  thì tam giác ở vị trí thấp hơn và tại thời điểm  $t_1$  thì tam giác chuyển động đến vị trí cao hơn. Về mặt lý thuyết thì có thể sử dụng một BV lớn để bao phủ cả hai vị trí, dẫn đến việc phải tiến hành giải nhiều phương trình bậc ba để kiểm tra xem trong thời gian  $\Delta T$ , đỉnh có chạm vào tam giác hay không. Trong tình huống này, kỹ thuật lọc không thâm nhập là một cách hiệu quả và dễ dàng hơn nhiều để thực hiện phép kiểm tra này. Giả sử pháp tuyến của tam giác thay đổi từ  $\vec{n}_0$  thành  $\vec{n}_1$ , và  $\vec{n}_t$  là pháp tuyến tại thời điểm bất kỳ. Chiếu đỉnh  $a$  và  $p$  lên pháp tuyến  $\vec{n}_t$ , nếu trong thời gian  $\Delta T$  khoảng cách chiếu luôn khác không thì đỉnh và tam giác không thể tiếp xúc nhau. Để xác định được khoảng cách này cũng cần sử dụng đến các phép tính toán, tuy nhiên việc tính toán đơn giản hơn nhiều so với việc tính toán giải các phương trình bậc ba.

### Lọc pha hẹp

Giả thiết rằng các đỉnh của tam giác chuyển động với vận tốc không đổi trong khoảng thời gian  $t \in [0,1]$ . Phát hiện va chạm liên tục giữa hai tam giác chuyển động/biến dạng bao gồm một tập các phép kiểm tra VF hoặc EE. Cho

bốn điểm  $\vec{x}_i(t)$  và vận tốc không đổi của chúng  $\vec{v}_i(t)$  với  $i = (1, 2, 3, 4)$ , hàm khoảng cách  $d(t)$  giữa đỉnh  $\vec{x}_4$  và tam giác  $\vec{x}_1\vec{x}_2\vec{x}_3$  hoặc giữa cạnh  $\vec{x}_1\vec{x}_2$  với cạnh  $\vec{x}_3\vec{x}_4$  được xác định như sau:

$$d(t) = [\vec{x}_2(t) - \vec{x}_1(t)] \times [\vec{x}_3(t) - \vec{x}_1(t)] \cdot [\vec{x}_4(t) - \vec{x}_1(t)] \quad (2.4)$$

trong đó  $t \in [0,1]$ ,  $\vec{v}_i = \vec{x}_i(1) - \vec{x}_i(0)$ ,  $\vec{x}_i(t) = \vec{x}_i(0) + t\vec{v}_i$ , ký hiệu toán tử " $\times$ " biểu diễn phép tích chéo, ký hiệu toán tử " $\cdot$ " biểu diễn phép tích vô hướng. Để đơn giản, ta đặt  $\vec{x}_i = \vec{x}_i(0)$ , do đó công thức (2.1) có thể biểu diễn lại dưới dạng:

$$d(t) = [\vec{x}_{21} + t\vec{v}_{21}] \times [\vec{x}_{31} + t\vec{v}_{31}] \cdot [\vec{x}_{41} + t\vec{v}_{41}] \quad (2.5)$$

trong đó sử dụng cách viết ngắn gọn  $\vec{x}_{ij}$  và  $\vec{v}_{ij}$  để biểu thị  $\vec{x}_i - \vec{x}_j$  và  $\vec{v}_i - \vec{v}_j$  tương ứng. Sau khi sắp xếp lại các thành phần thu được  $d(t)$  dưới dạng một đa thức bậc ba như sau:

$$d(t) = a_3t^3 + a_2t^2 + a_1t + a_0 \quad (2.6)$$

trong đó

$$a_3 = \vec{v}_{21} \times \vec{v}_{31} \cdot \vec{v}_{41}$$

$$a_2 = \vec{v}_{21} \times \vec{v}_{31} \cdot \vec{x}_{41} + (\vec{v}_{21} \times \vec{x}_{31} + \vec{x}_{21} \times \vec{v}_{31}) \cdot \vec{v}_{41}$$

$$a_1 = (\vec{v}_{21} \times \vec{x}_{31} + \vec{x}_{21} \times \vec{v}_{31}) \cdot \vec{x}_{41} + \vec{x}_{21} \times \vec{x}_{31} \cdot \vec{v}_{41}$$

$$a_0 = \vec{x}_{21} \times \vec{x}_{31} \cdot \vec{x}_{41}$$

Bốn điểm đồng phẳng (tức là  $d(t) = 0$ ) là điều kiện cần thiết để xảy ra va chạm giữa hai tam giác. Do đó, một phương trình bậc ba có thể được suy ra. Nghiệm  $t \in [0,1]$  của phương trình  $d(t) = 0$  sẽ là thời gian tiếp xúc của hai tam giác. Trong thuật toán, va chạm được phát hiện khi khoảng cách nhỏ hơn ngưỡng cho trước. Nếu không tồn tại  $t \in [0,1]$  thì không có va chạm giữa hai tam giác trong khoảng thời gian  $[0,1]$ .

NCS sử dụng kết hợp thuật toán lọc dựa trên các phép toán đại số và thuật toán lọc dựa trên các đặc điểm hình học của Tang [16] liên quan đến tính đồng phẳng của các cặp VF, EE.

Từ Bảng 1.1 trong chương 1, có thể thấy rằng việc sử dụng Định lý Vincent cho hàm khoảng cách  $d(t)$  có thể nhanh chóng xác định sự tồn tại của nghiệm của phương trình khoảng cách. Khi  $VAR(g(t))$  bằng 1 hoặc 3, có thể đánh giá rằng hàm khoảng cách có nghiệm. Trong trường hợp này, phương trình khoảng cách cần được tính toán chính xác. Nếu  $VAR(g(t))$  bằng 0 thì phương trình không có nghiệm nguyên trên  $[0,1]$ .  $VAR(g(t))$  bằng 2 thì sự tồn tại của các nghiệm là không chắc chắn. Lúc này, nghiệm có thể không tồn tại hoặc có thể có 2 nghiệm. Trong trường hợp này,  $g(t)$  cần được phân tích thêm. Đầu tiên, phân tích  $g(t)$  thành  $g_1(t)$  và  $g_2(t)$  như trong công thức dưới đây. Hệ số  $b_i$  với  $i = (1, 2, 3, 4)$  trong công thức (2.4) có thể được tính bằng  $a_i$  với  $i = (1, 2, 3, 4)$  trong công thức (2.3).

$$g(t) = (t + 1)^n f\left(\frac{1}{t+1}\right) = b_3 t^3 + b_2 t^2 + b_1 t + b_0 \quad (2.7)$$

trong đó

$$b_3 = a_0$$

$$b_2 = a_1 + 3a_0$$

$$b_1 = a_2 + 2a_1 + 3a_0$$

$$b_0 = a_3 + a_2 + a_1 + a_0$$

và

$$g_1(t) = g(t + 1) = b_3 t^3 + (b_2 + 3b_3)t^2 + (b_1 + 2b_2 + 3b_3)t + (b_0 + b_1 + b_2 + b_3) \quad (2.8)$$

và

$$g_2(t) = (t+1)^3 g\left(\frac{1}{t+1}\right) = b_0 t^3 + (b_1 + 3b_0)t^2 + (b_2 + 2b_1 + 3b_0)t + (b_0 + b_1 + b_2 + b_3) \quad (2.9)$$

Nếu  $VAR(g_1(t)) = 0$  thì  $f(t)$  (tức là  $d(t)$ ) không có nghiệm trên  $\left[0, \frac{1}{2}\right]$ .

Nếu  $VAR(g_2(t)) = 0$  thì  $f(t)$  không có nghiệm nào trên  $\left[\frac{1}{2}, 1\right]$ .

Nếu  $VAR(g(t)) = 2$  và  $VAR(g_1(t)) = 0$  và  $VAR(g_2(t)) = 0$  thì ta có thể kết luận rằng  $d(t)$  không có nghiệm nào trên  $[0,1]$ . Do đó có thể tránh được việc giải một số lượng lớn các phương trình bậc ba.

### 2.1.1.2. Mô tả thuật toán AB2PF

Thuật toán AB2PF pha rộng được mô tả như sau:

---

#### **Algorithm1**

---

**Đầu vào:** Mô hình 3D được biểu diễn dạng lưới tam giác.

**Đầu ra:** Các cặp đối tượng có khả năng xảy ra va chạm.

---

```

1: BV ← CreateBV(Vi);
2: BVH ← CreateBVH(BV, top);
3: BVTT ← TraversalTree(BVH);
4: for node i in BVTT do
    if isLeaf(Nai) AND isLeaf(Nbi) then
        if testOverlap(Nai, Nbi) goto Algorithm2(Nai, Nbi);
        else return exit;
        end if
    end if
end for

```

---

Pha rộng loại trừ các cặp đối tượng không thể giao nhau (không có khả năng xảy ra va chạm), kết quả của pha này là chọn lọc ra được các đối tượng có khả năng va chạm (nếu có). Thuật toán pha rộng Algorithm1 thực hiện tính

toán tại mỗi frame để trả về các cặp đối tượng có khả năng xảy ra va chạm với độ phức tạp thuật toán là  $O(n \log_2 n)$ .

Thuật toán AB2PF pha hẹp được mô tả như sau:

---

**Algorithm2**

---

**Đầu vào:** Tọa độ 04 đỉnh của cặp VF hoặc cặp EE cần kiểm tra va chạm.

**Đầu ra:** Kết luận va chạm/không va chạm.

---

```

1:  $f \leftarrow \text{testCo}(VF, EE);$ 
2:  $nv \leftarrow \text{VAR}(f, 0, 1);$ 
3: if  $nv=0$  then
    return false;
else
    return true;
end if

```

---

Pha hẹp tính toán để xác định chính xác có xảy ra sự va chạm hay không. Việc đầu tiên là thuật toán pha hẹp Algorithm2 tiến hành kiểm tra tính đồng phẳng của các cặp VF (hoặc EE). Sau đó áp dụng phương pháp đại số để xác định sự tồn tại hay không tồn tại nghiệm của phương trình khoảng cách và đưa ra các kết luận tương ứng. Độ phức tạp thuật toán là  $O(n)$ .

### 2.1.2. Kết quả và thảo luận

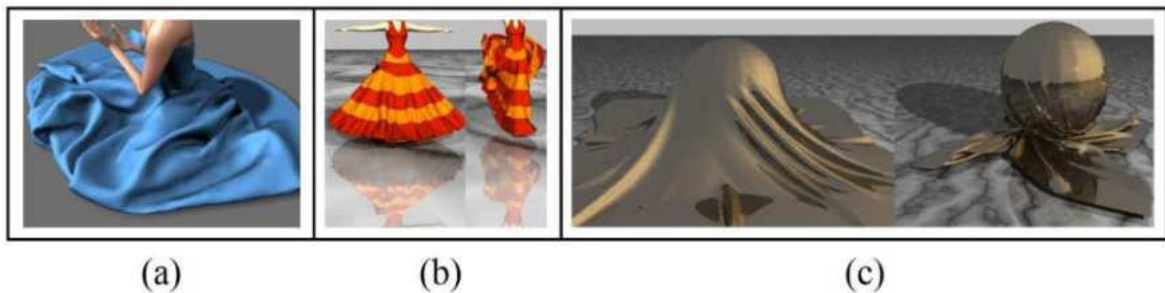
#### 2.1.2.1. Môi trường thử nghiệm

Môi trường thử nghiệm là máy tính CPU Intel (R) Core (TM) i7-7820HQ @ 2,90 GHz, sử dụng ngôn ngữ lập trình C++ trên hệ điều hành Windows và các tập dữ liệu thử nghiệm của Nhóm nghiên cứu UMD GAMMA thuộc Trường Đại học Maryland và hãng Walt Disney, Hoa Kỳ.

#### 2.1.2.2. Dữ liệu thử nghiệm

Ba bộ dữ liệu được sử dụng để đánh giá hiệu suất của các thuật toán bao gồm:





Hình 2.4. Các bộ dữ liệu trong thư viện mở GAMMA: (a) Bộ dữ liệu Princess, (b) Bộ dữ liệu Flamenco, (c) Bộ dữ liệu Cloth-ball

Princess (Hình 2.4a): Mô phỏng một cô gái mặc chiếc váy dài thực hiện động tác ngồi xuống và đứng lên nhẹ nhàng. Mô hình chiếc váy gồm 40.000 tam giác.

Flamenco (Hình 2.4b): Mô phỏng một cô gái mặc chiếc váy xòe có 07 lớp diềm vải sặc sỡ thực hiện các động tác múa theo vũ điệu Flamenco. Bộ dữ liệu này có số lượng vải tự va chạm rất lớn. Mô hình chiếc váy gồm 49.000 tam giác.

Cloth-ball (Hình 2.4c): Mô phỏng một mảnh vải (gồm 92.230 tam giác) phủ trên đỉnh của quả cầu và khi quả cầu cuộn tròn cuộn theo mảnh vải dẫn đến có nhiều phần vải tự va chạm.

Bảng 2.1. Bảng mô tả các bộ dữ liệu

Bộ dữ liệu	Số tam giác	Số đỉnh	Số cạnh	Số frames
Princess	40.000	20.000	60.000	464
Flamenco	49.000	26.000	75.000	352
Cloth-ball	92.230	46.598	138.825	94

Mỗi bộ dữ liệu có nhiều frames được mô tả thông tin chi tiết trong Bảng 2.1. Từ dữ liệu về các mô hình, có thể thấy rằng nếu tất cả các đỉnh, mặt, cạnh được kiểm tra theo từng cặp theo cách “vét cạn” mà không sử dụng các thuật toán lọc thì lượng tính toán là rất lớn.

### 2.1.2.3. Kết quả thử nghiệm, so sánh và đánh giá

Kết quả trong Bảng 2.2 cho thấy hiệu quả của phương pháp đề xuất. So sánh cho thấy mô hình Flamenco có hiệu quả loại bỏ ở mức cao. Các thuật toán lọc hai pha có thể phát huy đặc tính lọc tốt khi mô hình vải có mức độ biến dạng rất lớn.

Kết quả thử nghiệm phương pháp đề xuất được so sánh với các phương pháp của Tang [10, 16] và Zhang [17, 18] với đơn vị thời gian phát hiện va chạm là *ms*.

