

Kỹ thuật phát hiện nhanh và chạm của chất liệu vải tương tác trong môi trường thực tại ảo

Nghiêm Văn Hưng^{1,2}, Đặng Văn Đức³, Nguyễn Văn Căn², Hoàng Việt Long², Trịnh Hiền Anh³

¹ Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Hà Nội, Việt Nam

² Trường Đại học Kỹ thuật - Hậu cần Công an nhân dân, Bộ Công an, Bắc Ninh, Việt Nam

³ Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam, Hà Nội, Việt Nam

Tác giả liên hệ: Nghiêm Văn Hưng, ngiemvanhung1985@gmail.com

Ngày nhận bài: 29/07/2022, ngày sửa chữa: 12/11/2022, ngày duyệt đăng: 30/11/2022

Định danh DOI: 10.32913/mic-ict-research-vn.v2022.n2.1128

Tóm tắt: 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 bài báo này, chúng tôi trình bày một số cải tiến để phát hiện nhanh va chạm của mô hình vải được biểu diễn dưới dạng mặt tam giác. Sự tương tác của vải với các vật thể khác và với chính nó như gấp, cuộn làm cho quá trình phát hiện va chạm trở nên phức tạp hơn. Kỹ thuật 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.

Từ khóa: Phát hiện va chạm, mô phỏng, mô hình 3D.

Title: Fast Collision Detection Technique of Interactive Fabrics in Virtual Reality Environment

Abstract: Collisions are one of the fundamental problems that need to be dealt with in computer graphics and virtual reality systems. Collision detection is the first step, which is decisive for the system's collision handling tasks. In this paper, we present some improvements for fast collision detection of fabric model represented as triangle mesh. The interaction of the fabric with other objects and with itself, such as folding and rolling, makes the collision detection process more complicated. The proposed technique using the broad-phase filter algorithm and the narrow-phase filter algorithm is tested and gives good results on three different fabric model datasets in GAMMA library, the average collision detection speed is about 2.3 times faster times compared to the method of Tang et al., about 1.34 times faster than the method of Zhang et al.

Keywords: Collision detection, simulation, 3D modeling.

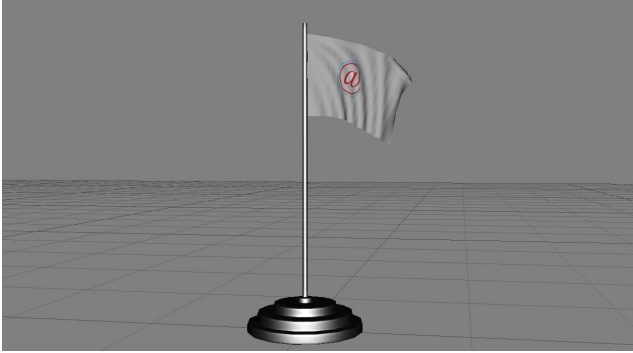
I. GIỚI THIỆU

Phát hiện va chạm là một trong những tác vụ cơ bản của các hệ thống mô phỏng thực tại ảo, đồ họa máy tính, điều khiển robotics, games,... 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. Bài báo này tập trung vào vấn đề phát hiện va chạm của mô hình vải được biểu diễn dưới dạng mặt tam giác. Hình 1 minh họa quá trình va chạm của vải trên một lá cờ có hình biểu trưng của Hội thảo VNICT đang tung bay trong gió.

Kế thừa và phát triển từ các phương pháp của Tang và

cộng sự [10], Zhang và cộng sự [20] về phát hiện va chạm trong các mô hình vật thể biến dạng, kỹ thuật đề xuất trong bài báo này cho phép tăng nhanh tốc độ phát hiện va chạm trong các mô hình vải khác nhau thuộc bộ dữ liệu GAMMA (Geometric Algorithms for Modeling, Motion, and Animation) thuộc Trường Đại học Maryland, Hoa Kỳ.

