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

TÓM TẮT LUẬN ÁN TIẾN SĨ MÁY TÍNH

Ngành: Hệ thống thông tin

Mã số: 9 48 01 04

Hà Nội - 2024

Công trình được hoàn thành tại: Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Người hướng dẫn khoa học: PGS.TS. Đặng Văn Đức, Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Phản biện 1: …….

Phản biện 2: …….

Phản biện 3: …….

Luận án được bảo vệ trước Hội đồng đánh giá luận án tiến sĩ cấp Học viện họp tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam vào hồi …… giờ ……, ngày …… tháng …… năm 20…

Có thể tìm hiểu luận án tại:

1. Thư viện Học viện Khoa học và Công nghệ
2. Thư viện Quốc gia Việt Nam

MỞ ĐẦU

1. Tính cấp thiết của luận án

Cuộc Cách mạng công nghiệp lần thứ tư với những tiến bộ đột phá của khoa học và công nghệ 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.

Phát hiện va chạm (*Collision Detection*) là một trong những tác vụ cơ sở của các hệ thống mô phỏng VR, đồ họa máy tính, điều khiển robotics, games,... Các đối tượng trong mỗi hệ thống có những chuyển động riêng của nó, chuyển động đó có thể va chạm với đối tượng khác, hoặc có thể va chạm với môi trường, chướng ngại vật.

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 (*Bounding Volume Hierarchies - BVH*) hoặc dựa trên kỹ thuật tính toán hàm khoảng cách (*Signed Distance Function - SDF*). Đối với các mô hình vật thể biến dạng (như chất liệu vải) thì các quá trình duyệt, tái cấu trúc BVH và tính toán SDF cần sử dụng rất nhiều tài nguyên của hệ thống gây ra tình trạng *nghẽn nút cổ chai*. Vấn đề đặt ra là cần thiết kế thuật toán có thể hoạt động tốt trên các mô hình vật thể khác nhau.

Tại Việt Nam, những nghiên cứu về VR chủ yếu trong việc mô hình hóa hay mô phỏng trưng bày ảo mà chưa có nhiều nghiên cứu về bài toán phát hiện va chạm trong vật thể biến dạng và cánh tay cobot. Các nghiên cứu về phương pháp phát hiện va chạm của mô hình vật thể biến dạng như 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.

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. Đối với lĩnh vực robotics, robot cộng tác (*Collaborative robot - Cobot*) đang dần thay thế robot truyền thống. 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.

Để phát huy hiệu 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. Vì vậy, nghiên cứu sinh đã chọn đề tài "***Nghiên cứu một số kỹ thuật phát hiện va chạm trong vật thể biến dạng và cánh tay cobot***" làm đề tài luận án của mình.

2. Mục tiêu nghiên cứu

Mục tiêu của luận án là đề xuất phương pháp để nâng cao hiệu quả phát hiện va chạm trong vật thể biến dạng và cánh tay cobot 6 bậc tự do. Kết quả nghiên cứu được sử dụng trong các ứng dụng mô phỏng VR và lĩnh vực điều khiển cánh tay cobot 6 bậc tự do.

3. Nội dung nghiên cứu

Để thực hiện được mục tiêu đặt ra, luận án thực hiện các nội dung nghiên cứu sau: Công nghệ VR và cánh tay cobot; bài toán phát hiện va chạm của các đối tượng mô hình 3D (vật thể rắn và vật thể chất liệu vải) trong môi trường ảo; bài toán phát hiện va chạm của cánh tay cobot 6 bậc tự do; một số thuật toán và phương pháp phát hiện va chạm; đề xuất cải tiến phương pháp phát hiện va chạm, thử nghiệm và đánh giá kết quả nghiên cứu.