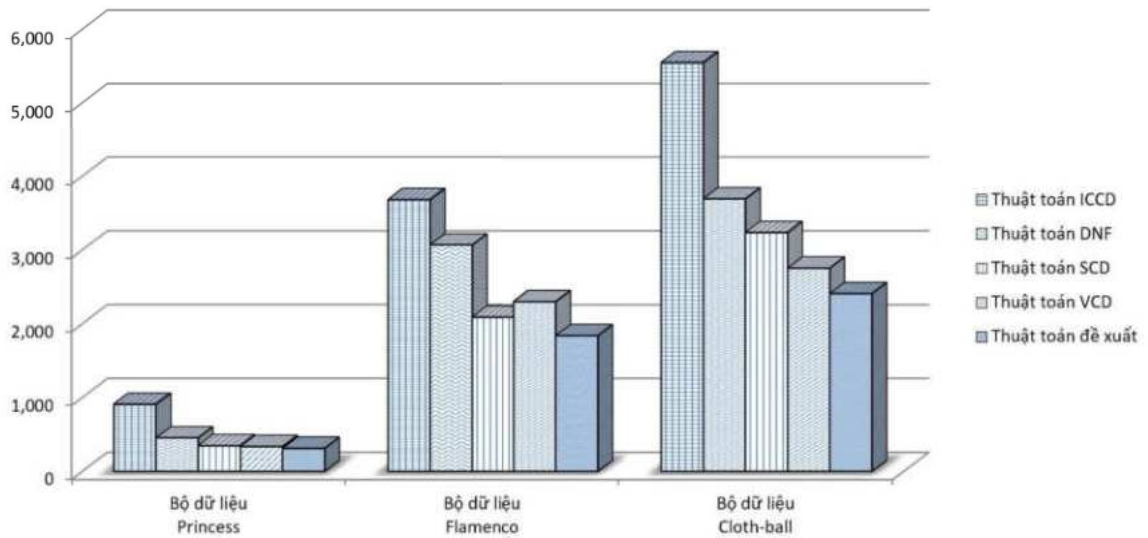
*Bảng 2.2.* Bảng thời gian phát hiện va chạm trung bình của các thuật toán

<b>Bộ dữ liệu</b>	<b>Thuật toán ICCD của Tang [10]</b>	<b>Thuật toán DNF của Tang [16]</b>	<b>Thuật toán SCD của Zhang [17]</b>	<b>Thuật toán VCD của Zhang [18]</b>	<b>Thuật toán đề xuất</b>
Princess	910	455	347	337	<b>313</b>
Flamenco	3.683	3.069	2.090	2.302	<b>1.842</b>
Cloth-ball	5.542	3.695	3.238	2.756	<b>2.411</b>

Phương pháp đề xuất có hai ưu điểm nổi bật:

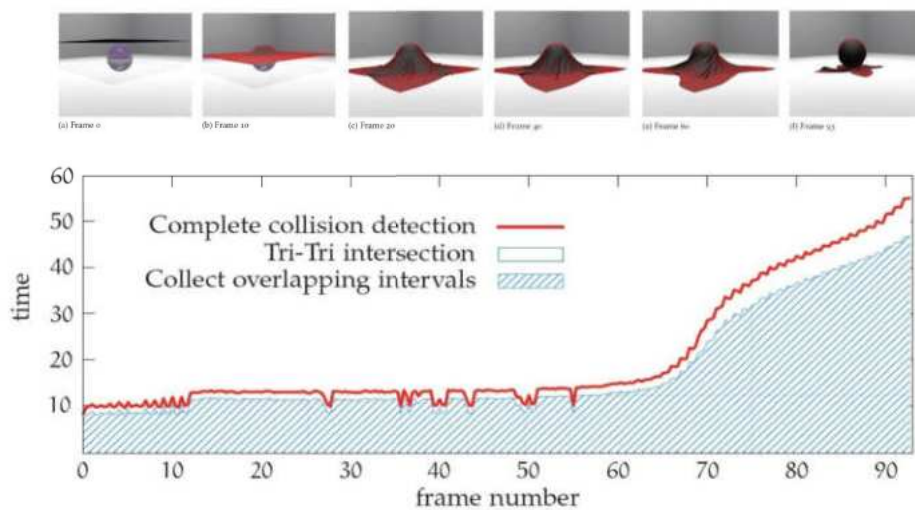
*Thứ nhất* là thời gian thực hiện kiểm tra chồng lấp khối bao và duyệt BVH được giảm đáng kể đối với các mô hình vải trong bộ dữ liệu thử nghiệm vì loại bỏ sớm các đối tượng không có khả năng va chạm.

*Thứ hai* là chiến lược lọc hai pha hiệu quả giảm được dương tính giả và tăng tốc phát hiện va chạm.



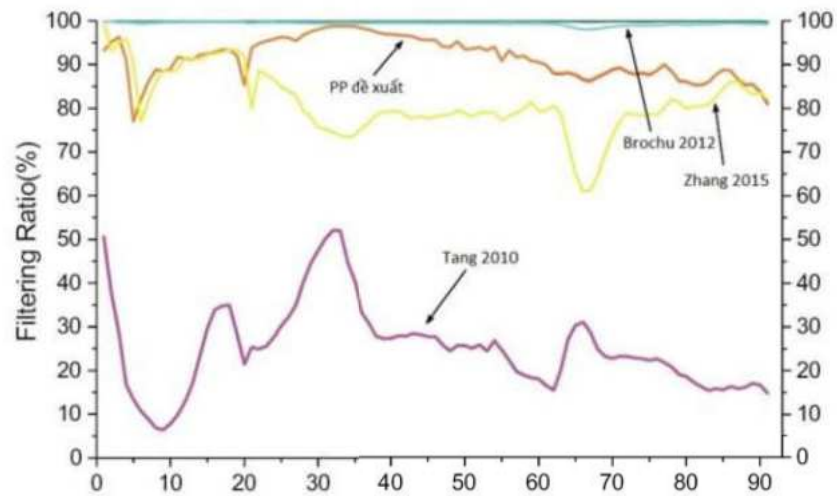
Hình 2.5. Kết quả thực thi các thuật toán

Biểu đồ so sánh trong Hình 2.5 cho thấy phương pháp đề xuất có kết quả chạy nhanh hơn. Lý do là NCS sử dụng kết hợp cả yếu tố hình học và đại số để loại bỏ các cặp VF và các cặp EE không va chạm, tiến hành đánh giá sự tồn tại nghiệm của phương trình khoảng cách, tiết kiệm một phần thời gian cho các thao tác giải phương trình, giảm số lượng phép toán phức tạp.



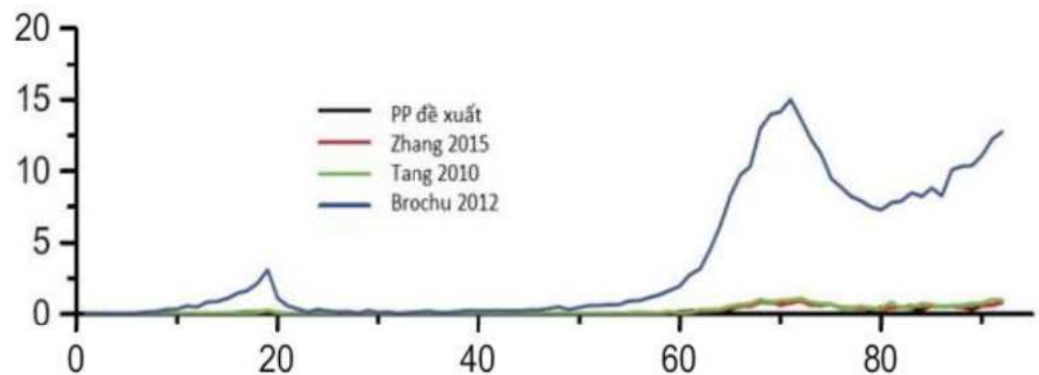
Hình 2.6. Phát hiện và va chạm tại các frames trong bộ dữ liệu Cloth-ball

Hình 2.6 cho thấy quá trình phát hiện và va chạm tại các frames trong bộ dữ liệu Cloth-ball, từ frame 60 trở đi số lượng va chạm tăng rất nhanh.



Hình 2.7. Hiệu suất lọc của các thuật toán

Minh họa trong Hình 2.7 là biểu đồ so sánh về độ chính xác (hiệu suất lọc) của các thuật toán phát hiện và chạm trên bộ dữ liệu Cloth-Ball.



Hình 2.8. So sánh chi phí thực thi các thuật toán

Biểu đồ trong Hình 2.8 so sánh chi phí tính toán của các thuật toán phát hiện và chạm trên bộ dữ liệu Cloth-Ball.

*Hạn chế:* Thuật toán đề xuất chỉ áp dụng cho trường hợp cấu trúc lưới tương đối ổn định, nếu mô hình bị rách hoặc vỡ trong quá trình va chạm thì phương pháp chưa xử lý hiệu quả. Ngoài ra, năng lực tính toán của GPU đã được tăng lên đáng kể trong những năm gần đây, để có thể phát hiện và chạm theo thời gian thực ngày càng cao thì phương pháp đề xuất cần mở rộng cách tiếp cận với GPU.

## 2.2. Đề xuất phương pháp nâng cao hiệu quả phát hiện va chạm của mô hình vật thể rắn

### 2.2.1. Phương pháp đề xuất

Trong phần này, NCS đề xuất thuật toán phát hiện va chạm của mô hình vật thể rắn dựa trên phương pháp khử Fourier-Motzkin (*Collision Detection Algorithm Based on Fourier-Motzkin Elimination - ABFME*) [CT5]. Các thuật toán truyền thống thường được sử dụng trong trường hợp này là SAT (Separating Axis Theorem) [50] dựa trên Định lý về trục phân tách nhưng tốc độ phát hiện va chạm còn khá chậm. Một cách tiếp cận khác là biểu diễn bài toán phát hiện va chạm dưới dạng một hệ bất phương trình tuyến tính, theo đó, NCS cải tiến phương pháp khử Fourier-Motzkin để tăng tốc phát hiện va chạm. Kết quả thử nghiệm cho thấy thuật toán đề xuất phát hiện va chạm nhanh hơn thuật toán SAT khoảng 4,17 lần.

Với phương pháp phát hiện va chạm gần đúng, các đối tượng được bao quanh bởi các khối đa diện lồi được gọi là khối bao. Việc phát hiện va chạm giữa các khối bao đa diện lồi không chỉ là một bài toán hình học đơn thuần mà còn có thể xem như một bài toán tối ưu. Coi mỗi khối đa diện lồi là thể tích giao nhau của các nửa không gian (mỗi nửa không gian được biểu diễn bằng một bất phương trình tuyến tính), việc kiểm tra va chạm của khối đa diện tương đương với một bài toán quy hoạch tuyến tính.

Kế thừa và phát triển từ thuật toán FMB của Baillehache, NCS nghiên cứu cải tiến phương pháp khử Fourier-Motzkin theo hướng áp dụng Định lý của Imbert [57] để tăng tốc phát hiện va chạm của các mô hình 3D. Thuật toán đề xuất của NCS được trình bày chi tiết trong phần 2.2.1.1 dưới đây.

#### 2.2.1.1. Ý tưởng thuật toán

Việc kiểm tra xem hai đa giác lồi có giao nhau hay không có thể được phát biểu như một bài toán quy hoạch tuyến tính (Linear Programming - LP). Hai đa giác lồi không giao nhau khi và chỉ khi giữa chúng tồn tại một mặt

phẳng phân cách. Các hệ số của phương trình mặt phẳng tách được coi là ẩn số. Các ràng buộc tuyến tính được hình thành bằng cách áp đặt rằng tất cả các đỉnh của đa giác thứ nhất nằm trong một nửa không gian của mặt phẳng này và các đỉnh của đa giác thứ hai nằm trong nửa không gian kia. Các thuật toán quy hoạch tuyến tính được sử dụng để kiểm tra xem có giải pháp khả thi nào cho tập các ràng buộc đã cho hay không.

Quy ước thông tin về khối bao  $\mathcal{A}$  là tọa độ của một đỉnh ( $\vec{A}$ ), ba vectơ cạnh ( $\vec{u}, \vec{v}, \vec{w}$ ) từ đỉnh này. Khối bao được xác định với 6 giá trị  $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$  như sau:

$$\forall (x, y, z) \in [0.0, 1.0]^3:$$

$$\text{Đặt } \Delta = x \cdot \vec{u} + y \cdot \vec{v} + z \cdot \vec{w}$$

$$\overline{(x_{min}, y_{min}, z_{min})} \leq \vec{A} + \Delta \leq \overline{(x_{max}, y_{max}, z_{max})} \quad (2.7)$$

Xét hai khối bao  $\mathcal{A}$  và  $\mathcal{B}$ , lần lượt được xác định bởi  $(\vec{A}, \vec{u}, \vec{v}, \vec{w})$  và  $(\vec{B}, \vec{u}', \vec{v}', \vec{w}')$ , với  $\vec{u} = (u_1, u_2, u_3)$ ,  $\vec{v} = (v_1, v_2, v_3)$ ,  $\vec{w} = (w_1, w_2, w_3)$ ,  $\vec{u}' = (u'_1, u'_2, u'_3)$ ,  $\vec{v}' = (v'_1, v'_2, v'_3)$ ,  $\vec{w}' = (w'_1, w'_2, w'_3)$ ,  $\vec{A} = (A_1, A_2, A_3)$ ,  $\vec{B} = (B_1, B_2, B_3)$ .