Phần còn lại của bài báo được cấu trúc như sau: Mục II trình bày về các nghiên cứu liên quan. Mục III trình bày kỹ thuật đề xuất. Kết quả thử nghiệm được trình bày trong mục IV. Mục V phân tích đánh giá, mục VI là kết luận và định hướng nghiên cứu tiếp theo.



Hình 1. Mô phỏng một lá cờ đang tung bay theo gió, trong đó có các phần vải va chạm và tự va chạm

II. CÁC NGHIÊN CỨU LIÊN QUAN

Đã có nhiều công bố cải tiến kỹ thuật phát hiện va chạm trong các mô hình vật thể biến dạng [3, 4, 15, 16], hầu hết đều dựa trên cấu trúc phân hệ vùng bao (Bounding Volume Hierarchies - BVH). Đối với các mô hình vật thể biến dạng như vải thì chi phí 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 trúc biến dạng khác nhau. Curtis và cộng sự [4] 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 [4] 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, Brochu [1], Tang [13] và Wang [14] đưa ra các yêu cầu cao hơn về độ chính xác tính toán của hệ thống.

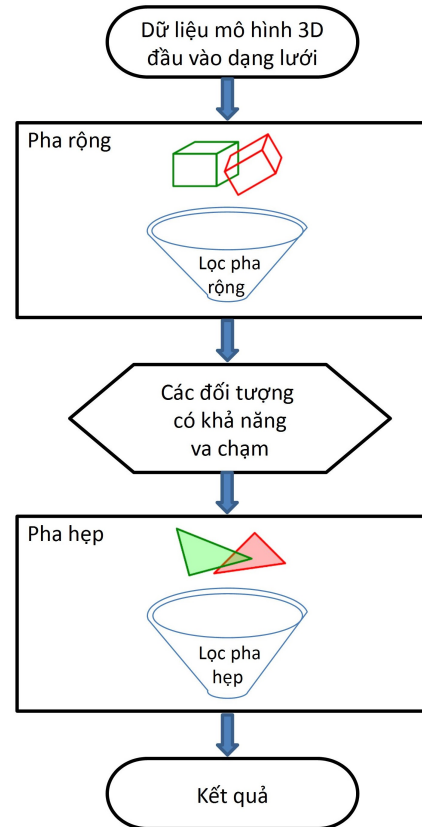
Các hệ thống mô phỏng thực tại ảo thường xuyên phải thực thi các tác vụ phát hiện va chạm. Quá trình phát hiện va chạm chiếm rất nhiều thời gian trong các mô phỏng có sự chuyển động, đồng thời cũng là phần khó và phức tạp về tính toán, thậm chí có thể gây ra tình trạng nghẽn nút cổ chai [5, 9]. Nhìn 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 [15, 16] và hướng thứ hai là sử dụng thuật toán lọc [5, 12, 19, 20]. Tuy nhiên, trong những mô phỏng va chạm mà các mô hình có mức độ biến dạng rất lớn thì phương pháp sử dụng thuật toán lọc trong công bố của Du [5] và Tang [12] tỏ ra kém hiệu quả vì không xử lý được trường hợp lưới tam giác bị lật.

III. KỸ THUẬT ĐỀ XUẤT

1. A. Mô tả kỹ thuật

a) Mô tả chung

Nghiên cứu của chúng tôi kế thừa và phát triển trên cơ sở giải pháp lọc của Tang [10] và Zhang [20] vì tính khả



Hình 2. Sơ đồ tổng quát kỹ thuật đề xuất

thi và phù hợp đối với vật thể chất liệu vải. Kết quả thử nghiệm cho thấy kỹ thuật 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. 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 là đầu vào của pha hẹp. Pha hẹp tính toán để xác định chính xác những đối tượng thực sự va chạm. Chúng tôi 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. Kỹ thuật 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 xử lý 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 ba chiều đầu vào và sử dụng phương pháp ICCD 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 khối bao BV (Bounding Volume) 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 [12] được sử dụng để xác định tính đồng phẳng, và sau đó phương pháp lọc dựa vào phép toán đại số của Zhang [20] được sử dụng để xác định sự tồn tại nghiệm của phương trình khoảng cách.

b) Lọc pha rộng

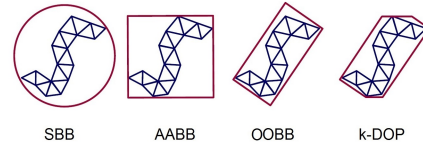
Việc phát hiện va chạm trên các mô hình vải tập trung vào hai vấn đề chủ yếu là các phép kiểm tra cơ sở và tự va chạm.

