

VỀ PHƯƠNG PHÁP XÂY DỰNG PHÂN HỆ VÙNG BAO TỰ ĐỘNG CHO ĐỐI TƯỢNG 3D

Nguyễn Đức Hoàng

Học viện Công nghệ Bưu chính Viễn thông

Tóm tắt: Phân hệ vùng giới hạn (Bounding volume hierarchy - BVH) hay phân hệ vùng bao là một kiến trúc dạng cây cho một tập các đối tượng hình học. Việc lựa chọn vùng bao thường được xác định trên cơ sở phù hợp với các đối tượng và thường theo mô hình từ trên xuống (top-down), hoặc từ dưới lên (bottom-up) hoặc thêm vào (add in) cho một dạng hộp bao cụ thể. Đối với các đối tượng 3D, cần giải quyết các va chạm có thể xuất hiện giữa các đối tượng. Bài báo này đề cập đến phương pháp xây dựng một phân hệ vùng bao (BVH) tự động một đối tượng 3D. Phương pháp đề xuất dựa trên việc sử dụng nhiều dạng hộp bao khác nhau phù hợp với thực tế hoạt động của đối tượng. Kỹ thuật đã được thử nghiệm và tỏ ra hiệu quả đối với các mô hình đối tượng 3D được xây dựng theo phương pháp liên tục.

Từ khóa: Phân hệ vùng bao, nhiều dạng hộp bao, nhận dạng va chạm.

I. MỞ ĐẦU

Trong thế giới đồ họa, việc xây dựng các phân hệ vùng bao (Bounding volume hierarchy - BVH) là rất cần thiết, nhằm nhận dạng va chạm giữa các đối tượng khác nhau để có thể biểu diễn các hiệu ứng của chúng [1,2,3]. Một phân hệ vùng bao (BVH) là phân hệ phổ biến dùng để đơn giản hóa việc biểu diễn các đối tượng bằng cách sử dụng các thành phần của các hình học bao quanh đối tượng. Phân hệ cho phép đóng gói các đối tượng phức tạp bằng các vùng bao đơn giản. Do đó, các phân hệ này rất có ích trong việc phát hiện va chạm của các đối tượng [2,3].

Phân hệ vùng bao đóng vai trò quan trọng trong việc biểu diễn các vật thể, cho phép giải quyết nhiều vấn đề trong lý thuyết và ứng dụng của nhận dạng va chạm, dò tia [3, 4, 5, 6]. Các kỹ thuật này cho phép giải các bài toán trong nhiều lĩnh vực như robotic, đồ họa máy tính, đồ họa động, trò chơi điện tử, thực tại ảo, mô phỏng và biểu diễn có khả năng tương tác.

Các phân hệ vùng bao BVH đã được đề xuất và áp dụng tới nay là một cách tiếp cận thành công nhất trong các hệ thống biểu diễn đồ họa hiện hành [7].

Thời gian tính toán cho các hệ thống này thể hiện độ ưu việt của các phân hệ BVH [5, 8].

Theo [2], các phân hệ vùng bao phổ biến nhất gồm: Phân hệ vùng bao hình khối cầu (Sphere), phân hệ vùng bao có định hướng OBB (Oriented Bounding Box) hay hình hộp chữ nhật, phân hệ khối lập phương AABB (Axis-Aligned Bounding Box) và phân hệ vùng bao k-DOP (Discrete Oriented Polytopes) [2].

Phân hệ vùng bao khối cầu (Sphere) [9] và khối lập phương (AABB) [7] tạo ra phép thử chồng lấn đơn giản nhất. Trong khi đó, phân hệ vùng bao khối chữ nhật (OBB) [7] và khối đa diện rời rạc có hướng (k-DOP) [5, 9] cho biểu diễn khít nhất.

Trong [10], Beckmann và các tác giả đã đưa ra giải thuật cho cây AABB. Palmer và các tác giả trong [11], Hubbard và các tác giả trong [9] đã đưa ra giải thuật cho cây khối cầu để giải quyết vấn đề đơn giản hóa. Trong khi đó, Gottschalk và các tác giả trong [4, 5] đã đưa ra giải thuật cho khối OBB. Klosowski và các tác giả trong [12, 13] đã đưa ra giải thuật cho khối đa diện k-DOP để giải quyết vấn đề về độ khít của hộp bao. Van den Bergen và các tác giả trong [14] đã đưa ra một phương thức đơn giản để phân tách các hộp chữ nhật OBB được biết đến với tên SAT lite. Giải thuật này chỉ sử dụng 6 trong số 15 hệ trục tọa độ so giải thuật gốc, do đó giảm được thời gian tính toán.