Nếu  $\mathcal{A}$  là một khối lập phương thì nó được xác định như sau:

$$\left( (x, y, z) \in [0.0, 1.0]^3: \vec{A} + x \cdot \vec{u} + y \cdot \vec{v} + z \cdot \vec{w} \right) \quad (2.8)$$

Nếu  $\mathcal{A}$  là một khối tứ diện thì nó được xác định như sau:

$$\left( (x, y, z) \in [0.0, 1.0]^3: \begin{cases} \vec{A} + x \cdot \vec{u} + y \cdot \vec{v} + z \cdot \vec{w} \\ x + y + z \leq 1.0 \end{cases} \right) \quad (2.9)$$

Nếu  $\mathcal{B}$  là một khối lập phương thì nó được xác định như sau:

$$\left( (x', y', z') \in [0.0, 1.0]^3: \vec{B} + x' \cdot \vec{u}' + y' \cdot \vec{v}' + z' \cdot \vec{w}' \right) \quad (2.10)$$

Nếu  $\mathcal{B}$  là một khối tứ diện thì nó được xác định như sau:

$$\left( (x', y', z') \in [0.0, 1.0]^3 : \begin{cases} \vec{B} + x' \cdot \vec{u}' + y' \cdot \vec{v}' + z' \cdot \vec{w}' \\ x' + y' + z' \leq 1.0 \end{cases} \right) \quad (2.11)$$

Nếu biểu diễn  $\mathcal{B}$  trong cơ sở của  $\mathcal{A}$  là  $(\vec{B}_{\mathcal{A}}, \vec{u}'_{\mathcal{A}}, \vec{v}'_{\mathcal{A}}, \vec{w}'_{\mathcal{A}})$ , ta có:

$$\left( (x', y', z') : \begin{cases} \vec{B}_{\mathcal{A}} + x' \cdot \vec{u}'_{\mathcal{A}} + y' \cdot \vec{v}'_{\mathcal{A}} + z' \cdot \vec{w}'_{\mathcal{A}} \\ x' + y' + z' \leq 1.0 \end{cases} \right) \quad (2.12)$$

với:

$$\left( (\vec{u}'_{\mathcal{A}}) (\vec{v}'_{\mathcal{A}}) (\vec{w}'_{\mathcal{A}}) (\vec{B}_{\mathcal{A}}) \right) = ((\vec{u}) (\vec{v}) (\vec{w}))^{-1} \cdot ((\vec{u}') (\vec{v}') (\vec{w}')) (\vec{B} - \vec{A})$$

Khi đó  $\mathcal{A}$  và  $\mathcal{B}$  tiếp xúc với nhau nếu:

$$\exists (x', y', z') \in [0.0, 1.0]^3 : \begin{cases} \vec{0} \leq \vec{B}_{\mathcal{A}} + x' \cdot \vec{u}'_{\mathcal{A}} + y' \cdot \vec{v}'_{\mathcal{A}} + z' \cdot \vec{w}'_{\mathcal{A}} \leq \vec{1} \\ \left( \vec{B}_{\mathcal{A}} + (x', y', z') \circ (\vec{u}_{\mathcal{A}} + \vec{v}_{\mathcal{A}} + \vec{w}_{\mathcal{A}}) \right) \cdot \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \end{pmatrix} \leq 1.0 \\ x' + y' + z' \leq 1.0 \end{cases} \quad (2.13)$$

Ký hiệu  $\vec{m} = (m_1, m_2, m_3)$  và  $\vec{n} = (n_1, n_2, n_3)$

$$\begin{aligned} \vec{m} &= -\vec{B}_{\mathcal{A}} \\ \vec{n} &= \vec{1} - \vec{B}_{\mathcal{A}} \end{aligned}$$

Khi đó (2.13) tương đương với:

$$\exists (x', y', z') \in [0.0, 1.0]^3 : \begin{cases} m_1 \leq x' \cdot u'_{\mathcal{A}1} + y' \cdot v'_{\mathcal{A}1} + z' \cdot w'_{\mathcal{A}1} \leq n_1 \\ m_2 \leq x' \cdot u'_{\mathcal{A}2} + y' \cdot v'_{\mathcal{A}2} + z' \cdot w'_{\mathcal{A}2} \leq n_2 \\ m_3 \leq x' \cdot u'_{\mathcal{A}3} + y' \cdot v'_{\mathcal{A}3} + z' \cdot w'_{\mathcal{A}3} \leq n_3 \\ x' \cdot (u'_{1\mathcal{A}} + v'_{1\mathcal{A}} + w'_{1\mathcal{A}}) + \\ y' \cdot (u'_{2\mathcal{A}} + v'_{2\mathcal{A}} + w'_{2\mathcal{A}}) + \\ z' \cdot (u'_{3\mathcal{A}} + v'_{3\mathcal{A}} + w'_{3\mathcal{A}}) \leq 1.0 + m_1 + m_2 + m_3 \\ x' + y' + z' \leq 1.0 \end{cases} \quad (2.14)$$

Cuối cùng,  $\mathcal{A}$  và  $\mathcal{B}$  tiếp xúc với nhau nếu hệ bất phương trình tuyến tính sau có nghiệm và các nghiệm của hệ này xác định điểm va chạm của  $\mathcal{A}$  và  $\mathcal{B}$ :

$$\mathcal{R}^3: \left\{ \begin{array}{l} (x_1, x_2, x_3) \in \\ a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 \leq b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 \leq b_2 \\ a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 \leq b_3 \\ a_{41} \cdot x_1 + a_{42} \cdot x_2 + a_{43} \cdot x_3 \leq b_4 \\ a_{51} \cdot x_1 + a_{52} \cdot x_2 + a_{53} \cdot x_3 \leq b_5 \\ a_{61} \cdot x_1 + a_{62} \cdot x_2 + a_{63} \cdot x_3 \leq b_6 \\ x_1 \leq 1.0 \\ -x_1 \leq 0.0 \\ x_2 \leq 1.0 \\ -x_2 \leq 0.0 \\ x_3 \leq 1.0 \\ -x_3 \leq 0.0 \\ (\sum_{i=1}^3 a_{1i}) \cdot x_1 + (\sum_{i=1}^3 a_{3i}) \cdot x_2 + (\sum_{i=1}^3 a_{5i}) \cdot x_3 \leq 1.0 - b_2 - b_4 - b_6 \\ x_1 + x_2 + x_3 \leq 1.0 \end{array} \right.$$

với:

$$\begin{array}{llll} a_{11} = u'_{\mathcal{A}1} & a_{12} = v'_{\mathcal{A}1} & a_{13} = w'_{\mathcal{A}1} & b_1 = n_1 \\ a_{21} = -u'_{\mathcal{A}1} & a_{22} = -v'_{\mathcal{A}1} & a_{23} = -w'_{\mathcal{A}1} & b_2 = -m_1 \\ a_{31} = u'_{\mathcal{A}2} & a_{32} = v'_{\mathcal{A}2} & a_{33} = w'_{\mathcal{A}2} & b_3 = n_2 \\ a_{41} = -u'_{\mathcal{A}2} & a_{42} = -v'_{\mathcal{A}2} & a_{43} = -w'_{\mathcal{A}2} & b_4 = -m_2 \\ a_{51} = u'_{\mathcal{A}3} & a_{52} = v'_{\mathcal{A}3} & a_{53} = w'_{\mathcal{A}3} & b_5 = n_3 \\ a_{61} = -u'_{\mathcal{A}3} & a_{62} = -v'_{\mathcal{A}3} & a_{63} = -w'_{\mathcal{A}3} & b_6 = -m_3 \end{array}$$

### *Phương pháp khử Fourier-Motzkin*

Phương pháp khử Fourier-Motzkin được xem như là sự tổng quát của phương pháp khử Gaussian đối với các hệ bất phương trình tuyến tính. Phương pháp này bao gồm việc loại bỏ một biến của hệ và viết lại một hệ mới cho phù hợp. Sau đó, thao tác loại bỏ được lặp lại trên một biến khác trong hệ mới, và cứ tiếp tục như vậy cho đến khi thu được một hệ chỉ có một biến. Từ đó, có thể thu được nghiệm cho mỗi biến nếu nó tồn tại. Việc loại bỏ biến được thực hiện như sau.



Viết hệ bất phương trình tuyến tính  $\mathcal{J}$  dưới dạng:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + \cdots + a_{1n} \cdot x_n \leq b_1 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + \cdots + a_{2n} \cdot x_n \leq b_2 \\ \vdots \\ a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \cdots + a_{mn} \cdot x_n \leq b_m \end{cases} \quad (2.15)$$

với:

$$i \in 1, 2, \dots, m$$

$$j \in 1, 2, \dots, n$$

$$x_i \in \mathcal{R}$$

$$a_{ij} \in \mathcal{R}$$

$$b_j \in \mathcal{R}$$

Để khử biến đầu tiên  $x_1$ , nhân mỗi bất phương trình với  $\frac{1}{|a_{i1}|} x_1$  trong đó

$a_{i1} \neq 0$ . Hệ (2.15) trở thành:

$$\begin{cases} x_1 + a'_{i2} \cdot x_2 + \cdots + a'_{in} \cdot x_n \leq b'_i & \text{với } (i \in \mathcal{J}_+) \\ a_{i2} \cdot x_2 + \cdots + a_{in} \cdot x_n \leq b_i & \text{với } (i \in \mathcal{J}_0) \\ -x_1 + a'_{i2} \cdot x_2 + \cdots + a'_{in} \cdot x_n \leq b'_i & \text{với } (i \in \mathcal{J}_-) \end{cases} \quad (2.16)$$

trong đó  $\mathcal{J}_+ = \{i: a_{i1} > 0.0\}$ ,  $\mathcal{J}_0 = \{i: a_{i1} = 0.0\}$ ,  $\mathcal{J}_- = \{i: a_{i1} < 0.0\}$  và

$$a'_{ij} = \frac{a_{ij}}{|a_{i1}|}, b'_i = \frac{b_i}{|a_{i1}|}$$

Khi đó  $x_1, x_2, \dots, x_n \in \mathcal{R}$  là một nghiệm của  $\mathcal{J}$  khi và chỉ khi:

$$\begin{cases} \left( \sum_{j=2}^n ((a'_{kj} + a'_{lj}) \cdot x_j) \right) \leq b'_k + b'_l & \text{với } (k \in \mathcal{J}_+, l \in \mathcal{J}_-) \\ \sum_{j=2}^n (a_{ij} \cdot x_j) \leq b_i & \text{với } (i \in \mathcal{J}_0) \end{cases} \quad (2.17)$$

và

$$\max_{l \in \mathcal{J}_-} \left( \sum_{j=2}^n (a'_{lj} \cdot x_j) - b'_l \right) \leq x_1 \leq \min_{k \in \mathcal{J}_+} \left( b'_k - \sum_{j=2}^n (a'_{kj} \cdot x_j) \right) \quad (2.18)$$

Tiến hành tương tự áp dụng trên hệ được xác định bởi (2.17) để loại bỏ biến thứ hai  $x_2$ , và cứ tiếp tục như vậy cho đến khi thu được bất phương trình:

$$\max_{l \in \mathcal{J}_-'''} (-b_l''') \leq x_n \leq \min_{k \in \mathcal{J}_+'''} (b_k''') \quad (2.19)$$

Nếu (2.19) không có nghiệm thì hệ  $\mathcal{J}$  cũng vô nghiệm. Nếu nó có nghiệm thì các giá trị nhỏ nhất và lớn nhất là giới hạn của biến  $x_n$ . Có thể nhận được một nghiệm cụ thể cho hệ  $\mathcal{J}$  bằng cách chọn một giá trị cho  $x_n$  giữa các giá trị giới hạn này, cho phép ta đặt một giá trị cụ thể cho biến  $x_{n-1}$  và cứ thế thực hiện đệ quy về đến  $x_1$ .

*Áp dụng phương pháp khử Fourier-Motzkin để phát hiện va chạm*

Phương pháp khử Fourier-Motzkin có thể được áp dụng đối với hệ bất phương trình (2.11) xác định bài toán phát hiện va chạm. Sau đó, các giá trị giới hạn của  $x_n$  trong (2.18) cho biết khoảng giá trị lớn nhất có thể có của  $x_n$  nếu hệ có một nghiệm. Tiếp tục áp dụng chiến thuật khử từ (2.13) với một tập hợp  $n-1$  biến khác được khử để có được giới hạn giá trị cho một biến khác. Và cứ tiếp tục như vậy cho đến khi có được các giá trị giới hạn cho mỗi biến.

Tuy nhiên, vẫn cần xác định xem hệ thực sự có nghiệm hay không. Chỉ dựa vào sự tồn tại của các giá trị giới hạn này có thể tạo ra các kết quả *đương tính giả* (False Positives - FP) như trong ví dụ về hai tứ diện sau:

$$\begin{array}{l|l} \vec{A} = (0.5, 0.5, 0.5) & \vec{B} = (-0.6, -0.6, -0.6) \\ \vec{u} = (-1.0, 0.0, 0.0) & \vec{u}' = (1.0, 0.0, 0.0) \\ \vec{v} = (0.0, -1.0, 0.0) & \vec{v}' = (0.0, 1.0, 0.0) \\ \vec{w} = (0.0, 0.0, -1.0) & \vec{w}' = (0.0, 0.0, 1.0) \end{array}$$

Trong trường hợp này, có các giá trị giới hạn thỏa mãn hệ bất phương trình cho từng biến một cách độc lập  $((0.3-0.8), (0.3-0.8), (0.3-0.8))$ , nhưng thực tế thì hệ không có nghiệm.

Nếu va chạm được phát hiện, khối bao được xác định bởi các giá trị giới hạn sẽ định vị vị trí xảy ra va chạm. Điều này được tận dụng để giảm số lượng các nút được kiểm tra trong cấp độ tiếp theo của cấu trúc cây. Quá trình phát hiện va chạm bao gồm thực hiện duyệt cấu trúc cây và kiểm tra va chạm trên các cặp nút dựa trên phương pháp khử Fourier-Motzkin.

### 2.2.1.2. Mô tả thuật toán ABFME

Mô tả thuật toán đề xuất như sau:

---

#### **Algorithm3**

---

**Đầu vào:** 03 tham số ( $P$ ,  $Cl$  là các vùng không gian khối bao,  $T$  là nút gốc của cây).

**Đầu ra:** Thông tin về va chạm hoặc false (không có va chạm).

---

```

1:  $M_i \leftarrow \text{isSubLeft}(T)$ ;
2: for node  $i$  in  $T$  do
    if  $M_i \in Cl$  then
        return false;
    else
        Test()
        if (Test()==false) exit;
        else return Algorithm3(isSub( $P$ ), isSub( $M$ ),  $T$ );
        end if
    end if
end for

```

---

Thủ tục kiểm tra va chạm **Test**() được trình bày dưới đây sẽ trả về giá trị true nếu có va chạm và trả về giá trị false nếu không xảy ra va chạm, độ phức tạp thuật toán là  $O(n \log_2 n)$ .

Mô tả thủ tục kiểm tra va chạm **Test**() như sau:

---

#### **Test**

---

**Đầu vào:** 02 tham số ( $P$ ,  $Q$  là các đối tượng cần kiểm tra va chạm).

**Đầu ra:** True nếu có va chạm, false nếu không có va chạm.

---

```

1:  $s_0 \leftarrow s(0)$ ;
2:  $q_0 \leftarrow Q(P)$ ;
3:  $f \leftarrow \text{fme}()$ ;
4: if  $f=0$  then
    return false;

```

```

else
    return true;
end if

```

## 2.2.2. Kết quả và thảo luận

### 2.2.2.1. Thử nghiệm thứ nhất

Trong các thử nghiệm, NCS cài đặt các thuật toán bằng ngôn ngữ C++ trên máy tính HP Workstation Z420 GTX750 CPU Intel Xeon Core E5-2665 (8 nhân) 3,1 GHz và RAM 16 GB.

Kịch bản thử nghiệm thứ nhất, NCS đã tạo ra 1.000.000 cặp tứ diện/hình khối lập phương ngẫu nhiên, sau đó thực hiện phát hiện va chạm trên từng cặp đối tượng bằng thuật toán SAT và thuật toán đề xuất. NCS tính toán tỷ lệ (*thời gian thực hiện thuật toán SAT*)/(*thời gian thực hiện thuật toán đề xuất*) như trong Bảng 2.3.