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 vật thể ba chiều (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 tế ảo sử dụng phương pháp phát hiện va chạm dựa trên các khối bao. Khối bao là một không gian hình học khép kín bao bọc hoàn toàn đối tượng. Hình 3 cho thấy một số loại khối bao được sử dụng trong các kỹ thuật phát hiện va chạm: khối bao căn chỉnh theo trục AABB (Axis Aligned Bounding Box), khối bao hình cầu SBB (Sphere Bounding Box), khối bao xác định theo hướng đối tượng OOB (Object-Oriented Bounding Box) và khối bao đa diện rời rạc có hướng k-DOP (k-Discrete Oriented Polytopes). Bảng I thể hiện kết quả so sánh các loại khối bao phổ biến theo các tiêu chí đánh giá khác nhau.

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 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



Hình 3. Một số loại khối bao được sử dụng trong các kỹ thuật phát hiện va chạm

va chạm và mức độ chính xác của phương pháp phát hiện va chạm,... Trong bài báo này, chúng tôi lựa chọn sử dụng khối bao k-DOP để thỏa hiệp giữa các tiêu chí đánh giá và thuận lợi cho việc 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 [4], Du [5, 6], Tang [10, 12, 13] và Zhang [19, 20].

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. Nếu muốn kiểm tra xem hai tam giác t_a và t_b có cắt nhau trong thời gian ΔT 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.

Chúng tôi cho rằng không cần thiết phải sử dụng đến 15 phép kiểm tra, có thể cải tiến và thay thế bằng một chiến lược mới để giảm số lượng phép kiểm tra. Với t_a và t_b là kề nhau thì 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_h) 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. 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) [4]. Khi BV của hai tam giác chồng lên nhau, về mặt lý thuyết thì cần tiến hành 15 phép kiểm tra. Nhưng trên thực tế, nếu chúng tôi sử dụng các E-BV thì chỉ cần 2 phép kiểm tra.

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 ΔT , đỉnh có chạm vào tam giác hay không. Trong tình huống này, kỹ thuật lọc không thâm nhập là một cách hiệu quả

Bảng I
SO SÁNH CÁC LOẠI KHỐI BAO

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

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. Chiều đỉnh a và p lên pháp tuyến $\vec{n}_t$, nếu trong thời gian ΔT khoảng cách chiều luôn khác không thì đỉnh và tam giác không thể tiếp xúc nhau. Để xác định được khoảng cách này cần sử dụng đến các phép tính toán, tuy nhiên việc tính toán đơn giản hơn nhiều so với việc giải các phương trình bậc ba.

c) 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 $x_i(t)$ và vận tốc không đổi của chúng $v_i(t)$ với $i=(1,2,3,4)$, hàm khoảng cách $d(t)$ giữa đỉnh x_4 và tam giác $x_1x_2x_3$ hoặc giữa cạnh x_1x_2 với cạnh x_3x_4 được xác định như sau:

$$d(t) = [\vec{x}_2(t) - \vec{x}_1(t)] \times [\vec{x}_3(t) - \vec{x}_1(t)] \times [\vec{x}_4(t) - \vec{x}_1(t)] \quad (1)$$

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

$$d(t) = [\vec{x}_{2_1} + tv_{2_1}] \times [\vec{x}_{3_1} + tv_{3_1}] \times [\vec{x}_{4_1} + tv_{4_1}] \quad (2)$$

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

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

trong đó: $a_3 = \vec{x}_{2_1} \times \vec{x}_{3_1}$.