Tuy nhiên, việc xây dựng một phân hệ vùng bao vẫn có vấn đề nan giải về khả năng giảm thiểu tính toán trong khi vẫn phải bảo đảm được độ chính xác khi biểu diễn [2]. Các phân hệ vùng bao thường là một cây trong đó các đối tượng hoàn chỉnh được thể hiện chặt chẽ và phù hợp với mọi cấp độ của phân hệ. Ngoài ra, mỗi vùng bao có yêu cầu cụ thể về thời gian tính toán và độ chính xác. Chẳng hạn, phân hệ vùng bao khối cầu là bất biến đối với phép quay và dịch chuyển, do vậy cấu trúc của nó và việc kiểm tra sai lệch đơn giản hơn nhiều so với phân hệ khác, ví dụ các phân hệ OBB. Tuy nhiên, độ khít của vùng bao khối cầu lại kém hơn so với các phân hệ vùng bao OBB [1, 2, 6].

Trong bài báo này, tác giả sẽ tập trung vào hai loại phân hệ vùng bao phổ biến và xem xét vấn đề xây dựng một phân hệ vùng bao tự động cho các đối tượng 3D nhằm tối ưu cả về mặt độ khít của hệ bao và độ đơn giản của phép thử chồng lấn.

Tác giả liên hệ: Nguyễn Đức Hoàng
email: hoangnc@ptit.edu.vn

Đến tòa soạn: 12/2/2019, chỉnh sửa: 12/4/2019, chấp nhận đăng: 13/5/2019.

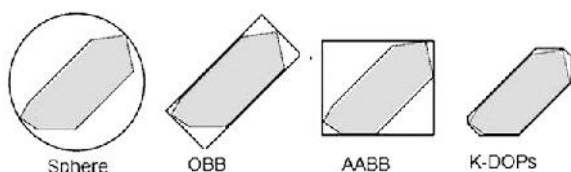
Vấn đề xây dựng phân hệ vùng bao (BVH) cho một đối tượng 3D dựa trên việc sử dụng nhiều dạng hộp bao được đề cập tới với hai mục tiêu: giảm thời gian tính toán nhưng vẫn đạt được độ chính xác.

Cấu trúc phần còn lại của bài báo như sau. Phần II của bài trình bày về nguyên tắc phân hệ vùng bao (BVH). Phần III trình bày về kỹ thuật xây dựng hệ bao tự động với nhiều dạng hộp bao. Phần IV đưa ra một số kết quả thử nghiệm. Phần V là kết luận của bài.

II. PHÂN HỆ VÙNG BAO

A. Các phân hệ vùng bao cơ bản

Ví dụ về các phân hệ vùng bao cơ bản được trình bày trên hình 1.



Hình 1. Các phân hệ vùng bao (BVH) cơ bản

Phân hệ vùng bao Sphere dựa trên việc đặt một khối đa diện lồi trong một hình cầu. Hình cầu bên ngoài cho phép liên kết khối đa diện, được sử dụng để nhanh chóng xác định tính không giao nhau (va chạm) giữa các đối tượng (các khối đa diện). Hình cầu bên trong được sử dụng để xác định giao điểm giữa các đa diện. Ưu điểm của vùng bao hình cầu là hiệu quả trong việc tính toán các giao điểm và các khoảng cách. Mặc dù các mặt cầu là bất biến đối với phép quay hay dịch chuyển, song chúng không thật sự phù hợp cho các khối đa diện kéo theo chiều dài [2].

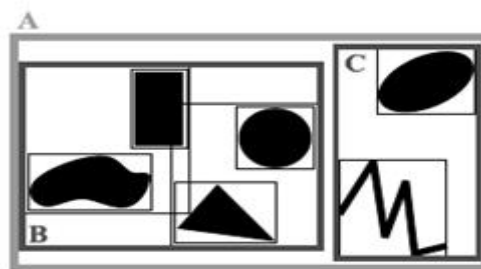
Phân hệ vùng bao OBB sử dụng định hướng đa diện và một hình chữ nhật được tính toán để bao đối tượng. Ưu điểm của phân hệ này là sự bất biến đối với phép quay và dịch chuyển. Ta có thể di chuyển hoặc xoay đối tượng và cả vùng bao nó cũng nhau. Tuy nhiên, tính toán về độ va chạm khó khăn hơn so với các phân hệ khác. Mặc dù vậy, một số nghiên cứu đã chỉ ra, OBB tiệm cận nhanh hơn so với các phân hệ khác [4, 5, 8].