*Bảng 2.3.* Bảng so sánh tỷ lệ thời gian phát hiện va chạm của các thuật toán trong thử nghiệm thứ nhất

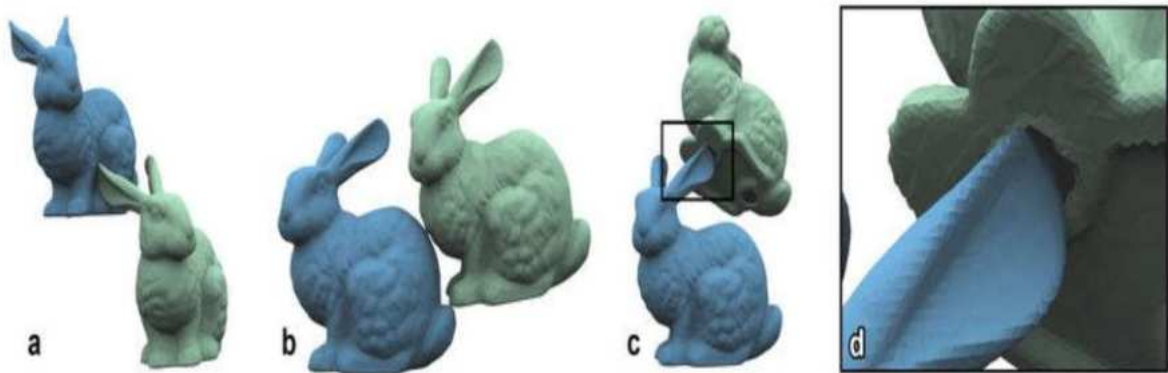
Ngữ cảnh	Lập phương – Lập phương		Lập phương – Tứ diện		Tứ diện – Tứ diện	
	Va chạm	Không va chạm	Va chạm	Không va chạm	Va chạm	Không va chạm
Tĩnh	1,39	2,01	2,44	1,52	4,17	1,12
Động	2,58	1,41	1,37	1,23	1,85	1,05

Các tỷ lệ trong Bảng 2.3 luôn có giá trị hơn 1, điều đó thể hiện thuật toán đề xuất nhanh hơn thuật toán SAT.

### 2.2.2.2. Thử nghiệm thứ hai

Kịch bản thử nghiệm thứ hai, NCS thực nghiệm phát hiện va chạm của mô hình thỏ được cung cấp bởi Phòng thí nghiệm Đồ họa máy tính của Đại học Stanford, Hoa Kỳ<sup>2</sup>.

Mô hình này bao gồm 69.451 hình tam giác và có định dạng tệp là .ply, loại khối bao được sử dụng là OOBB.



Hình 2.9. Thử nghiệm phát hiện va chạm của các mô hình thỏ

NCS thực hiện phát hiện va chạm trên cặp mô hình thỏ bằng thuật toán SAT và thuật toán đề xuất. Sau đó NCS tính toán tỷ lệ (*thời gian thực hiện thuật toán SAT*)/(*thời gian thực hiện thuật toán đề xuất*) như trong Bảng 2.4.

Bảng 2.4. Bảng so sánh tỷ lệ thời gian phát hiện va chạm của các thuật toán trong thử nghiệm thứ hai

Ngữ cảnh	Va chạm	Không va chạm
Tĩnh	1,32	1,84
Động	2,45	1,34

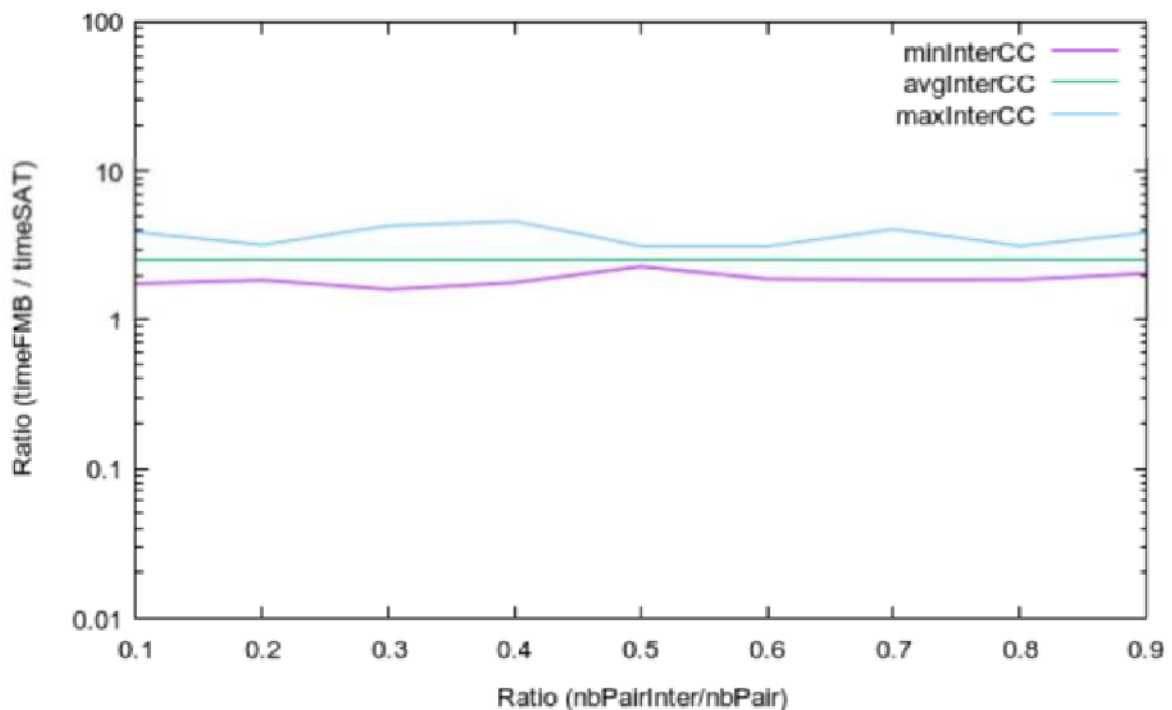
Các tỷ lệ này có giá trị lớn hơn 1 đã chứng tỏ hiệu quả tăng tốc độ phát hiện va chạm của thuật toán đề xuất.

<sup>2</sup> <https://graphics.stanford.edu/software/scanview/models/bunny.html>

### 2.2.2.3. Thảo luận

#### Ưu điểm

Từ các kết quả thử nghiệm trong cả hai kịch bản thử nghiệm cho thấy thuật toán đề xuất nhanh hơn thuật toán SAT, do đó thuật toán đề xuất sẽ là lựa chọn tốt hơn cho các ứng dụng mô phỏng 3D. Thuật toán SAT thực hiện phép chiếu của tất cả các đỉnh lên một trục và sau đó kiểm tra kết quả là điều kiện giao nhau, không thể xem trước trục nào sẽ dẫn đến điều kiện được kiểm tra cho một cặp không giao nhau. Trong thuật toán đề xuất, các điều kiện phụ thuộc vào cách xây dựng hệ phương trình tuyến tính.



Hình 2.10. Đánh giá hiệu suất của thuật toán đề xuất

Thuật toán đề xuất có ưu điểm là tính đơn giản, dễ thực hiện, phù hợp với kỹ thuật tính toán song song và có thể thích ứng với số chiều không gian lớn hơn. Đặc biệt, thuật toán đề xuất sử dụng cấu trúc dữ liệu tương thích với thuật toán SAT nên chúng có thể được kết hợp trong cùng một hệ thống để tận dụng lợi thế của cả hai thuật toán.

### *Hạn chế*

*Thứ nhất*, thuật toán đề xuất có thể đạt đến hiệu suất tốt hơn nữa bằng cách sắp xếp lại thứ tự các bất phương trình trong hệ; tuy nhiên, chi phí dành cho việc sắp xếp lại tốn kém hơn lợi ích mà nó mang lại.

*Thứ hai*, mặc dù có tính khả thi nhưng chiến lược song song hóa chưa được áp dụng. Đây sẽ là những hướng cải tiến cho nghiên cứu của NCS trong thời gian tới.

### **2.3. Kết luận Chương 2**

Trong chương 2, NCS trình bày các kết quả nghiên cứu cải tiến phát hiện nhanh và chạm của mô hình vải được biểu diễn dưới dạng mặt tam giác. Phương pháp đề xuất sử dụng thuật toán lọc pha rộng và lọc pha hẹp được thử nghiệm và cho kết quả tốt trên ba bộ dữ liệu mô hình vải khác nhau trong thư viện GAMMA, tốc độ phát hiện va chạm trung bình nhanh gấp khoảng 2,3 lần so với phương pháp của Tang và cộng sự, nhanh gấp khoảng 1,34 lần so với phương pháp của Zhang và cộng sự. Trong tương lai, NCS dự kiến nghiên cứu phương pháp phát hiện va chạm với các cấu trúc lưới mô hình có thể bị cắt hoặc rách, vỡ,... và mở rộng cách tiếp cận này với GPU để cải thiện và tăng tốc thuật toán hơn nữa.

Cũng trong chương 2, NCS trình bày các kết quả nghiên cứu ứng dụng quy hoạch tuyến tính để tăng tốc phát hiện va chạm của các mô hình 3D. Đầu tiên, NCS biểu diễn bài toán phát hiện va chạm giữa các đối tượng thành một hệ bất phương trình tuyến tính, sau đó trình bày một kỹ thuật cải tiến cách giải hệ bất phương trình này để cải thiện tốc độ tính toán phát hiện va chạm. Từ các kết quả thử nghiệm cho thấy, thuật toán đề xuất nhanh hơn thuật toán SAT và được chứng tỏ là lựa chọn tốt hơn cho các ứng dụng mô phỏng 3D. Nghiên cứu của NCS có thể được mở rộng theo hướng đề xuất những cách hiệu quả hơn để triển khai trên các nền tảng phần cứng khác nhau. Ngoài ra,

NCS dự kiến cải tiến cách tiếp cận này theo phương pháp xử lý song song, cũng có thể lai ghép kết hợp với các thuật toán khác.

Một phần kết quả nghiên cứu trình bày tại chương 2 được công bố tại [CT4, CT5] trong Danh mục công trình công bố liên quan đến luận án.



### **CHƯƠNG 3. PHƯƠNG PHÁP PHÁT HIỆN VA CHẠM CỦA CÁNH TAY COBOT 6 BẬC TỰ DO**

Cobot là những robot có thể tiếp xúc với con người một cách trực tiếp, phát huy đồng thời được các lợi thế của cả con người và cobot để tăng hiệu suất công việc. Cobot hoạt động thân thiện và tương tác với con người vì được lập trình để phát hiện va chạm an toàn. Do đó yêu cầu phát hiện chính xác và nhanh chóng các va chạm của cánh tay cobot là một chủ đề thu hút được sự quan tâm của nhiều nhà nghiên cứu. Trong chương 3, NCS đề xuất hướng tiếp cận áp dụng phương pháp học máy có giám sát là SVR (Support Vector Regression) cho kết quả phát hiện va chạm hiệu quả với cánh tay cobot CURA6 trên bộ dữ liệu của Intema.

Từ việc phân tích đánh giá các nghiên cứu liên quan trong chương 1, để khắc phục các hạn chế của những nghiên cứu hiện nay (sự phụ thuộc vào ma sát của động cơ, đặc biệt là ma sát Coulomb), NCS đề xuất áp dụng phương pháp SVR để phát hiện va chạm của cánh tay cobot CURA6 trên cơ sở các phép đo dòng điện cùng với mô hình động lực học của cobot. Hướng nghiên cứu của NCS có ưu điểm là không yêu cầu bất kỳ cảm biến bên ngoài bổ sung nào khác, so với [34] yêu cầu cảm biến mômen xoắn khớp tại các khớp.

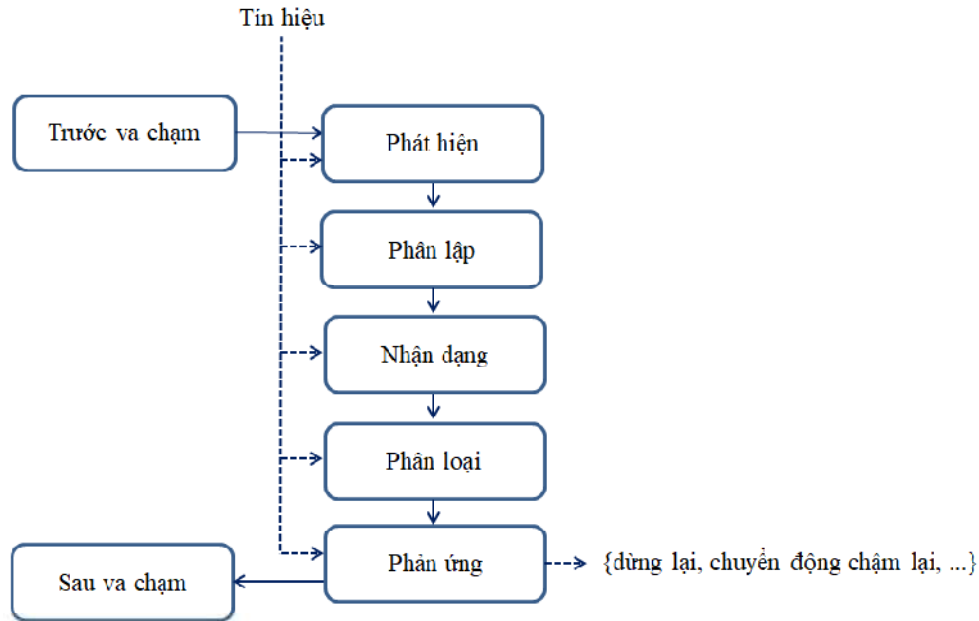
Phương pháp của NCS không yêu cầu bù ma sát, chi phí tính toán trong phương pháp đề xuất ít hơn đáng kể cho cả huấn luyện và suy luận (đầu vào 4 chiều) trong khi phương pháp của Y. J. Heo [36] có đầu vào 66 chiều.

#### **3.1. Phương pháp đề xuất**

##### ***3.1.1. Phương pháp phát hiện va chạm***

Bài toán xử lý va chạm có thể được chia thành những bài toán con là: Phát hiện va chạm (detection), phân lập va chạm (isolation), nhận dạng va chạm (identification), phân loại va chạm (classification) và phản ứng va chạm (reaction). Trọng tâm nghiên cứu của NCS là bài toán phát hiện va chạm, đây

chính là bước đầu tiên, có tính chất định hướng các tác vụ khác trong toàn bộ hệ xử lý va chạm [29].



Hình 3.1. Vị trí của bài toán phát hiện va chạm trong hệ xử lý va chạm [29]

Trên cơ sở phương pháp SVR [62] là nền tảng, NCS đề xuất phát triển mô hình phát hiện va chạm cho cánh tay cobot CURA6. Quy ước: Một tín hiệu  $f$ -chiều thay đổi theo thời gian  $s(t) \in \mathbb{R}^f$  được lấy mẫu tại khoảng thời gian lấy mẫu  $t_l$  trên một cửa sổ thời gian có kích thước  $t_w$ ; số lượng mẫu là  $N + 1 = t_w/t_l + 1 \in \mathbb{N}$ .