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 $[0,1]$.

Chúng tôi kết hợp thuật toán lọc dựa trên các phép toán

đại số và thuật toán lọc dựa trên các đặc điểm hình học [12] liên quan đến tính đồng phẳng của các cặp VF, EE.

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 tránh được việc phải giải các phương trình bậc ba với chi phí tính toán khá cao. Cụ thể, chúng tôi áp dụng quy tắc dấu Descartes và định lý Vincent để phân ly nghiệm.

Định lý 1. (Quy tắc dấu Descartes) 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 liên tiếp là khác dấu. Ký hiệu $VAR(f(t))$ là số cặp hệ số khác không liên tiếp có dấu ngược nhau. Trên thực tế, ta chỉ quan tâm đến đoạn $[0,1]$. Định lý Vincent là kết quả của quy tắc Descartes về dấu khi ánh xạ t từ $(0,\infty)$ đến $(0,1)$ với ánh xạ $t \mapsto 1/(t+1)$.

Định lý 2. (Định lý Vincent) Xét hàm $g(t) = (t+1)^n f(1/(t+1))$, nếu $VAR(g(t))=0$, 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 $[0,1]$. Về hàm khoảng cách $d(t)$ trong công thức (3) là $f(t)$, theo Định lý 2, $g(t)$ được biểu diễn dưới dạng công thức (4). Áp dụng quy tắc dấu Descartes cho $g(t)$, ta có kết luận như trong Bảng II dưới đây:

Bảng II
SỰ TỒN TẠI CỦA NGHIỆM

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

Từ Bảng II có thể thấy rằng sử dụng Định lý 2 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 (4) có thể được tính bằng a_i với $i=(1,2,3,4)$ trong công thức (3).

$$g(t) = (t+1)^n f(1/t+1) = b_3 t^3 + b_2 t^2 + b_1 t + b_0 \quad (4)$$

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

$$\begin{aligned} g_1(t) &= g(t+1) \\ &= b_3 t^3 + (b_2 + 3b_3)t^2 + (b_1 + 2b_2 + 3b_3)t \\ &\quad + (b_0 + b_1 + b_2 + b_3) \end{aligned} \quad (5)$$

và

$$\begin{aligned} g_2(t) &= (t+1)^3 g(1/t+1) \\ &= b_0 t^3 + (b_1 + 3b_0)t^2 + (b_2 + 2b_1 + 3b_0)t \\ &\quad + (b_0 + b_1 + b_2 + b_3) \end{aligned} \quad (6)$$

Nếu $VAR(g_1(t)) = 0$ thì $f(t)$ (tức là $d(t)$) không có nghiệm trên $[0, 1/2]$.

Nếu $VAR(g_2(t)) = 0$ thì $f(t)$ không có nghiệm nào trên $[1/2, 1]$.

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

2. Thuật toán

a) Thuật toán pha rộng

Pha rộng loại trừ các cặp đối tượng 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 Algorithm1 thực hiện tính toán tại mỗi frame để trả về các cặp đối tượng có khả năng xảy ra va chạm.

b) Thuật toán pha hẹp

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.

Thuật toán 1: Algorithm1 //Lọc pha rộng

```

1 Dữ liệu vào: Mô hình 3D được biểu diễn dạng lưới tam
   giác.
2 Dữ liệu ra: Các cặp đối tượng có khả năng xảy ra va
   chạm.
3 begin
4   Tạo BV;
5   Thiết lập cấu trúc phân cấp BVH theo chiều từ trên
   xuống;
6   Thiết lập cấu trúc cây BVTT dựa vào phép duyệt cấu
   trúc phân cấp BVH;
7   Kiểm tra sự chồng lấp, loại bỏ các cặp đối tượng
   không có khả năng xảy ra va chạm tương ứng là
   với nút lá của cây BVTT; if không chồng lấp then
8     end
9   exit;
10  (có khả năng xảy ra va chạm) thì chuyển sang thực
   hiện Algorithm2;
11 end