AABB là một hình chữ nhật được sắp xếp theo trục bao quanh khối đa diện. Ưu điểm của AABB là: 1) dễ dàng tìm thấy một hình chữ nhật phù hợp, 2) AABB là bất biến đối với phép dịch chuyển, 3) AABB cho phép thử đơn giản. Tuy nhiên, AABB không bất biến đối với phép quay, do đó, những thay đổi theo hướng của các đối tượng đòi hỏi phải thay đổi trong phân hệ vùng bao chữ nhật. Nhiều nghiên cứu đã tìm cách lai ghép giữa OBB và AABB nhằm hạn chế các nhược điểm của AABB.

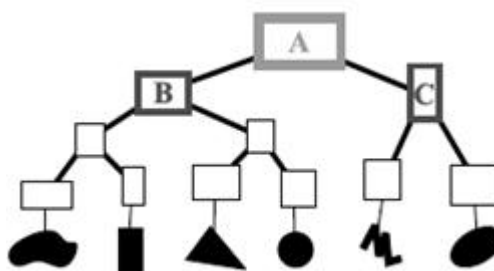
Phân hệ k-DOP là một dạng AABB tổng quát. k-DOP là một đa giác lồi chứa đối tượng, được xây dựng bằng cách lấy một số k của các mặt phẳng định hướng thích hợp ở vô cực và đưa nó lại gần đối tượng cho đến khi chúng va chạm nhau. Các DOP phổ biến nhất được tính bằng 6 mặt phẳng thẳng hàng trục (hộp giới hạn hướng trục), 10 mặt phẳng thẳng hàng trục (hộp giới hạn vát trên các cạnh thẳng đứng), 18 mặt phẳng thẳng hàng trục (vát trên tất cả các cạnh) hoặc 26 trục-

các mặt phẳng thẳng hàng (vát trên tất cả các cạnh và góc) [2].

Hình 2 là ví dụ về một vùng bao sử dụng hình chữ nhật AABB và Hình 3 biểu thị phân hệ vùng bao tương ứng sử dụng hình chữ nhật.



Hình 2. Vùng bao sử dụng hình chữ nhật



Hình 3. Phân hệ vùng bao sử dụng hình chữ nhật

Theo [4], thời gian tính toán cho các phân hệ vùng bao được theo công thức sau:

$$T = N_v \times C_v + N_p \times C_p$$

Trong đó:

- T là tổng thời gian tính toán.
- N_v là số các phép thử của một cặp hệ bao chồng lấn.
- C_v là thời gian của phép thử cho một cặp các hệ bao.
- N_p là số các phép thử của một cặp hình cơ bản chồng lấn.
- C_p là thời gian của phép thử cho một cặp các hình cơ bản.

Điều này chứng tỏ một phân hệ vùng bao hoạt động dựa trên hai yếu tố: độ khít của hệ bao so với đối tượng (N_v, N_p) và độ đơn giản của phép thử chồng lấn trên một cặp hệ bao (C_v).

B. Hộp bao

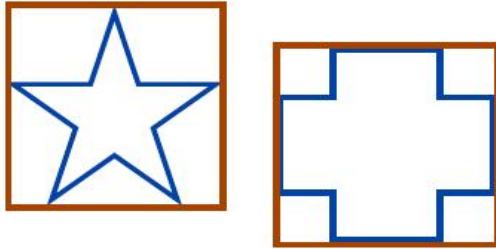
Đối với các đối tượng 3D, việc giải quyết các bài toán như nhận dạng va chạm, dò tia, ... cần phải xem xét đến bề mặt cũng như phần thể tích bên trong của đối tượng. Việc này trở nên phức tạp và rất tốn tài nguyên nếu đối tượng xem xét có hình dạng phức tạp.

Để phân tích các tác động lên các đối tượng này, hộp bao được sử dụng. Thay vì việc cần phải xem xét toàn bộ đối tượng, hộp bao cho phép việc chỉ cần tính toán dựa trên các hình hình học đơn giản. Đối với các bài toán không yêu cầu độ chính xác quá cao, việc xem xét giới hạn ở phân tích bề mặt (3D) hoặc đường bao (2D) của hộp bao.