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

1.1. Giới thiệu

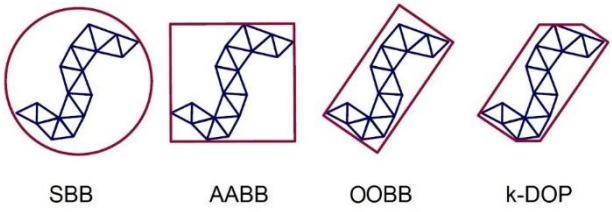
Bài toán phát hiện va chạm trong môi trường thực tại ảo được phát biểu như sau: Cho n đối tượng $\{O_0, O_1, \dots, O_{n-1}\}$, phát hiện va chạm là quá trình xác định xem các đối tượng có giao nhau hay không, tức là kiểm tra $O_i \cap O_j, \forall i \neq j,\ i = 0, \dots, n-1, j = 0, \dots, n-1$ và $n \in \mathbb{N}$. Trong đó, $O_i = \{p_0^i, p_1^i, \dots, p_k^i\}$ là một tập gồm k mặt cơ sở 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 $k, m \in \mathbb{N}$.

Kết quả trả về của bài toán phát hiện va chạm ở dạng Có/Không, nếu có va chạm xảy ra thì cần tính toán các điểm tiếp xúc và thời điểm tiếp xúc.

1.2. Tổng quan các nghiên cứu liên quan

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

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 1.1 cho thấy một số loại khối bao được sử dụng trong các kỹ thuật phát hiện va chạm.



Hình 1.1. Một số loại khối bao được sử dụng phổ biến trong các kỹ thuật phát hiện va chạm

Hình 1.1 mô tả một số dạng khối bao thường dùng trong các kỹ thuật phát hiện va chạm, trong đó 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.

1.2.2. 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 cảm biến được phát triển để đáp ứng nhu cầu này.

1.3. Một số hạn chế của các phương pháp truyền thống

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

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 OOBB. Tuy nhiên, các hình bao dạng OOBB lại cho kết quả chân thực hơn vì nó mô phỏng chính xác hơn thể tích của vật thể.

Với hai khối bao OOBBs, 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}. 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.

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

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].

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 bổ của nó, nên khó tích hợp với các thuật toán khác. 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 [15] và [16] kém hiệu quả vì không xử lý được trường hợp lưới tam giác bị lật.

1.3.3. 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 việc sử dụng các 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.

1.4. Kiến thức nền tảng

1.4.1. Thực tại ảo

1.4.2. Cấu tạo mô hình 3D

1.4.3. Mô hình vật thể biến dạng

1.4.4. Quy tắc dấu Descartes và Định lý Vincent

Quy tắc dấu Descartes [16]: Gọi $f(t)$ là một đa thức theo lũy thừa giảm dần của t. Biến đổi dấu xảy ra bất cứ khi nào các hệ số khác không liền kề là khác dấu. Ký hiệu $VAR\big(f(t)\big)$ là số cặp hệ số khác không liền kề 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 $t \to \frac{1}{t+1}$.

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

1.4.5. Cobot và cấu tạo của cánh tay cobot

Các bộ phận chủ yếu của một cobot 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).

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:

$$Precision = \frac{TP}{TP+FP}$$

và:

$$Recall = \frac{TP}{TP+FN}$$

với *TP* là số lượng dương tính thật, *FP* là số lượng dương tính giả và *FN* là số lượng âm tính giả (va chạm được coi là dương tính).

1.5.2. Tiêu chí đánh giá các loại khối bao

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.

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.

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.