Quá trình lập trình thử nghiệm được thực hiện bằng ngôn ngữ Python. Các phương pháp đề xuất đã khắc phục được những trở ngại về sự không chắc chắn của các tham số trong mô hình động lực học và các hiệu ứng không được mô hình hóa (ma sát, nhiễu đo cảm biến).

### 3.1.1.2. Phát hiện va chạm bằng SVR

Support vector machine (SVM) rất hiệu quả trong lớp các bài toán phân loại nhị phân và dự báo của học có giám sát. Thuật toán này có ưu điểm là hoạt động tốt đối với những mẫu dữ liệu có kích thước lớn và thường mang lại kết quả vượt trội so với lớp các thuật toán khác trong học có giám sát.

Ưu điểm của SVM là:

- Hoạt động hiệu quả với không gian có số chiều cao (high dimensional spaces).
- Thuật toán tiêu tốn ít bộ nhớ vì chỉ sử dụng các điểm trong tập hỗ trợ để dự báo trong hàm quyết định.
- Có thể tạo ra nhiều hàm quyết định từ những hàm kernel khác nhau. Thậm chí sử dụng đúng kernel có thể giúp cải thiện thuật toán lên đáng kể.

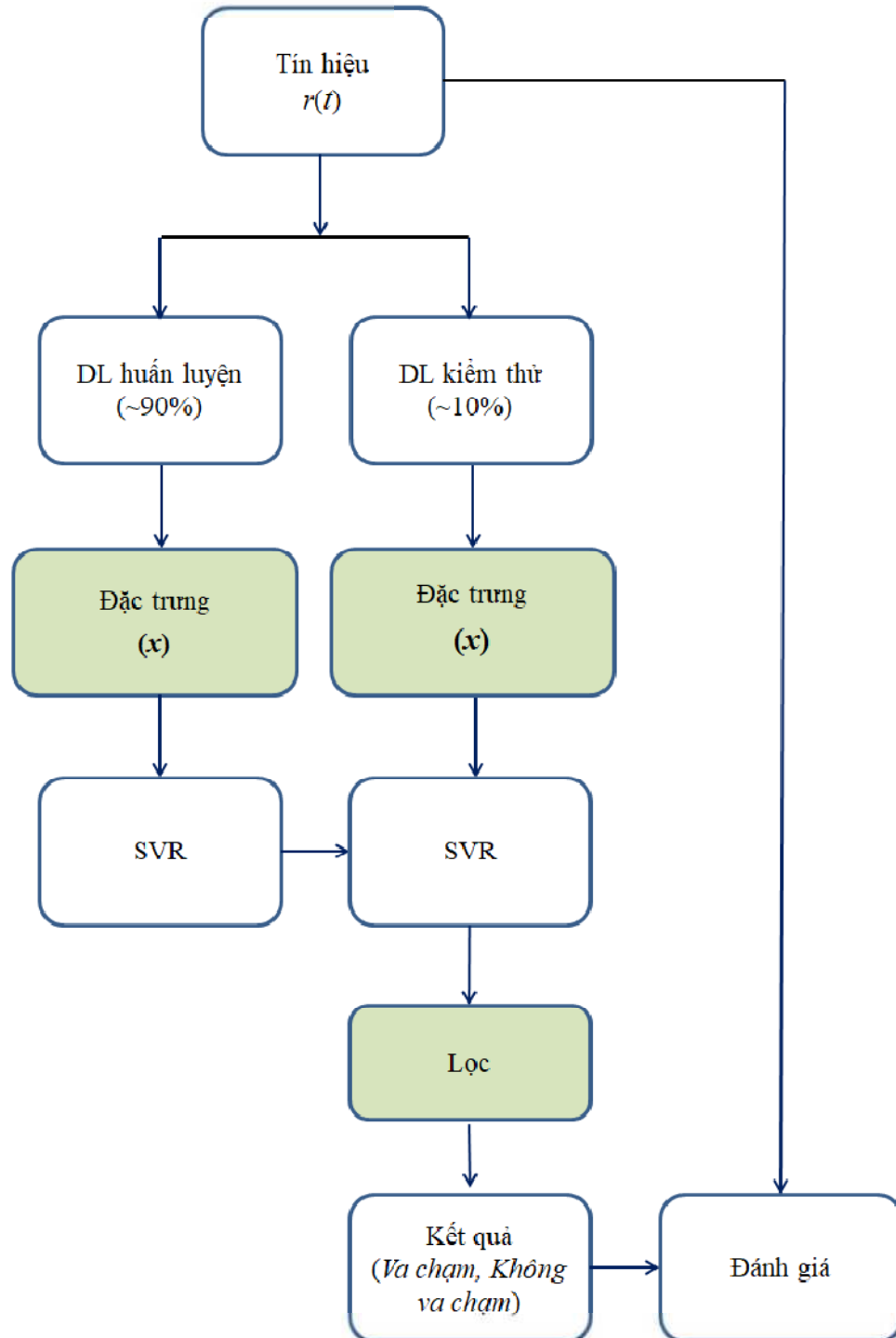
Chính vì tính hiệu quả mà SVM thường được áp dụng nhiều trong các tác vụ phân loại và dự báo, có thể liệt kê một số ứng dụng của thuật toán SVM đó là:

- Mô hình chuẩn đoán bệnh.
- Mô hình phân loại ảnh.
- Mô hình phân loại tin tức, phân loại văn bản, phân loại thư rác.
- Mô hình phát hiện gian lận.

Thuật toán hồi quy vector hỗ trợ (Support Vector Regression - SVR) là một loại máy học hồi quy hỗ trợ hồi quy tuyến tính và phi tuyến tính. Máy học vector hỗ trợ có thể được sử dụng cho bài toán hồi quy, giữ lại những đặc trưng chính để máy học tận dụng tối đa nhằm tạo ra vùng biên tối đa. Nhưng thuật toán SVR, dù chung một ý tưởng với thuật toán phân loại SVM, có một số đặc điểm khác biệt.

SVR cố gắng tìm ra hàm mục tiêu khớp với dữ liệu huấn luyện tốt nhất, nhưng đồng thời cũng cố giảm thiểu mất mát giữa giá trị đoán nhận và giá trị thực. Hàm mục tiêu ở đây được gọi là biên, và những điểm nằm gần nhất với nó được gọi là các vector hỗ trợ. Ý tưởng then chốt của SVR chính là đặt ra lề cho phép mất mát ở xung quanh các các vector hỗ trợ. Nhờ đó, mô hình trở nên khái quát hơn với dữ liệu mới, giúp mô hình có thể chịu đựng được nhiễu hoặc sự biến thiên trong dữ liệu. SVR có thể được sử dụng cho các bài toán

hồi quy, như là dự đoán chứng khoán, giá nhà cửa,... SVR được lựa chọn sử dụng vì thuật toán có thể dùng các hàm nhân (kernels) để biến đổi dữ liệu và chiều dữ liệu lên các chiều cao hơn, do có thể tìm được các hàm mục tiêu phù hợp với dáng hình của dữ liệu.



Hình 3.2. Sơ đồ phương pháp SVR đề xuất

SVR lấy giá trị tuyệt đối của đầu ra theo công thức (1.12) được trình bày trong chương 1 làm đầu vào.

Thiết kế vectơ đặc trưng: Thiết kế vectơ đặc trưng bao gồm chuyển tín hiệu thành dạng vectơ và gán nhãn cho các đặc trưng. Để lấy mẫu theo thời gian, qua khảo sát thực nghiệm NCS đặt  $t_W = 80$  ms và  $t_I = 8$  ms. Sau khi lấy mẫu đầu ra của bộ quan sát động lượng và biến đổi thành ma trận  $R \in \mathbb{R}^{n \times (N+1)}$ , giá trị các hàng 1, 2, 3 và 4 của  $R$  tạo thành một vectơ duy nhất, dẫn đến kết quả vectơ đặc trưng  $x(t)$  như sau:

$$x = [rj_1 \quad rj_2 \quad rj_3 \quad rj_4]^T \in \mathbb{R}^{4(N+1)} \quad (3.1)$$

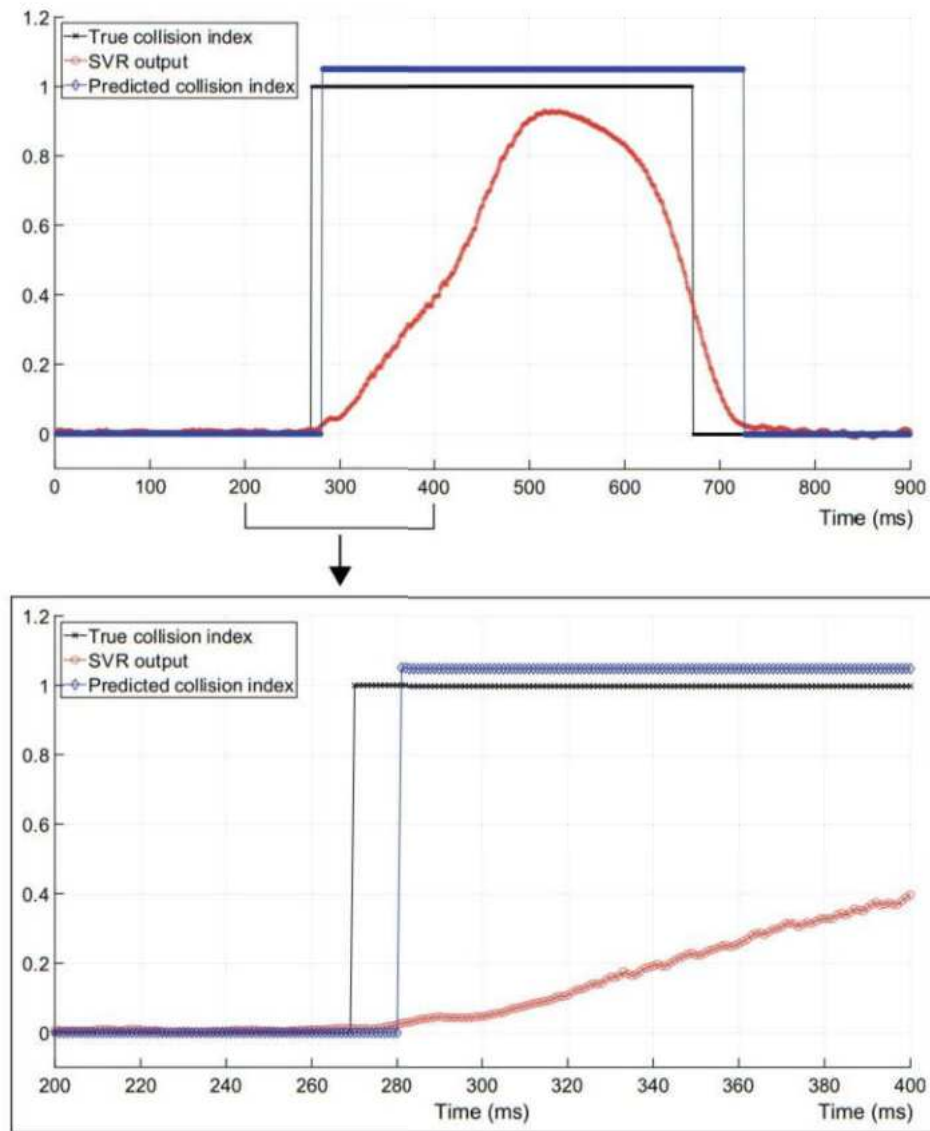
trong đó  $rj_i \in \mathbb{R}^{1 \times (N+1)}$  biểu thị giá trị của hàng có thứ tự  $i$  của  $R$ . Lưu ý rằng hàng thứ  $i$  của  $R$  biểu thị đầu ra quan sát động lượng được lấy mẫu của khớp  $i$ . So với việc sử dụng tín hiệu của tất cả các khớp, kích thước của vectơ đặc trưng  $x$  này nhỏ hơn, giảm lượng tính toán cho SVR.

Huấn luyện SVR: Vì hàm mũ yêu cầu tính toán nhiều hơn các hàm đa thức, nên NCS sử dụng hàm SRQ [65] thay vì hàm RBF:

$$K_{SRQ}(x, z) = \frac{1}{\left(1 + \frac{\|x-z\|^2}{\sigma^2}\right)^2} \quad (3.2)$$

với các tham số  $C = 1$ ,  $\varepsilon = 0.02$  (dựa theo [62]) và  $\sigma = 3$ .

Điều chỉnh độ nhạy phát hiện va chạm: Đầu ra của SVR được huấn luyện là một giá trị vô hướng được chuẩn hóa thành một đơn vị được mô tả như trong Hình 3.1 sau đó được đặt ngưỡng và chuyển thành chỉ số va chạm nhị phân tương ứng theo công thức (1.14) đã trình bày trong chương 1.



Hình 3.3. Đầu ra của SVR và chỉ số va chạm tương ứng

Giá trị đầu ra gần bằng 0 khi không có lực bên ngoài tác dụng và tăng khi mômen xoắn bên ngoài ước tính của khớp 1–4 tăng và giảm khi mômen xoắn bên ngoài giảm. Mômen xoắn bên ngoài ước tính càng lớn và số lượng khớp bị tác động của ngoại lực càng nhiều thì giá trị đầu ra SVR càng lớn.

### 3.1.1.3. Lọc đầu ra

Tiến hành lọc đầu ra để giảm dương tính giả (cảnh báo sai). Va chạm chỉ được cảnh báo nếu đầu ra của phương pháp phát hiện va chạm liên tục là *True* trong khoảng thời gian  $t_c$  ms. Bộ lọc đầu ra loại bỏ các lỗi (cảnh báo sai

với thời gian kéo dài trong phạm vi  $1 - t_c$  ms) giúp phát hiện va chạm hiệu quả hơn, NCS đặt  $t_c=3$  ms.

### 3.1.2. Bộ dữ liệu thử nghiệm

Bộ dữ liệu này được tải về từ trang Gitlab: <https://gitlab.com/intema-gdansk/cura6-dataset/-/tree/main> [25]. Bộ dữ liệu gồm hai bộ dữ liệu con là bộ dữ liệu ngưỡng 1,6 [A] và bộ dữ liệu ngưỡng 2,0 [A].

Có 15 lát (slices) chuyển động ngẫu nhiên của cobot đã được chuẩn bị để làm dữ liệu huấn luyện, mỗi lát chứa 10000 mẫu (7 phút chuyển động). Tốc độ của mỗi khớp được bộ giải thiết kế an toàn nằm trong khoảng 25% tốc độ tối đa của động cơ. 03 lát được thu thập mà cobot không có tải trọng và 12 lát còn lại có tải (đơn vị tính bằng gam) từ danh sách sau {782, 1016, 1282, 2298, 2757, 4039}, với 02 lát cho mỗi tải (phạm vi tải xấp xỉ là từ 500 g đến 4000 g) [25]. Trong đó, dữ liệu huấn luyện (90%) và dữ liệu thử nghiệm (10%).