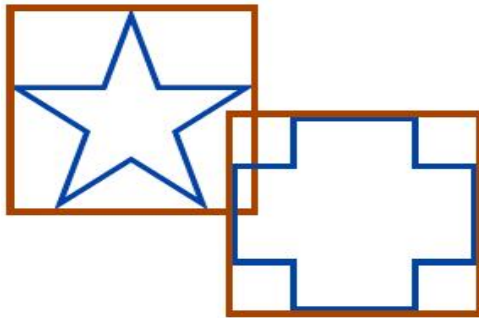
Tuy nhiên, để bảo đảm độ đơn giản tính toán, các bài toán sử dụng hộp bao thường đưa ra các giả thiết sau đây:

- Các phép tính chỉ dừng lại ở mức gần đúng.
- Tính chính xác của các phép tính sẽ dựa trên độ khít của đường bao.

Phép tính chỉ dừng ở mức gần đúng do rất khó xác định được độ va chạm giữa các đối tượng. Hình 4 biểu thị các hộp bao không có va chạm (không có chồng lấn hộp bao). Hình 5 biểu thị các hộp bao có va chạm (nghĩa là có chồng lấn hộp bao).



Hình 4. Không có va chạm (không có chồng lấn hộp bao)

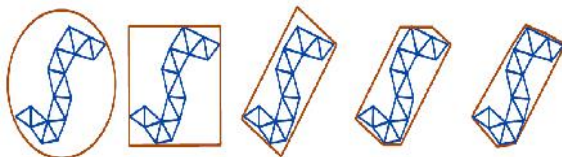


Hình 5. Có va chạm (có chồng lấn hộp bao)

Các dạng hộp bao thường được sử dụng để xây dựng phân hệ vùng bao cho đối tượng, bao gồm:

- Hộp bao khối cầu: Sphere
- Hộp bao khối lập phương: AABB
- Hộp bao khối chữ nhật có hướng: OBB
- Hộp bao khối đa diện rời rạc có hướng: k-DOP
- Hộp bao khối lồi: convex hull

Hình 6 biểu diễn các dạng hộp bao cơ bản để xây dựng phân hệ vùng bao cho đối tượng.



Hình 6. Các dạng hộp bao

Hộp bao khối cầu (Sphere):

Hộp bao khối cầu được biểu diễn bởi tâm (c) và bán kính khối cầu (r). Hai khối cầu không chồng lấn nhau khi:

$$(c_1 - c_2) \cdot (c_1 - c_2) > (r_1 - r_2)^2$$

Hộp bao khối lập phương (AABB):

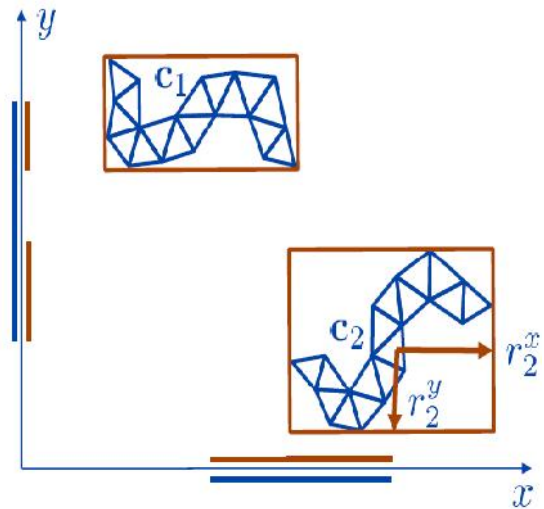
Hộp bao khối lập phương AABB được biểu diễn bởi tâm hộp (c) và tham số chiều dài các cạnh (rx, ry, rz)

Hai khối hộp lập phương không chồng lấn lên nhau khi (xét trong miền không gian 2D):

$$|(c_1 - c_2) \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}| > r_1^x + r_2^x$$

$$|(c_1 - c_2) \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}| > r_1^y + r_2^y$$

Hình 7 mô tả va chạm (có chồng lấn) giữa hai khối hộp lập phương AABB.