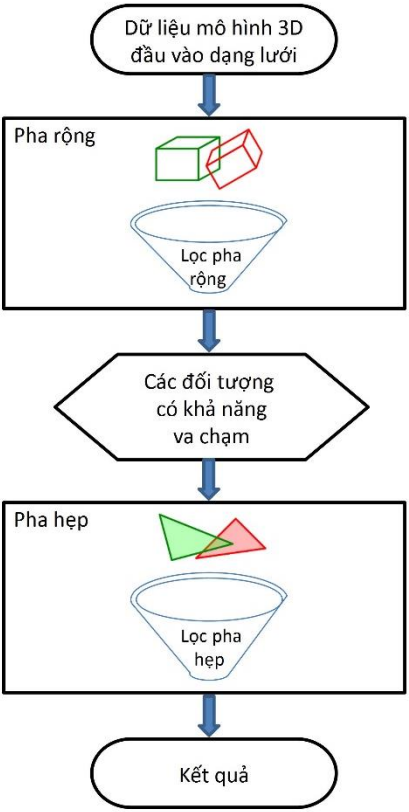
Một phần kết quả nghiên cứu trình bày tại chương 1 đã được NCS công bố tại [CT1, CT2, CT3] trong Danh mục công trình công bố liên quan đến luận án.

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

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

NCS kế thừa và phát triển trên cơ sở giải pháp lọc 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. Quá trình phát hiện va chạm được chia thành hai pha là *pha rộng* và *pha hẹp* như minh họa trong Hình 2.1.



Hình 2.1. Sơ đồ tổng quát thuật toán đề xuất

Thuật toán 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: Tạo BV;

2: Thiết lập cấu trúc phân cấp BVH theo chiều từ trên xuống;

3: Thiết lập cấu trúc cây BVTT dựa vào phép duyệt cấu trúc phân cấp BVH;

4: Kiểm tra sự chồng lấp, loại bỏ các cặp cặp đối tượng không có khả năng xảy ra va chạm tương ứng là với nút lá của cây BVTT.

4.1: if không chồng lấp thì exit;

4.2: else (có khả năng xảy ra va chạm) thì chuyển sang thực hiện Algorithm2.

Thuật toán 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 (thông tin điểm tiếp xúc, vectơ pháp tuyến của mặt tiếp xúc,…) hoặc kết luận không va chạm.

1: Kiểm tra và thực hiện lọc dựa vào các đặc điểm hình học (tính đồng phẳng của cặp VF hoặc cặp EE);

2: Tính VAR(f(x)) để phân tích sự tồn nghiệm trong [0, 1];

3: Áp dụng quy tắc dấu Descartes và định lý Vincent

3.1: if không tồn tại nghiệm thì loại bỏ và kết luận các cặp VF (hoặc cặp EE) không xảy ra va chạm;

3.2: else (có tồn tại nghiệm) áp dụng định lý Vincent và giải phương trình để kết luận va chạm (thông tin điểm tiếp xúc, vectơ pháp tuyến của mặt tiếp xúc,…).

Pha hẹp tính toán để xác định chính xác có xảy ra sự va chạm hay không. Việc đầu tiên là thuật toán 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.

2.1.2. Kết quả và thảo luận

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.

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

(a) (b) (c)

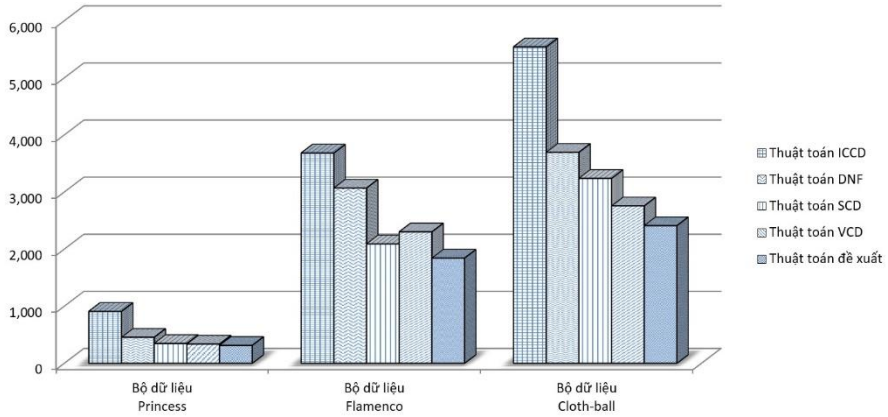
Hình 2.2. Minh họa trên các bộ dữ liệu trong thư viện mở GAMMA: (a) Bộ dữ liệu Princess, (b) Bộ dữ liệu Flamenco, (c) Bộ dữ liệu Cloth-ball

Kết quả 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 thuật toán đề xuất được so sánh với thuật toán ICCD của Tang và cộng sự [10], thuật toán DNF của Tang và cộng sự [16], thuật toán SCD của Zhang và cộng sự [17] và thuật toán VCD của Zhang và cộng sự [18] với đơn vị thời gian phát hiện va chạm là *ms*.