Các tín hiệu cần được chuẩn hóa trong phạm vi [0,1] trước khi được xử lý làm đầu vào. Đầu ra không xét đến ma sát được mô tả bằng phương trình:

$$r = K_o \left( p(t) - \int_0^t (\tau_m - \hat{\beta} + r) ds - p(0) \right) \quad (3.3)$$

với:

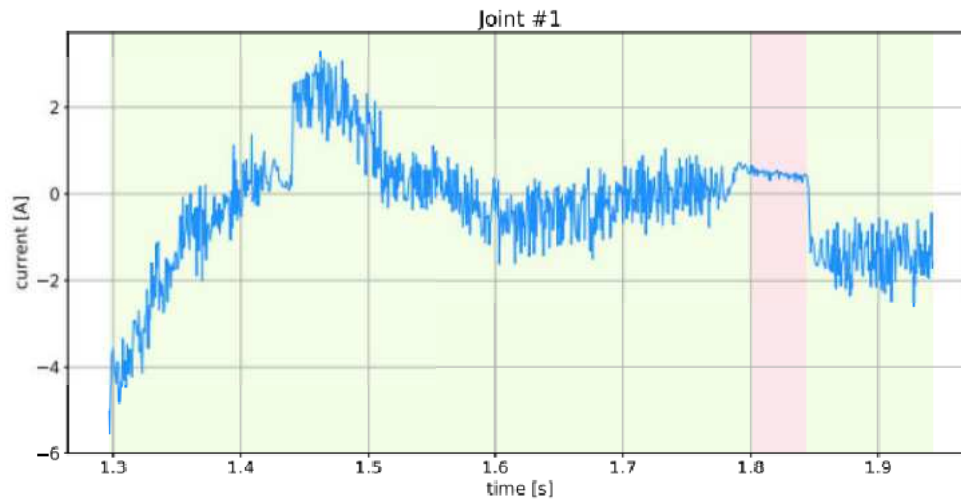
$$\dot{p} = \tau_m - \hat{\beta}(q, \dot{q}) + r \quad (3.4)$$

và:

$$\dot{r} = K_o(\dot{p} - \dot{\hat{p}}) \quad (3.5)$$

Trong các điều kiện lý tưởng, mối quan hệ động giữa  $r(t)$  và mômen xoắn khớp ngoài  $\tau_{ext}$  là:

$$\dot{r} = K_O(\tau_{ext} - \tau_F - r) \quad (3.6)$$



(a) Current of the engine #1 of the analyzed robot.



(b) The actual velocity of the robot and the set (target) speed.



Hình 3.4. Ví dụ về dữ liệu [25]



## 3.2. Kết quả và thảo luận

Thử nghiệm dùng các độ đo được mô tả ở chương 1 như *Precision*, *Recall*..., được tính toán từ các ma trận nhầm lẫn. Với *TP* biểu thị số lượng dương tính thật (đoán đúng va chạm), *FP* biểu thị số lượng dương tính giả (không va chạm mà phán đoán nhầm là va chạm), *FN* biểu thị số lượng âm tính giả (va chạm mà phán đoán nhầm là không va chạm), và *TN* là số lượng âm tính thật (đoán đúng không va chạm).

Thử nghiệm đánh giá các độ đo của SVR với các ngưỡng phát hiện khác nhau. Sau đó, đối với mỗi phương pháp, với ngưỡng cố định và tham số bộ lọc đầu ra cố định, đo độ trễ phát hiện trung bình và số lần phát hiện không thành công đối với các va chạm. Đối với SVR, trong khi tăng dần ngưỡng phát hiện  $\alpha$  từ  $\alpha = -0.08$  (tất cả các nhãn được dự đoán là 1) lên  $\alpha = 0.98$  (tất cả các nhãn được dự đoán là 0), thực hiện tính toán các độ đo cho từng giá trị ngưỡng cho kết quả chi tiết được trình bày ở các mục tiếp theo.

Thử nghiệm đánh giá cho thấy phương pháp đề xuất cho kết quả tốt. Tiếp theo, NCS sẽ trình bày kết quả chi tiết với phương pháp SVR. Lý do lựa chọn SVR để phát triển tiếp vì đây là phương pháp học máy có giám sát đơn giản, dễ áp dụng, yêu cầu về dữ liệu để huấn luyện thấp hơn các phương pháp học sâu, mà vẫn cho kết quả tốt. Kết quả thu được sẽ được so sánh với [25] (sử dụng cùng bộ dữ liệu thử nghiệm). Lưu ý, trong nghiên cứu này, NCS chỉ tập trung vào vấn đề phát hiện va chạm.

### 3.2.1. Kết quả với bộ dữ liệu ngưỡng 1,6 [A]

Tất cả các bộ dữ liệu của trường này hợp đều có thực tế 65 va chạm. Số trường hợp không va chạm thì khác nhau.

*Bảng 3.1. Ma trận nhầm lẫn, Tốc độ max V 10% - Ngưỡng 1,6 [A]*

Tốc độ max V 10% Ngưỡng 1,6 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	60	29
	Không va chạm	5	113

Trong tổng số 207 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 60/65 trường hợp va chạm, phát hiện đúng 113/142 trường hợp không va chạm; chỉ có 34 trường hợp bị nhầm lẫn (5 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 29 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.2. Ma trận nhầm lẫn, Tốc độ max V 20% - Ngưỡng 1,6 [A]*

Tốc độ max V 20% Ngưỡng 1,6 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	59	40
	Không va chạm	6	135

Trong tổng số 240 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 59/65 trường hợp va chạm, phát hiện đúng 135/175 trường hợp không va chạm; chỉ có 46 trường hợp bị nhầm lẫn (6 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 40 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.3. Ma trận nhầm lẫn, Tốc độ max V 30% - Ngưỡng 1,6 [A]*

Tốc độ max V 30% Ngưỡng 1,6 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	57	76
	Không va chạm	8	136

Trong tổng số 277 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 57/65 trường hợp va chạm, phát hiện đúng 136/212 trường hợp không va chạm; có 84 trường hợp bị nhầm lẫn (8 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 76 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.4. Ma trận nhầm lẫn, Tốc độ max V 40% - Ngưỡng 1,6 [A]*

Tốc độ max V 40% Ngưỡng 1,6 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	60	65
	Không va chạm	5	158

Trong tổng số 288 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 60/65 trường hợp va chạm, phát hiện đúng 158/223 trường hợp không va chạm; có 70 trường hợp bị nhầm lẫn (5 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 65 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.5. Ma trận nhầm lẫn, Tốc độ max V 50% - Ngưỡng 1,6 [A]*

Tốc độ max V 50% Ngưỡng 1,6 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	59	90
	Không va chạm	6	180

Trong tổng số 335 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 59/65 trường hợp va chạm, phát hiện đúng 180/270 trường hợp không va chạm; có 96 trường hợp bị nhầm lẫn (6 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 90 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.6. Ma trận nhầm lẫn, Tốc độ max V 60% - Ngưỡng 1,6 [A]*

Tốc độ max V 60% Ngưỡng 1,6 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	59	87
	Không va chạm	6	167

Trong tổng số 319 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 59/65 trường hợp va chạm, phát hiện đúng 167/254 trường hợp không va chạm; có 93 trường hợp bị nhầm lẫn (6 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 87 trường hợp không va chạm mà phán đoán nhầm là va chạm).

Tổng hợp lại, ta có thể thấy với trường hợp Ngưỡng 1,6 [A] khả năng phát hiện đúng va chạm là tốt, còn khả năng phát hiện đúng không va chạm không tốt bằng.

### **3.2.2. Kết quả với bộ dữ liệu ngưỡng 2,0 [A]**

Tất cả các bộ dữ liệu của trường này hợp đều có thực tế 65 va chạm. Số trường hợp không va chạm thì khác nhau.

*Bảng 3.7. Ma trận nhầm lẫn, Tốc độ max V 10% - Ngưỡng 2,0 [A]*

Tốc độ max V 10% Ngưỡng 2,0 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	51	3
	Không va chạm	14	113

Trong tổng số 181 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 51/65 trường hợp va chạm, phát hiện đúng 113/116 trường hợp không va chạm; chỉ có 17 trường hợp bị nhầm lẫn (14

trường hợp va chạm mà phán đoán nhầm là không va chạm, và 3 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.8. Ma trận nhầm lẫn, Tốc độ max V 20% - Ngưỡng 2,0 [A]*

Tốc độ max V 20% Ngưỡng 2,0 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	50	4
	Không va chạm	15	135

Trong tổng số 204 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 50/65 trường hợp va chạm, phát hiện đúng 135/139 trường hợp không va chạm; chỉ có 19 trường hợp bị nhầm lẫn (15 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 4 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.9. Ma trận nhầm lẫn, Tốc độ max V 30% - Ngưỡng 2,0 [A]*

Tốc độ max V 30% Ngưỡng 2,0 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	53	10
	Không va chạm	12	136

Trong tổng số 211 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 53/65 trường hợp va chạm, phát hiện đúng 136/146 trường hợp không va chạm; có 22 trường hợp bị nhầm lẫn (12 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 10 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.10. Ma trận nhầm lẫn, Tốc độ max V 40% - Ngưỡng 2,0 [A]*

Tốc độ max V 40% Ngưỡng 2,0 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	56	10
	Không va chạm	9	159

Trong tổng số 234 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 56/65 trường hợp va chạm, phát hiện đúng 159/169 trường hợp không va chạm; có 19 trường hợp bị nhầm lẫn (9 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 10 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.11. Ma trận nhầm lẫn, Tốc độ max V 50% - Ngưỡng 2,0 [A]*

Tốc độ max V 50% Ngưỡng 2,0 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	56	16
	Không va chạm	9	180

Trong tổng số 261 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 56/65 trường hợp va chạm, phát hiện đúng 180/196 trường hợp không va chạm; có 25 trường hợp bị nhầm lẫn (9 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 16 trường hợp không va chạm mà phán đoán nhầm là va chạm).

*Bảng 3.12. Ma trận nhầm lẫn, Tốc độ max V 60% - Ngưỡng 2,0 [A]*

Tốc độ max V 60% Ngưỡng 2,0 [A]		Thực tế	
		Va chạm	Không va chạm
Phán đoán	Va chạm	59	12
	Không va chạm	6	167

Trong tổng số 244 trường hợp, phương pháp đề xuất đã cho kết quả cụ thể như sau: Phát hiện đúng 59/65 trường hợp va chạm, phát hiện đúng 167/179 trường hợp không va chạm; có 18 trường hợp bị nhầm lẫn (6 trường hợp va chạm mà phán đoán nhầm là không va chạm, và 12 trường hợp không va chạm mà phán đoán nhầm là va chạm).

### 3.2.3. Thảo luận

Các kết quả thu được của NCS, so sánh với M. Czubenko và nhóm nghiên cứu [25] (sử dụng cùng dữ liệu thử nghiệm), cho kết quả tương đương. Ưu điểm của phương pháp dựa trên SVR do NCS đề xuất phù hợp với ngữ cảnh dữ liệu huấn luyện hạn chế.

*Bảng 3.13.* Đánh giá kết quả và so sánh với công bố của Czubenko [25]

Tốc độ max V	Ngưỡng Threshold	Recall		Precision		Specificity		Accuracy		f1-Score	
		NCS	[25]	NCS	[25]	NCS	[25]	NCS	[25]	NCS	[25]
10%	1,6 [A]	0,923	0,908	0,674	0,670	0,796	0,796	0,836	0,831	0,779	0,771
20%		0,908	0,923	0,596	0,606	0,771	0,777	0,808	0,817	0,720	0,732
30%		0,877	0,862	0,429	0,427	0,642	0,646	0,697	0,697	0,576	0,571
40%		0,923	0,954	0,480	0,492	0,709	0,713	0,757	0,767	0,632	0,649
50%		0,908	0,908	0,396	0,399	0,667	0,670	0,713	0,716	0,551	0,554
60%		0,908	0,908	0,404	0,404	0,657	0,657	0,708	0,708	0,559	0,559
10%	2,0 [A]	0,785	0,800	0,944	0,945	0,974	0,974	0,906	0,912	0,857	0,867
20%		0,769	0,754	0,926	0,942	0,971	0,978	0,907	0,907	0,840	0,838
30%		0,815	0,800	0,841	0,852	0,932	0,938	0,896	0,896	0,828	0,825
40%		0,862	0,877	0,848	0,851	0,941	0,941	0,919	0,923	0,855	0,864
50%		0,862	0,892	0,778	0,795	0,918	0,923	0,904	0,916	0,818	0,841
60%		0,908	0,908	0,831	0,831	0,933	0,933	0,926	0,926	0,868	0,868

Tổng hợp lại, ta có thể thấy với trường hợp Ngưỡng 2,0 [A] khả năng phát hiện đúng không va chạm là tốt, còn phát hiện đúng va chạm không tốt bằng (điều này trái ngược với trường hợp Ngưỡng 1,6 [A]).

Từ kết quả thu được có thể thấy, ngưỡng cường độ dòng điện, ảnh hưởng đến việc phát hiện va chạm/không va chạm. Với thí nghiệm này, Ngưỡng 2,0 [A] phát hiện sự thay đổi dòng điện kém hơn (nên phát hiện va chạm kém hơn), kết quả phân tích thiên về việc cho là không có sự thay đổi dòng điện, mà số trường hợp không va chạm trong dữ liệu rất nhiều, nên dẫn

đến kết quả phán đoán không va chạm đúng nhiều hơn. Ngược lại, Ngưỡng 1,6 [A]) nhạy trong việc phát hiện thay đổi dòng điện hơn, nên phát hiện va chạm tốt hơn. Cũng chính vì vậy, nên trường hợp không va chạm cũng bị hiểu lầm thành va chạm nhiều hơn.

Do điều kiện nghiên cứu hạn chế, không có điều kiện thu thập dữ liệu thực tế theo kịch bản mong muốn, bắt buộc phải phụ thuộc vào dữ liệu được chia sẻ từ cộng đồng nghiên cứu. Kịch bản thử nghiệm bị giới hạn. Không hoàn toàn hiểu hết được các ý đồ của tác giả bộ dữ liệu (ví dụ việc số lượng các va chạm trong mỗi kịch bản là như nhau, nhưng số lượng không va chạm lại khác nhau).

### **3.3. Kết luận Chương 3**

Phương pháp phát hiện va chạm dựa trên SVR đề xuất đã phát hiện va chạm của cánh tay cobot 6-DoF CURA6 hiệu quả. Ưu thế nổi bật của thuật toán đề xuất là chỉ yêu cầu phép đo các cảm biến dòng điện của động cơ cùng với mô hình động lực học của cobot; không cần mô hình hóa hoặc xác định các mômen ma sát trong các khớp. Qua thử nghiệm trên bộ dữ liệu của cobot CURA6, phương pháp dựa trên SVR chỉ yêu cầu điều chỉnh một tham số không đổi. Kết quả so sánh phương pháp dựa trên SVR yêu cầu về dữ liệu va chạm ít hơn để huấn luyện, phù hợp với ngữ cảnh dữ liệu hạn chế.