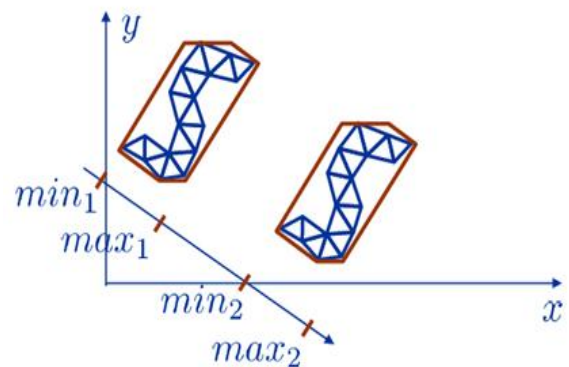


Hình 7. Va chạm giữa hai hộp bao khối lập phương AABB

Hộp bao khối đa diện rời rạc có hướng (k-DOP):

Hộp bao khối đa diện rời rạc có hướng k-DOP được xác định bởi hai tham số: k/2 trung bình; k/2 khoảng cách lớn nhất - nhỏ nhất.

Như vậy nếu trong miền không gian 2D, có thể coi hộp bao khối lập phương AABB là 4-DOP. Trong miền không gian 3D có thể coi hộp bao khối lập phương AABB là 6-DOP.



Hình 8. Biểu diễn hai khối OBB

Hai cặp hộp đa diện sẽ không chồng lấn lên nhau khi (xét trong miền không gian 2D):

$$E_{direction} : \max_1 < \min_2 \vee \min_1 < \max_2$$

Hộp bao khối chữ nhật có hướng (OBB):

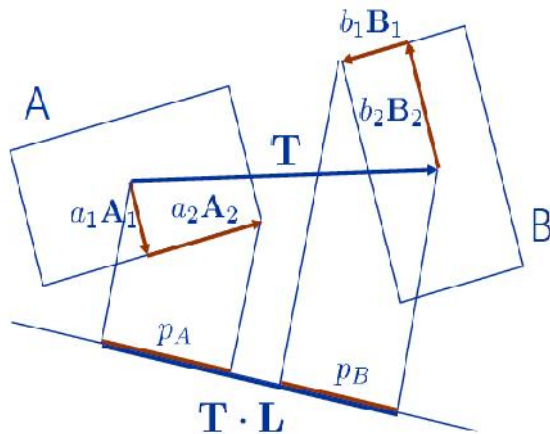
Hộp bao khối chữ nhật có hướng OBB giống như khối hộp lập phương AABB nhưng có khả năng xoay.

Bài toán xác định không chồng lấn đối với khối hộp OBB được cụ thể hóa như sau:

• **Trong miền không gian 2D:**

OBB được biểu diễn bởi các tham số (xem Hình 9):

- A1, A2, B1, B2: biểu diễn pháp tuyến vuông góc của hai đối tượng A và B.
- a1, a2, b1, b2 biểu thị số đo các cạnh của hai hộp bao.
- L là pháp tuyến chỉ hướng.
- T là khoảng cách giữa các hộp bao A và B
- $p_A = a_1A_1L + a_2A_2L$
- $p_B = b_1B_1L + b_2B_2L$



Hình 9. Xác định và chạm giữa hai khối OBB

A và B không chồng lấn nhau khi:

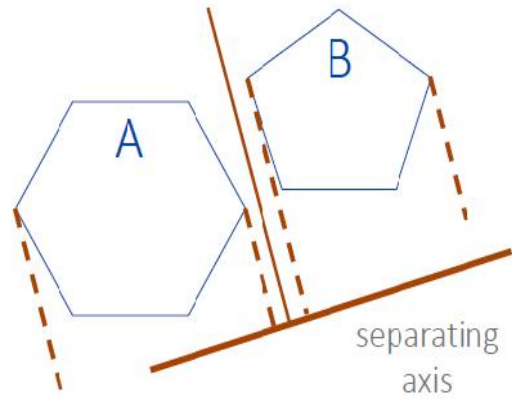
$$E_{L:T.L} > p_A + p_B$$

Để xét hai đối tượng lồi có chồng lấn lên nhau hay không, một trục tọa độ phân tách v sẽ được xác định giữa hai đối tượng. Đối với các đối tượng này một số các trục cần xem xét như sau:

- Trục song song với mặt trung bình của A
- Trục song song với mặt trung bình của B
- Trục song song với mặt cắt tại các góc của các hộp bao A và B

• **Trong miền không gian 3D:**

Để xác định chồng lấn, các trục cần xem xét gồm 15 trục để xác định được trục tọa độ phân tách.