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 2.3. Kết quả thực thi các thuật toán

Biểu đồ so sánh trong Hình 2.3 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 thời gian cho các thao tác giải phương trình, giảm số lượng phép toán phức tạp.

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

2.2.1. Phương pháp đề xuất

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.

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: Danh sách L chứa thông tin về các va chạm, nếu không có va chạm thì L rỗng.

1: Bắt đầu duyệt với nút con trái đầu tiên trong cây T;

2: Lặp với các nút khác (gọi M là nút đang xét)

2.1: Thực hiện phép kiểm tra nhanh xem nút đang xét có nằm trong vùng không gian Cl hay không

2.2: if nút đang xét không nằm trong Cl thì return false;

2.3: else (P và M có khả năng xảy ra va chạm)

Thực hiện thủ tục ***Test()*** để return true nếu có va chạm và lưu trữ, return false nếu không có va chạm;

Nếu phát hiện có va chạm, tiếp tục kiểm tra giữa các nút con của P và M;

Chừng nào các nút đang xét là nút lá thì thuật toán đã duyệt đến hai nút sâu nhất trên cây va chạm với nhau.

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.

Mô tả thủ tục kiểm tra va chạm như sau:

```
Test
Đầu vào: 02 tham số (P, Q là các đối tượng cần kiểm tra va chạm).
Đầu ra: Trả về true nếu có va chạm, trả về false nếu không có va chạm.
1: Khởi tạo một biến để lưu trữ số lượng bất phương trình;
2: Lấy cơ sở của Q thể hiện trong cơ sở của P;
3: Khởi tạo các hệ số của ma trận biểu diễn hệ bất phương trình;
4: Loại bỏ biến đầu tiên trong hệ ban đầu;
5: Tiến hành đệ quy với các biến tiếp theo;
6: Chuyển đổi các giá trị giới hạn trong cơ sở của P và trả về kết quả tương ứng.
```

2.2.2. Kết quả và thảo luận

Thử nghiệm thứ nhất

Kịch bản thử nghiệm: 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. Cuối cùng, NCS tính toán tỷ lệ (*thời gian thực hiện thuật toán SAT*)/(*thời gian thực hiện thuật toán đề xuất*) như trong Bảng 2.1.

Bảng 2.1. 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 2.1 luôn có giá trị hơn 1, điều đó có nghĩa là thuật toán đề xuất nhanh hơn thuật toán SAT.

Thử nghiệm thứ hai

Kịch bản thử nghiệm: 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. 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.

NCS đã 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 trình bày trong Bảng 2.2.

Bảng 2.2. 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.

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.

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 va 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 xử lý 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

3.1. Phương pháp đề xuất

3.1.1. Phương pháp phát hiện va chạm

Trên cơ sở phương pháp SVR [62] làm nền tảng, NCS nghiên cứu 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_I trên một cửa sổ thời gian có kích thước t_W; số lượng mẫu là $N + 1 = {}^{t_W}/_{t_I} + 1 \in \mathbb{N}$.

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).