So với kết quả nghiên cứu trong [25] và [36], phương pháp đề xuất chiếm ưu thế trong ngữ cảnh xử lý tác động của các tải trọng khác nhau. Tuy nhiên, việc xác thực cho các tải trọng ngẫu nhiên không xác định vẫn là một bài toán khó. Hơn nữa, bài toán phát hiện va chạm của hai hay nhiều cánh tay cobot rất phức tạp và cần được nghiên cứu chuyên sâu hơn. Đối với các cobot được sản xuất hàng loạt, một vấn đề thực tế khác là các quy trình cần thiết để phát hiện va chạm hiệu quả phải được nhân rộng trên quy mô lớn. Đây sẽ là những chủ đề mở cho nghiên cứu trong tương lai của NCS.



Một phần kết quả nghiên cứu trình bày tại chương 3 được công bố tại [CT6] trong Danh mục công trình công bố liên quan đến luận án.

## KẾT LUẬN VÀ KIẾN NGHỊ

### Kết luận

Mỗi đối tượng trong môi trường ảo được tạo thành từ rất nhiều mặt tam giác nên chi phí để kiểm tra giao nhau của từng cặp mặt rất tốn kém, mất nhiều thời gian. Vì vậy, để giải quyết vấn đề này, hầu hết các hệ thống đều sử dụng phương pháp gần đúng để phát hiện va chạm. Điển hình là kỹ thuật sử dụng khối hình học bao quanh đối tượng. Trên thế giới đã có nhiều công bố cải tiến kỹ thuật phát hiện va chạm, hầu hết đều dựa trên cấu trúc phân hệ vùng bao BVH hoặc dựa trên kỹ thuật tính toán hàm khoảng cách SDF. Cấu trúc BVH và hàm SDF đã được áp dụng hiệu quả với các mô hình vật thể rắn. Đối với các mô hình vật thể biến dạng như chất liệu vải thì chi phí tính toán SDF cũng như duyệt và tái cấu trúc BVH là rất lớn, vấn đề đặt ra là cần thiết kế thuật toán có thể hoạt động tốt trên các cấu hình biến dạng khác nhau. Những nghiên cứu về bài toán phát hiện va chạm của các nhóm tác giả tại Việt Nam chủ yếu trong việc mô hình hóa hay mô phỏng trung bày ảo mà chưa có nhiều nghiên cứu về bài toán phát hiện va chạm trong môi trường ảo. Một số kết quả nghiên cứu bước đầu tập trung vào tính toán va chạm sử dụng kỹ thuật hộp bao theo hướng và ứng dụng trong công tác tuyên truyền, phổ biến về an toàn giao thông. Tuy nhiên, việc phát hiện va chạm đối với các đối tượng có hình dạng thay đổi thường xuyên như chất lỏng, chất khí, vải, lụa,... vẫn còn là một thách thức. Nhìn chung, các kết quả nghiên cứu trên tập trung vào phát hiện va chạm của mô hình vật thể rắn bằng kỹ thuật BVH. Các nghiên cứu về phát hiện va chạm của mô hình chất liệu vải còn chưa phổ biến, hướng nghiên cứu về kỹ thuật cải tiến phát hiện va chạm của các mô hình vật thể rắn vẫn có những vấn đề cần tiếp tục nghiên cứu sâu hơn. Xuất phát từ thực tế đó, NCS đã chọn hướng nghiên cứu cải tiến kỹ thuật phát hiện nhanh và chính xác va chạm của mô hình chất liệu vải và ứng dụng phương pháp quy hoạch tuyến tính phát hiện va chạm của các mô hình vật thể rắn trong môi trường VR đã được trình bày chi tiết trong chương 2.

Công nghệ VR và robotics có mối quan hệ kết hợp tương hỗ đặc biệt, kết quả của mối quan hệ tương hỗ này là sự xuất hiện của những thành tựu khoa học công nghệ tiên tiến, tạo ra những khả năng sản xuất mới và có tác động sâu sắc đến đời sống kinh tế, xã hội của con người. Cobot được áp dụng các giải pháp kỹ thuật để đảm bảo tương tác an toàn với người vận hành và môi trường xung quanh, kết hợp được các lợi thế của cả con người và robot để đạt được năng suất cao hơn. Việc phát hiện nhanh chóng và chính xác các va chạm có thể xảy ra trở nên cấp thiết hơn bao giờ hết. Do đó chức năng phát hiện va chạm là một trong những yêu cầu bắt buộc đối với các cobot. Vì vậy, trong những năm gần đây, các nghiên cứu về robotics đã và đang dần chuyển hướng tập trung vào nghiên cứu bài toán phát hiện va chạm của cánh tay cobot. NCS nghiên cứu trường hợp điển hình là cánh tay cobot sáu bậc tự do 6-DoF được thiết kế nhằm thực hiện các tác vụ chính xác và an toàn với con người được trình bày trong chương 3. NCS đề xuất áp dụng phương pháp học có giám sát SVR để phát hiện va chạm của cánh tay cobot CURA6 trên cơ sở các phép đo dòng điện cùng với mô hình động lực học của robot.

Nhằm phát huy hiệu quả của các kỹ thuật phát hiện va chạm phục vụ cho các ứng dụng VR và robotics, NCS nhận thấy việc nghiên cứu phát triển các kỹ thuật phát hiện va chạm là phù hợp với yêu cầu thực tế. Do đó, đề tài nghiên cứu phương pháp phát hiện va chạm của luận án là cần thiết, mang tính khoa học và khả năng thực tiễn cao.

Luận án có kết quả chính như sau:

- Đề xuất thuật toán AB2PF [CT4] cho kết quả phát hiện va chạm hiệu quả trên ba bộ dữ liệu mô hình vải khác nhau trong thư viện GAMMA, tốc độ phát hiện va chạm trung bình nhanh gấp khoảng 2,3 lần so với thuật toán của Tang và cộng sự, nhanh gấp khoảng 1,34 lần so với thuật toán của Zhang và cộng sự.

- Đề xuất thuật toán ABFME [CT5] để phát hiện va chạm của các cặp khối hình hộp lập phương và tứ diện. Bài toán phát hiện va chạm giữa các đối tượng được biểu diễn thành một hệ bất phương trình tuyến tính, luận án trình bày một kỹ thuật cải tiến cách giải hệ bất phương trình này để cải thiện tốc độ tính toán phát hiện va chạm. Thuật toán đề xuất nhanh hơn thuật toán SAT khoảng 4,17 lần.

- Đề xuất áp dụng các phương pháp học có giám sát SVR để phát hiện va chạm của cánh tay cobot CURA6 trên cơ sở các phép đo dòng điện cùng với mô hình động lực học của cobot.

### **Kiến nghị**

Bài toán phát hiện va chạm có tính ứng dụng cao, nếu được khai thác và áp dụng thuật toán phù hợp sẽ thu được những thông tin rất có giá trị và sử dụng được cho phát hiện va chạm trong vật thể biến dạng và cánh tay cobot 6 bậc tự do.

Kết quả của luận án góp phần nâng cao tốc độ và độ chính xác trong bài toán phát hiện va chạm của các đối tượng mô hình 3D biến dạng (vật thể rắn, vật thể chất liệu vải) và bài toán phát hiện va chạm của cánh tay cobot 6 bậc tự do CURA6. Luận án đã mở rộng khả năng sử dụng của bài toán phát hiện va chạm phù hợp với yêu cầu thực tế. Đề tài nghiên cứu phương pháp phát hiện va chạm trong vật thể biến dạng và cánh tay cobot của luận án là cần thiết, mang tính khoa học và khả năng thực tiễn cao. Nội dung được đề cập đến trong khuôn khổ luận án chưa thể nêu hết được một cách toàn diện về bài toán phát hiện va chạm, còn nhiều bài toán có thể áp dụng và thử nghiệm trong các nghiên cứu sau này.

## DANH MỤC CÔNG TRÌNH CÔNG BỐ LIÊN QUAN ĐẾN LUẬN ÁN

- [CT1] **Van Hung Nghiem**, Van Duc Dang, Hien Anh Trinh, Van Can Nguyen (2018), “Improving automatic bounding volume hierarchy to detect collision of rigid bodies in virtual environments”, *Kỷ yếu Hội nghị khoa học quốc gia lần thứ XI về nghiên cứu cơ bản và ứng dụng Công nghệ thông tin*, tr. 209-214, DOI: 10.15625/vap.2018.00027.
- [CT2] **Nghiêm Văn Hưng**, Đặng Văn Đức, Trịnh Hiền Anh, Hoàng Việt Long, Nguyễn Văn Căn (2019), “Kỹ thuật phát hiện nhanh và chạm của vải trong thực tại ảo sử dụng kỹ thuật tính toán song song”, *Kỷ yếu Hội nghị khoa học quốc gia lần thứ XII về nghiên cứu cơ bản và ứng dụng Công nghệ thông tin*, tr. 239-246, DOI: 10.15625/vap.2019.00031.
- [CT3] **Nghiêm Văn Hưng**, Đặng Văn Đức, Trịnh Hiền Anh, Nguyễn Văn Căn, Hoàng Việt Long (2019), “Kỹ thuật phát hiện va chạm của vải trong thực tại ảo sử dụng khối bao k-DOP trên hệ thống CPU đa nhân”, *Kỷ yếu Hội thảo khoa học quốc gia lần thứ XXII "Một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông"*, tr. 144-150.
- [CT4] **Nghiêm Văn Hưng**, Đặng Văn Đức, Nguyễn Văn Căn, Hoàng Việt Long, Trịnh Hiền Anh (2022), “Một cách tiếp cận ứng dụng quy hoạch tuyến tính trong bài toán phát hiện va chạm của các mô hình 3D”, *Kỷ yếu Hội nghị khoa học quốc gia lần thứ XV về nghiên cứu cơ bản và ứng dụng CNTT*, tr. 510-516, DOI: 10.15625/vap.2022.0262.
- [CT5] **Van Hung Nghiem**, Van Duc Dang, Van Can Nguyen, Viet Long Hoang, Hien Anh Trinh (2022), "Fast Collision Detection Technique of Interactive Fabrics in Virtual Reality Environment", *Res. Dev. Inf. Commun. Technol.*, vol. 12/2022, no. 2, pp. 49-56, 2022, ISSN: 1859-3526, DOI: 10.32913/mic-ict-research-vn.v2022.n2.1128.
- [CT6] **Van Hung Nghiem**, Van Duc Dang, Van Can Nguyen, Hien Anh Trinh (2023), "Collision Detection for 6-DoF Collaborative Robot Arm", *Res. Dev. Inf. Commun. Technol.*, vol. 12/2023, no. 2, pp. 73-79, 2023, ISSN: 1859-3526, DOI: 10.32913/mic-ict-research-vn.v2023.n2.1236.

**DANH MỤC TÀI LIỆU THAM KHẢO**

- [1] Tang M., Curtis S., Yoon S. E., 2019, ICCD: interactive continuous collision detection between deformable models using connectivity-based culling, *IEEEs Trans on Visualization and Computer Graphics*, 15(4), pp. 543-558.
- [2] Wozniak M., Pascher M., Ikeda B., Luebbbers M., Jena A., 2024, Virtual, Augmented, and Mixed Reality for Human-Robot Interaction (VAM-HRI), *Proceeding of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI' 2024)*, pp. 1361-1363, DOI: 10.1145/3610978.3638158.
- [3] Tong W., Curtis S., Yoon S. E., 2019, VCD: interactive continuous collision detection between deformables models using connectivity-based culling, *IEEEs Trans on Visualization and Computer Graphics*, 15(4), pp. 543-558.
- [4] Cieri R. L., Turner M. L., et al., 2021, Virtual and augmented reality: New tools for visualizing, analyzing, and communicating complex morphology, *J. Morphol.*, 282(12), pp. 1785-1800.
- [5] Wedel J., Štrákl M., Hriberšek M., et al., 2024, A novel particle-particle and particle-wall collision model for superellipsoidal particles, *Computational Particle Mechanics*, 11, pp. 211-234.
- [6] Wang X., et al., 2018, Efficient BVH-based Collision Detection Scheme with Ordering and Restructuring, *Journal of Computer Graphics Forum*, 37(2), pp. 227-237.
- [7] Fernández-Layos P., Merchante L., 2024, Convex Body Collision Detection Using the Signed Distance Function, *Computer-Aided Design*, 170, pp. 1-16.
- [8] Wang M., Cao J., 2021, A review of collision detection for deformable objects, *Computer Animation and Virtual Worlds*, 32(5), pp.1-23.

- [9] Curtis S., Tamstorf R., Manocha D., 2008, Fast collision detection for deformable models using representative-triangles, *Proceedings of the 2008 Symposium on Interactive 3D Graphics (SI3D' 2008)*, pp. 61-69.
- [10] Tang M., Curtis S., Yoon S. E., 2019, ICCD: interactive continuous collision detection between deformable models using connectivity-based culling, *IEEE Trans on Visualization and Computer Graphics*, 15(4), pp. 544-557.
- [11] Brochu T., Edwards E., Bridson R., 2012, Efficient geometrically exact continuous collision detection, *ACM Transactions on Graphics*, 31(4), pp. 1-7.
- [12] Tang M., Tong R. F., Wang Z. D., et al., 2014, Fast and exact continuous collision detection with bernstein sign classification, *ACM Transactions on Graphics*, 33(6), pp. 1-8.
- [13] Wang H. M., 2014, Defending continuous collision detection against errors, *ACM Transactions on Graphics*, 33(4), pp. 1-10.
- [14] Tan T., Weller R., Zachmann G., 2020, OpenCollBench - Benchmarking of Collision Detection & Proximity Queries as a Web-Service, *Proceedings of the 25th International Conference on 3D Web Technology*, pp. 1-9.
- [15] Du P., Tang M., Tong R. F., 2012, Fast continuous collision culling with deforming noncollinear filters, *Computer Animation and Virtual Worlds*, 23(3), pp. 375-383.
- [16] Tang M., Manocha D., Tong R. F., 2010, Fast continuous collision detection using deforming non-penetration filters, *Proceedings of the SIGGRAPH Symposium on Interactive 3D Graphics and Games 2010*, pp. 7-13.
- [17] Tang M., Curtis S., Yoon S. E., 2019, ICCD: interactive continuous collision detection between deformable models using connectivity-based

- culling, *IEEEs Trans on Visualization and Computer Graphics*, 15(4), pp. 543-558.
- [18] Tang M., Curtis S., Yoon S. E., 2019, ICCD: interactive continuous collision detection between deformable models using connectivity-based culling, *IEEEs Trans on Visualization and Computer Graphics*, 15(4), pp. 543-558.
- [19] Tang M., Curtis S., Yoon S. E., 2019, ICCD: interactive continuous collision detection between deformable models using connectivity-based culling, *IEEEs Trans on Visualization and Computer Graphics*, 15(4), pp. 543-558.
- [20] Dubosc C., Gorisse G., et al., 2021, Impact of Avatar Facial Anthropomorphism on Body Ownership, Attractiveness and Social Presence in Collaborative Tasks in Immersive Virtual Environments, *Comput. Graph.*, 101, pp. 82-92.
- [21] Marques B., Silva S., et al., 2022, A Conceptual Model and Taxonomy for Collaborative Augmented Reality, *IEEE Transactions on Visualization and Computer Graphics*, 28(12), pp. 5113-5133.
- [22] Badia S., Silva P., Branco D., Pinto A., Carvalho, C., Menezes, P., Almeida J., Pilacinski A., 2022, Virtual Reality for Safe Testing and Development in Collaborative Robotics: Challenges and Perspectives, *Electronics*, 11, 1726, doi: 10.3390/electronics11111726.
- [23] Proia S., Carli R., Cavone G., 2022, Control techniques for safe, ergonomic, and efficient human - robot collaboration in the digital industry: A survey, *IEEE Transactions on Automation Science and Engineering*, 19(3), pp.1798-1819, DOI: 10.1109/TASE.2021.3131011.
- [24] N. V. Thương, T. X. Thịnh, 2023, Các phương pháp nâng cao an toàn cho robot cộng tác người - máy, *Tạp chí Khoa học và Công nghệ Việt Nam (ISSN: 1859-4794)*, số 65 năm 2023, tr. 30-34.