```

Thuật toán 2: Algorithm2 //Lọc pha hẹp

```

1 Dữ liệu vào: Tọa độ 04 đỉnh của cặp VF hoặc cặp EE
   cần kiểm tra va chạm.
2 Dữ liệu ra: Kết luận va chạm (thông tin điểm tiếp xúc,
   vectơ pháp tuyến của mặt tiếp xúc,...) hoặc kết luận
   không va chạm.
3 begin
4   Kiểm tra và thực hiện lọc dựa vào các đặc điểm hình
   học (tính đồng phẳng của cặp VF hoặc cặp EE);
5   Tính VAR(f(x)) để phân tích sự tồn nghiệm trong [0,
   1];
6   Áp dụng quy tắc dấu Descartes và định lý Vincent;
7 end

```

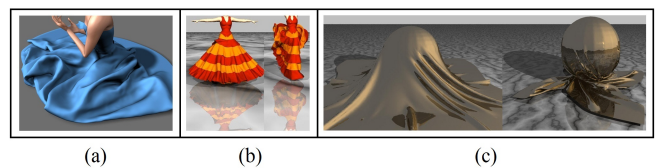
IV. THỬ NGHIỆM VÀ KẾT QUẢ

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 C++ trên Windows và các tập dữ liệu GAMMA (Đại học Maryland).

2. Dữ liệu

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



Hình 4. 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

Bộ dữ liệu Princess (Hình 4a): Mô phỏng một cô gái mặc chiếc váy dài thực hiện động tác múa nhẹ nhàng. Mô hình gồm 40000 tam giác.

Bộ dữ liệu Flamenco (Hình 4b): Mô phỏng một cô gái mặc chiếc váy xòe có 07 lớp vải thực hiện các động tác múa theo vũ điệu Flamenco. Bộ dữ liệu này có số lượng vải tự và chạm rất lớn. Mô hình gồm 49000 tam giác.

Bộ dữ liệu Cloth-ball (Hình 4c): Mô phỏng một mảnh vải (gồm 92230 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 và chạm.

Bảng III
THÔNG TIN VỀ BỘ DỮ LIỆU

Bộ dữ liệu	Số tam giác	Số đỉnh	Số cạnh	Số frames
Pincess	40000	20000	60000	464
Flamenco	49000	26000	75000	352
Cloth-ball	92230	46598	138825	94

Mỗi bộ dữ liệu có nhiều frames được mô tả thông tin chi tiết trong BẢNG 3. 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.

a) Kết quả

Kết quả trong BẢNG 4 cho thấy hiệu quả của phương pháp đề xuất được sử dụng trong bài báo này. So sánh cho thấy 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.

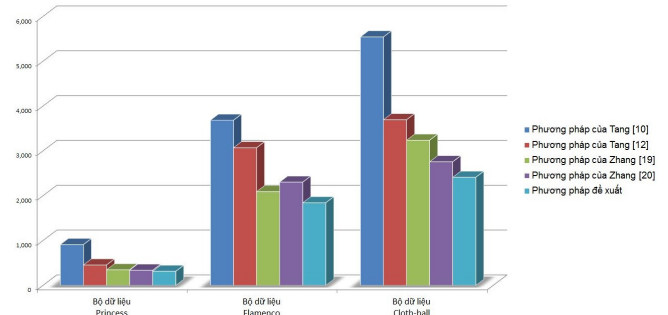
V. SO SÁNH, PHÂN TÍCH VÀ ĐÁNH GIÁ

1. So sánh và phân tích

Chúng tôi so sánh kết quả thử nghiệm phương pháp đề xuất với các phương pháp của Tang [10, 12] và Zhang [19, 20].

Phương pháp của chúng tôi 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 (False positive-FP) và tăng tốc phát hiện va chạm.