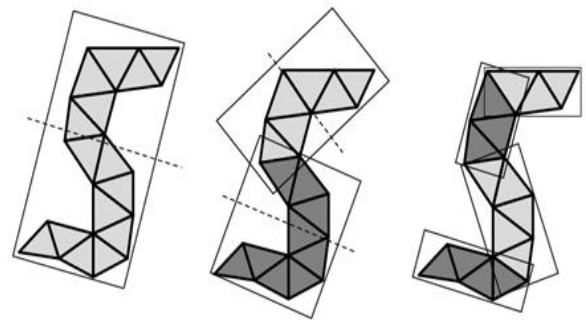


Hình 10. Xác định và chạm giữa hai khối đa diện

C. **Xác định phân hệ vùng bao (BVH)**

Phân hệ vùng bao (BVH) là một cấu trúc dữ liệu dạng cây được xây dựng trên cơ sở phân tích các đối tượng được xem xét dựa trên cơ sở các hộp bao hình học. Tại các lá của phân hệ là các dạng hình học cơ bản.

Hình 11 mô tả một phân hệ vùng bao được xây dựng bởi các hộp bao.



Hình 11. Xây dựng phân hệ vùng bao từ các hộp bao

Phân hệ vùng bao có các đặc điểm sau:

- Các nút trong một nhánh phải gần nhau hơn so với các nút khác. Càng xuống thấp thì các nút càng phải gần nhau hơn.
- Mỗi nút trong BVH cần có thể tích nhỏ nhất
- Tổng của các khối bao cần phải tối giản
- Các nút càng gần gốc thì càng quan trọng. Việc loại bỏ một nút gần gốc sẽ ảnh hưởng lớn hơn nhiều lần so với các nút ở xa.
- Thể tích trùng nhau của các nút đồng cấp phải tối giản.
- Độ khít: Độ khít có thể tính toán qua thể tích, cụ thể theo công thức sau [2]:

$$\tau = \frac{\text{volume}(B)}{\sum_{C \in C(B)} \text{volume}(C)}$$

Trong đó:

- C(B) là tập các nhánh con tại nút B

- volume(B) là thể tích của hệ bao tại B
- τ là độ khít

Giá trị của hệ bao được tính dựa trên các tham số sau đây [2]:

$$cost(H) = \sum_{n \in H} (C(n) \cdot volume(n))$$

Trong đó:

- H là hệ bao
- C(n) là tập các nhánh con tại nút n
- cost là giá trị hệ bao

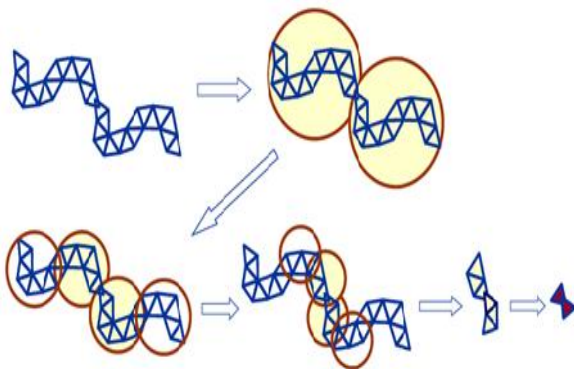
Để xác định một phân hệ vùng bao, ta xem xét các phương thức thiết lập cây và các phương thức kiểm tra cây như sau.

1) Phương thức thiết lập cây:

- Từ trên xuống: Chia đầu vào thành hai (hoặc nhiều) nhánh, bao chúng lại, sau đó tiếp tục chia nhỏ các nhánh đến khi mỗi nhánh chỉ chứa một hình cơ bản. Phương pháp này cho phép tạo ra cây đơn giản nhưng không được ứng dụng nhiều trong thực tế.
- Từ dưới lên: Bắt đầu với các hình cơ bản tại các nhánh, sau đó cộng gộp dần để xây dựng thành đối tượng ban đầu. Phương pháp này khó thực hiện nhưng nhìn chung có thể tập hợp thành cây tốt hơn.
- Thêm vào: Hai phương pháp trên sử dụng tất cả các hình cơ bản trước khi tổ hợp thành cây. Phương pháp thêm vào cho phép không cần sử dụng tất cả các hình cơ bản. Cây ban đầu được xây dựng là một cây rỗng và được xây dựng dần bằng việc xác định cây nhỏ nhất.