- [25] Czubenko M., Kowalczyk Z., 2021, A simple neural network for collision detection of collaborative robots, *Sensors*, 21(12), pp. 4235-4254.
- [26] Jiang Y., Wu Y., Wu X., Zhao B., 2023, Research on Collision Detection of Collaborative Robot using improved Momentum-based Observer, *Proceedings of the 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET' 2023)*, pp. 1-6.
- [27] Park K. M., Kim J., Park J., Park F. C., 2021, Learning-Based Real-Time Detection of Robot Collisions Without Joint Torque Sensors, *IEEE Robotics and Automation Letters*, 06(01), pp. 103-110.
- [28] Zurlo D., Heitmann T., Morlock, M., De Luca A., 2023, Collision Detection and Contact Point Estimation Using Virtual Joint Torque Sensing Applied to a Cobot, *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA' 2023)*, pp. 7533-7539.
- [29] Haddadin S., De Luca A., Albu-Schäffer A., 2017, Robot collisions: A survey on detection, isolation, and identification, *IEEE Trans. Robot.*, 33(06), pp. 1292-1312.
- [30] Cirillo A., Ficuciello F., Natale C., Pirozzi S., Villani L., 2016, A conformable force/tactile skin for physical human–robot interaction, *IEEE Robot. Autom. Lett.*, 01(01), pp. 41-48.
- [31] Strohmayer M., Wörn H., Hirzinger G., 2013, The DLR artificial skin Step i: Uniting sensitivity and collision tolerance, *Proceedings of the IEEE Int. Conf. Robot. Autom.*, pp. 1012-1018.
- [32] Tang M., Curtis S., Yoon S. E., 2019, ICCD: interactive continuous collision detection between deformable models using connectivity-based culling, *IEEEs Trans on Visualization and Computer Graphics*, 15(4), pp. 543-558.
- [33] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning

- and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [34] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1145-1157.
- [35] Narukawa K., Yoshiike T., Tanaka K., Kuroda M., 2017, Real-time collision detection based on one class SVM for safe movement of humanoid robot, *Proceedings of the IEEE-RAS 17th Int. Conf. Humanoid Robot*, pp. 791-796.
- [36] Heo Y. J., Kim D., Lee W., Kim H., Park J., Chung W. K., 2019, Collision detection for industrial collaborative robots: A deep learning approach, *IEEE Robot. and Autom. Lett.*, 04(02), pp. 740-746.
- [37] Bröhl C., Nelles J., et al., 2016, TAM Reloaded: A Technology Acceptance Model for Human-Robot Cooperation in Production Systems, *Proceedings of the HCI International 2016 - Posters' Extended Abstracts*, pp. 97-103.
- [38] Weistroffer V., Paljic A., 2013, A Methodology to Assess the Acceptability of Human-Robot Collaboration Using Virtual Reality, *Proceedings of the the 19th ACM Symposium on Virtual Reality Software and Technology*, pp. 39-48.
- [39] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [40] Burdea G. C., Coiffet P., 2024, *Virtual reality technology - 3rd Edition*, Wiley-IEEE Press, pp. 3-5.
- [41] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning

- and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [42] Weller R., 2013, *New geometric data structures for collision detection and haptics*, Springer, pp. 9-46.
- [43] Bergen G., 2017, Efficient collision detection of complex deformable models using AABB trees, *Journal of Graphics Tools*, 29(04), pp. 1-14.
- [44] Hubbard P. M., 2016, Approximating polyhedra with spheres for time-critical collision detection, *ACM Transaction on Graphics*, 95(3), pp. 179-210.
- [45] Gottschalk S., et al., 2020, Obbtree: A hierarchical structure for rapid interference detection, *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 171-180.
- [46] Klosowski J. T., Held M., Mitchell J. S. B., Sowizral H., Zikan K., 2018, Efficient collision detection using bounding volume hierarchies of k-DOPs, *IEEE Trans. on Visualization and Computer Graphics*, 49(1), pp. 21-37.
- [47] Mariotti E., Magrini E., Luca A. D., 2019, Admittance control for human-robot interaction using an industrial robot equipped with a F/T sensor, *Proceedings of the 2019 Int. Conf. Robot. Automat.*, pp. 6130-6136.
- [48] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [49] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.

- [50] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [51] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [52] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [53] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [54] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [55] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [56] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.

- [57] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [58] ISO 10218-1/2:2011, 2011, *Robots and robotic devices - Safety requirements for industrial robots*, International Organization for Standardization. Geneva, Switzerland.
- [59] Villani V., Pini F., Leali F., Secchi C., 2018, Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications, *Mechatronics*, 55, pp. 248-266.
- [60] Lynch K. M., Park F. C., 2017, *Modern Robot*. Cambridge University Press, pp. 100-115.
- [61] Gaz C., Magrini E., De Luca A., 2018, A model-based residual approach for human-robot collaboration during manual polishing operations, *Mechatronics*, 55, pp. 234-247.
- [62] Smola A. J., Schölkopf B., 2004, A tutorial on Support Vector Regression, *Statist. Comput.*, 14(03), pp. 199-222.
- [63] LeCun Y., et al., 2018, Gradient-based learning applied to document recognition, *Proc. IEEE*, 86(11), pp. 2278-2324.
- [64] Palaz D., Collobert R., Doss M. M., 2013, Estimating phoneme class conditional probabilities from raw speech signal using Convolutional Neural Networks, *Proc. Interspeech*, pp. 1766-1770.
- [65] Genton M. G., 2001, Classes of kernels for machine learning: A statistics perspective, *J. Mach. Learn. Res.*, 02, pp. 299-312.
- [66] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.

- [67] Tang M., Manocha D., Tong R. F., 2010, MCCD: Multi-core collision detection between deformable models using front-based decomposition, *Graphical Models*, 72(02), pp. 7-23.
- [68] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [69] Kowalczyk, Z., Czubenko, M., Merta, T., 2020, Interpretation and modeling of emotions in the management of autonomous robots using a control paradigm based on a scheduling variable, *Eng. Appl. Artif. Intell.*, 91, pp. 103-120.
- [70] Masłowski, D., Czubenko, M., 2019, Safety system for an industrial cooperating robot based on depth cameras, *Pomiary Autom. Robot.*, 23, pp. 41–46.
- [71] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [72] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [73] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [74] Villagrossi, E., Simoni, L., Beschi, M., Pedrocchi, N., Marini, A., Tosatti, L.M., Visioli, A., 2018, A virtual force sensor for interaction tasks with conventional industrial robots, *Mechatronics*, 50, pp. 78–86.

- [75] Kowalczyk, Z., Czubenko, M., 2021, Intelligent decision-making system for autonomous robots, *Int. J. Appl. Math. Comput. Sci.*, 21, pp. 621–635.
- [76] Varhegyi, T., Melik-Merkumians, M., Steinegger, M., Halmetschlager-Funek, G., Schitter, G., 2018, A Visual Servoing Approach for a Six Degrees-of-Freedom Industrial Robot by RGB-D Sensing. *Autom. Robot., Mechatronics*, 50, pp. 108–120.
- [77] Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y., 2018, OpenPose: Realtime multi-person 2D pose estimation using Part Affinity Fields, *Mechatronics*, 51, pp. 72–85.
- [78] Cho, C.N., Hong, J.T., Kim, H.J., 2019, Neural network based adaptive actuator fault detection algorithm for robot manipulators, *J. Intell. Robot. Syst.*, 95, pp. 137–147.
- [79] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [80] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [81] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [82] Zhung Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.

- [83] Zhang Z., Qian K., Schuller B.W., Wollherr D., 2021, An Online Robot Collision Detection and Identification Scheme by Supervised Learning and Bayesian Decision Theory, *IEEE Trans. Autom. Sci. Eng.*, 18(03), pp. 1144-1156.
- [84] Xu, T., Fan, J., Fang, Q., Zhu, Y., Zhao, J., 2020, A new robot collision detection method: A modified nonlinear disturbance observer based-on neural networks, *J. Intell. Fuzzy Syst.*, 38, pp. 175–186.
- [85] Xu, T., Fan, J., Fang, Q., Zhu, Y., Zhao, J., 2020, A new robot collision detection method: A modified nonlinear disturbance observer based-on neural networks, *J. Intell. Fuzzy Syst.*, 38, pp. 175–186.
- [86] Wu, J., Wang, J., You, Z., 2022, An overview of dynamic parameter identification of robots, *Robot. Comput. Integr. Manuf.*, 116, pp. 414–419.
- [87] Zeng, F., Xiao, J., Liu, H., 2023, Force/torque sensorless compliant control strategy for assembly tasks using a 6-DOF collaborative robot, *IEEE Access*, 174, pp. 108795–108805.
- [88] Park, K.J., 2021, Fourier-based optimal excitation trajectories for the dynamic identification of robots, *Robotica*, 34, pp. 625–633.
- [89] Indri, M., Trapani, S., Lazzero, I., 2020, Development of a virtual collision sensor for industrial robots, *Sensors*, 107, pp. 11-48.
- [90] Li, Z., Ye, J., 2021, A virtual sensor for collision detection and distinction with conventional industrial robots, *Sensors*, 29, pp. 23-40.
- [91] Xu, T., Fan, J., Fang, Q., Zhu, Y., Zhao, J., 2020, A new robot collision detection method: A modified nonlinear disturbance observer based-on neural networks, *J. Intell. Fuzzy Syst.*, 38, pp. 175–186.
- [92] He, S., Ye, J., Li, Z., Li, S., Wu, G., Wu, H., 2015, A momentum-based collision detection algorithm for industrial robots, *Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1253–1259.



- [93] Han, L., Xu, W., Li, B., Kang, P., 2022, Collision Detection and Coordinated Compliance on Momentum Observer, *IEEE/ASME Trans. Mechatron.*, 24, pp. 2261–2272.
- [94] Guo, M., Zhang, H., Feng, C., Liu, M., Huo, J., 2020, Manipulator residual estimation and its application in collision detection, *Ind. Robot Int. J.*, 45, pp. 354–362.
- [95] Popov, D., A., Mavridis, N., 2023, Collision detection, localization & classification sensors, *Proceedings of the 26th IEEE International Symposium on RO-MAN*, pp. 838–843.
- [96] Chen, S., 2023, A universal algorithm for sensorless collision detection of robot actuator faults, *Proceedings RO-MAN*, pp. 910-932.
- [97] Sneider, H., Frank, P.M., 2016, Observer-based residual evaluation, *IEEE Trans. Control Syst. Technol.*, 44, pp. 274–282.
- [98] Zhang, T., Ge, P., Zou, Y., He, Y., 2021, Robots Collision Detection on Time-Series Analysis, *J. Dyn. Syst. Meas. Control*, 143, pp. 41-63.
- [99] Birjandi, S.A.B., Kühn, J., Haddadin, S., 2022, Observer-extended direct method for IMU sensing, *IEEE Robot. Autom. Lett.*, 59, pp. 954–961.
- [100] Moon, F., Wang, G., Liu, N., 2019, Collision detection and identification on robot, *Sensors*, 19, pp. 80-112.
- [101] Wooten, J.M., Bevely, D.M., Hung, J.Y., 2024, Piezoelectric Polymer-Based CD for Robotic Applications, *Electronics*, 45, pp. 204–220.
- [102] Pang, G., Deng, J., Wang, F., Zhang, J., Pang, Z., Yang, G., 2018, Development of flexible robot skin for safe and natural human–robot collaboration, *Micromachines*, 9, pp. 57-80.
- [103] Rout, R., Cui, R., Han, Z., 2020, Modified Line-of-Sight Guidance Law With Adaptive Neural Network Control of Underactuated Marine Vehicles With State and Input Constraints, *IEEE Trans. Control Syst. Technol.*, 28, pp. 1902–1914.

- [104] Jiang, X., 2019, *Time varying coefficient AR and VAR models*, Springer: Berlin/Heidelberg, Germany, pp. 175–191.
- [105] Selvinj, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V.K., Soman, K., 2023, LSTM and CNN-sliding window model, *Proceedings of the ICACCI*, pp. 1643–1647.
- [106] Kim, T.Y., Cho, S.B., 2019, Predicting residential energy consumption using CNN-LSTM neural networks, *Energy*, 182, pp. 72–81.
- [107] Grafarend, E., Awange, J., 2024, *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Total Least Squares*, Springer: Berlin/Heidelberg, Germany.
- [108] Xu, H., Yu, Z., Lue, X., Wang, S., Li, S., Wang, S., 2020, Model predictive control-based path tracking control for automatic guided vehicles, *CVCI*, pp. 627-632.
- [109] Zhou C., Wang Y., 2022, Obstacle avoidance strategy for an autonomous deep deterministic policy gradient, *Engineering*, 24(3), pp. 110-134.
- [110] Phalae K., Simons E., 2016, Monitoring system based on ISO/IEC/IEEE 214501 standard, *IEEE Sensors Journal*, 16(12), pp. 5037-5045.
- [111] Razaai A., et al., 2022, A Review on Path Planning and Obstacle Avoidance for Autonomous Mobiles Robot, *Journal of Robotics.*, 22, pp. 1-14.
- [112] Khatijb O., 2023, Motioning coordination and reactive control of autonomous multi-manipulator system, *Journal Robotic Systems*, 35(4), pp. 300-319.
- [113] Khatib O., Yokoi K., Brock O., Chang K., Casal A., 2023, Robots in human capabilities, *International Journal of Robotics Research*, 112(7), pp. 684-696.