Biểu đồ trong Hình 5 so sánh cho thấy phương pháp đề xuất có kết quả chạy nhanh hơn. Lý do là chúng tôi 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



Hình 5. Một số loại khối bao được sử dụng trong các kỹ thuật phát hiện va chạm

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.

a) Hạn chế

Phương pháp trong bài báo 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 va chạm theo thời gian thực thì phương pháp đề xuất cần mở rộng cách tiếp cận với GPU.

VI. KẾT LUẬN, ĐỊNH HƯỚNG NGHIÊN CỨU PHÁT TRIỂN

Trong bài báo này, chúng tôi trình bày một số cải tiến để 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 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, chúng tôi 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.

VII. LỜI CẢM ƠN

Chúng tôi xin chân thành cảm ơn các tác giả của ba bộ dữ liệu mô hình vải khác nhau trong thư viện GAMMA, các tác giả Tang, Zhang và cộng sự.

TÀI LIỆU THAM KHẢO

- [1] Brochu T., Edwards E., Bridson R., "Efficient geometrically exact continuous collision detection", *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1-7, 2012.

Bảng IV
SO SÁNH THỜI GIAN PHÁT HIỆN VA CHẠM TRUNG BÌNH CỦA CÁC PHƯƠNG PHÁP

Bộ dữ liệu	Phương pháp của Tang [10]	Phương pháp của Tang [12]	Phương pháp của Zhang [19]	Phương pháp của Zhang [20]	Phương pháp đề xuất
Princess	910	455	347	337	313
Flamenco	3683	3069	2029	2302	1842
Cloth-ball	5542	3695	3238	2756	2411

- [2] Chitalu F. M., Dubach C., Komura T., "Binary Ostensibly-Implicit Trees for Fast Collision Detection", *Computer Graphics Forum*, vol. 39, no. 2, pp. 509-521, 2020.
- [3] Chitalu F. M., Dubach C., Komura T., "Bulk-synchronous Parallel Simultaneous BVH Traversal for Collision Detection on GPUs", in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 4-9, 2018.
- [4] Curtis S., Tamstorf R., Manocha D., "Fast collision detection for deformable models using representative-triangles", in *Proceedings of the Symposium on Interactive 3D Graphics and Games*, pp. 61-69, 2008.
- [5] Du P., Tang M., Tong R. F., "Fast continuous collision culling with deforming noncollinear filters", *Computer Animation and Virtual Worlds*, vol. 23, no. 3, pp. 375-383, 2012.
- [6] Du P., Zhao J., et al., "GPU accelerated real-time collision handling in virtual disassembly", *Journal of Computer Science and Technology*, vol. 30, no. 3, pp. 511-518, 2015.
- [7] Govindaraju N. K., Knott D., Jain N., et al., "Interactive collision detection between deformable models using chromatic decomposition", *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 991-999, 2005.
- [8] Meister D., Bittner J., "Parallel Locally-Ordered Clustering for Bounding Volume Hierarchy Construction", *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 3, pp. 1345-1353, 2018.
- [9] Tan T., Weller R., "OpenCollBench - Benchmarking of Collision Detection & Proximity Queries as a Web-Service", in *Proceedings of the 25th International Conference on 3D Web Technology*, pp. 1-9, 2020.
- [10] Tang M., Curtis S., Yoon S. E., "ICCD: interactive continuous collision detection between deformable models using connectivity-based culling", *IEEE Trans on Visualization and Computer Graphics*, vol. 15, no. 4, pp. 544-557, 2009.
- [11] Tang M., Manocha D., Tong R. F., "MCCD: Multi-core collision detection between deformable models using front-based decomposition", *Graphical Models*, vol. 72, no. 2, pp. 7-23, 2010.
- [12] Tang M., Manocha D., Tong R. F., "Fast continuous collision detection using deforming non-penetration filters", in *Proceedings of the SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 7-13, 2010.
- [13] Tang M., Tong R. F., Wang Z. D., et al., "Fast and exact continuous collision detection with bernstein sign classification", *ACM Transactions on Graphics*, vol. 33, no. 6, pp. 1-8, 2014.
- [14] Wang H. M., "Defending continuous collision detection against errors", *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 1-10, 2014.
- [15] Wang M., Cao J., "A review of collision detection for deformable objects", *Computer Animation and Virtual Worlds*, vol. 32, no. 5, pp.1-23, 2021.
- [16] Wang X., et al., "Efficient BVH-based Collision Detection Scheme with Ordering and Restructuring", *Journal of Computer Graphics Forum*, vol. 37, no. 2, pp. 227-237, 2018.
- [17] Wong T. H., Leach G., Zambetta F., "An adaptive octree grid for GPU-based collision detection of deformable objects", *Visual Computer*, vol. 30, no. 6, pp. 729-738, 2014.
- [18] Wu L., et al., "A Safe and Fast Repulsion Method for GPU-based Cloth Self Collisions", *ACM Transactions on Graphics*, vol. 40, no. 1, pp. 1-18, 2021.
- [19] Zhang X., Liu Y., "An algebraic non-penetration filter for continuous collision detection using Sturm theorem", in *Proceedings of the International Conference on Mechatronics and Automation*, pp. 761-766, 2015.
- [20] Zhang X., Liu Y., "A fast algebraic non-penetration filter for continuous collision detection", *Graphical Models*, vol. 80, no. 3, pp. 31-40, 2015.