Phát hiện va chạm bằng SVR

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, 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:

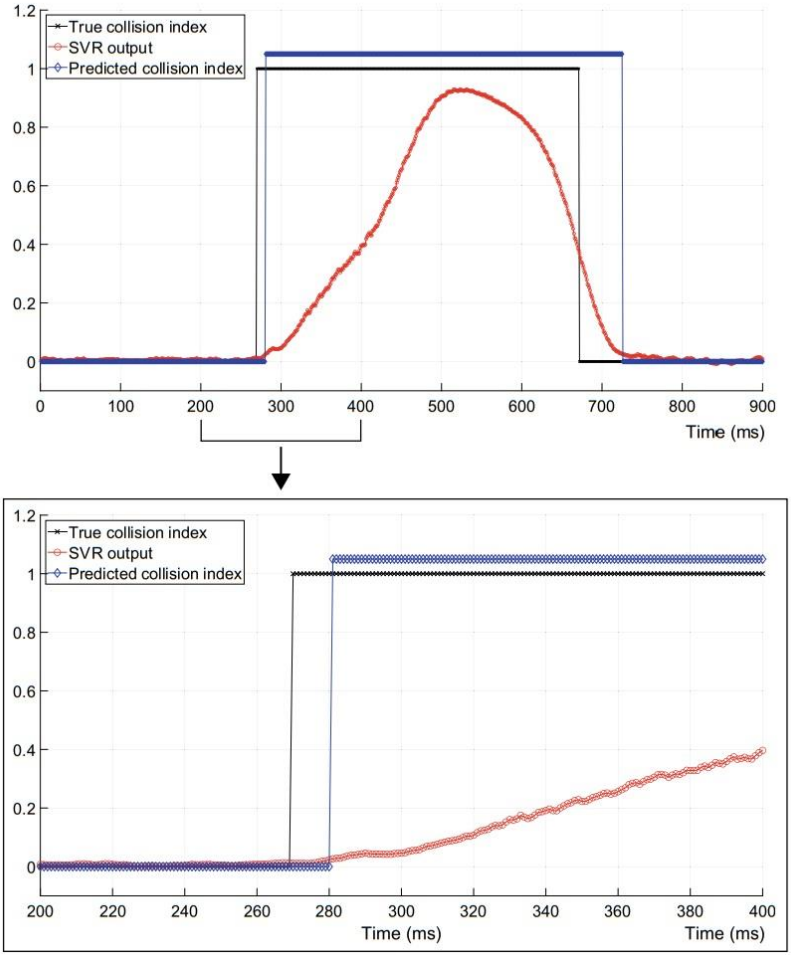
$$x = [rj_1 \quad rj_2 \quad rj_3 \quad rj_4]^T \in \mathbb{R}^{4(N+1)} \qquad (3.1)$$

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 $\sigma = 3$.



Hình 3.1. 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.

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 từ 500 g đến 4000 g) [25]. Trong đó, dữ liệu huấn luyện (90%) và dữ liệu thử nghiệm (10%).

3.2. Kết quả và thảo luận

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).

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 3.2. 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).

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 3.5. Đánh giá kết quả và so sánh với M. 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.

3.3. Kết luận chương 3

Phương pháp đề xuất dựa trên SVR đã phát hiện va chạm của cánh tay cobot 6-DoF CURA6 khá 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

Luận án có kết quả chính như sau:

- Đề xuất thuật toán phát hiện va chạm AB2PF 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 hơn các phương pháp truyền thống.

- Đề xuất thuật toán ABFME 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.

- 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-DoF.

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 (vật thể rắn và 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.

DANH MỤC CÁC BÀI BÁO ĐÃ XUẤT BẢN LIÊN QUAN ĐẾN LUẬN ÁN

1. **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 CNTT*, tr. 209-214, DOI: 10.15625/vap.2018.00027.
2. **Nghiêm Văn Hưng**, Đặng Văn Đức, Trịnh Hiền Anh, Hoàng Việt Long, Nguyễn Văn Căn (2019), "Kỹ thuật phát hiện nhanh va 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 CNTT*, tr. 239-246, DOI: 10.15625/vap.2019.00031.
3. **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.
4. **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 công nghệ thông tin*, tr. 510-516, DOI: 10.15625/vap.2022.0262.
5. **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.
6. **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.