SƠ LƯỢC VỀ TÁC GIẢ



Nghiêm Văn Hưng đang công tác tại Khoa Công nghệ thông tin, Trường Đại học Kỹ thuật – Hậu cần Công an nhân dân. Tốt nghiệp Cử nhân ngành CNTT tại Đại học Sư phạm Đà Nẵng năm 2008, tốt nghiệp Thạc sĩ ngành Hệ thống thông tin tại Học viện Kỹ thuật quân sự năm 2012. Nghiên cứu sinh ngành Hệ thống thông tin

tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Lĩnh vực nghiên cứu: Đa phương tiện, Mô phỏng, Thực tại ảo, Hệ thống thông tin.

Điện thoại: 0946.555.189

E-mail: nghiemvanhung1985@gmail.com



Dặng Văn Đức đang công tác tại Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Nhận học vị Tiến sĩ năm 1996, nhận chức danh Phó Giáo sư năm 2002. Lĩnh vực nghiên cứu: Đa phương tiện, GIS, Công nghệ phần mềm.

Điện thoại: 0912.223.163

E-mail: dvdudc@ioit.ac.vn



Nguyễn Văn Căn đang công tác tại Trường Đại học Kỹ thuật – Hậu cần Công an nhân dân. Nhận học vị Tiến sĩ chuyên ngành Cơ sở toán học cho tin học năm 2016. Lĩnh vực nghiên cứu: Công nghệ nhận dạng, Thị giác máy tính, Đa phương tiện, An ninh mạng, An toàn thông tin.

Điện thoại: 0986.919.333

E-mail: nguyenvancan@gmail.com



Hoàng Việt Long đang công tác tại Khoa Công nghệ thông tin, Trường Đại học Kỹ thuật – Hậu cần Công an nhân dân. Nhận học vị Tiến sĩ năm 2011, nhận chức danh Phó Giáo sư năm 2018. Lĩnh vực nghiên cứu: Học máy, An ninh mạng, An toàn thông tin, Đa phương tiện.

Điện thoại: 0918.700.525

E-mail: longhv08@gmail.com



Trịnh Hiền Anh đang công tác tại Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Tốt nghiệp Thạc sĩ ngành Công nghệ phần mềm tại Đại học Công nghệ-ĐHQG Hà Nội năm 2011. Nghiên cứu sinh tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam. Lĩnh vực

nghiên cứu quan: Đa phương tiện, Công nghệ mô phỏng, Thực tại ảo, Hệ thống thông tin.

Điện thoại: 0989.197.288

E-mail: hienanh@ioit.ac.vn