2) Phương thức kiểm tra đối với cây:

- Nếu hộp bao trên một tầng nào đó của hệ bao bị chồng lấn, các nhánh con của nó cần được kiểm tra
- Tại các lá, việc kiểm tra thực hiện đối với các hình hình học cơ bản
- Loại bỏ các phần đối tượng không chịu tác động



Hình 12. Ảnh hưởng của các va chạm tới các phần tử của hệ bao

III. PHƯƠNG PHÁP XÂY DỰNG PHÂN HỆ VÙNG BAO TỰ ĐỘNG VỚI NHIỀU DẠNG HỘP BAO

A. Giả thiết cho bài toán

Phương pháp xây dựng phân hệ vùng bao được thực hiện trên cơ sở các giả thiết sau đây:

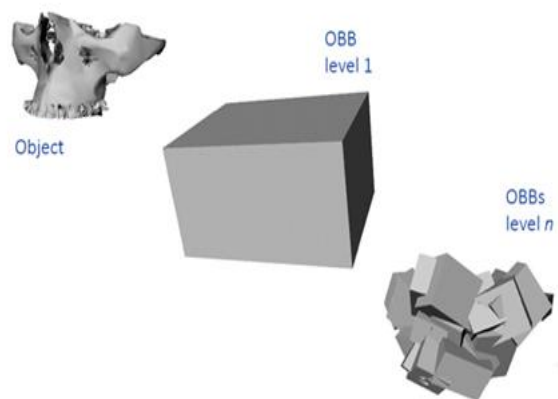
- Chỉ thực hiện trên hai vật thể rắn. Tính ưu việt của kỹ thuật được thể hiện qua việc cho hai vật thể rắn giống hệt nhau va chạm với nhau. Thời gian tính toán va chạm là tiêu chí để xem xét.
- Việc biểu diễn hệ bao đối tượng với nhiều dạng hộp bao sẽ giới hạn ở hai dạng hộp bao thuộc về mỗi phương hướng tối ưu.
- Một phân hệ vùng bao với hai dạng hộp bao được lựa chọn, trong đó mỗi nút hộp bao thuộc hướng khác sẽ được tăng cường bởi một hộp bao có hướng đơn giản.
- Phép thử với hộp bao hướng đơn giản sẽ được thực hiện trước để loại trừ các đối tượng ở xa

B. Xây dựng phân hệ bao tự động

Việc xây dựng tự động hệ bao có thể được coi là tự động xây dựng cấu trúc dữ liệu hình cây mô tả hệ bao [15].

Phương thức chung để xây dựng một hệ bao có thể được miêu tả như sau: hệ bao được xây dựng trên cơ sở một cây dữ liệu các hộp bao. Trong đó các hộp bao là các hình đơn giản được sắp xếp khít quanh nhau, bao phủ đối tượng cần xem xét. Các hộp này được đề cập đến ở phần II.

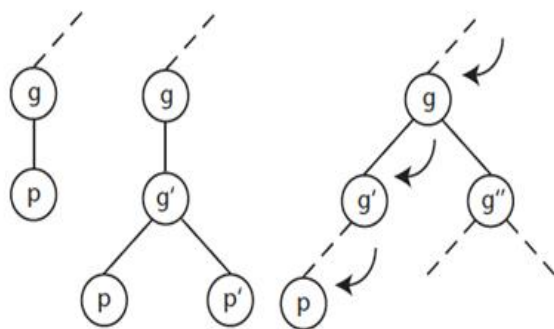
Hình 13 mô tả một ví dụ về một phân hệ vùng bao sử dụng các hộp bao OBB.



Hình 13. Ví dụ về một phân hệ vùng bao sử dụng các hộp bao OBB

C. Các giải thuật hỗ trợ xây dựng phân hệ bao tự động

Một số giải thuật có thể sử dụng để xây dựng hệ bao tự động như sau [16, 17, 18].



Hình 14. Giải thuật thêm dần

1) Giải thuật thêm dần

Giải thuật được đưa ra bởi Goldsmith [16]. Giải thuật được thiết lập dựa trên việc tính toán giá trị nhỏ nhất của cây khi thêm các hình cơ bản vào trong hệ.

Khi một hình cơ bản p được thêm vào trong một phân hệ được phân chia (xem Hình 14), giải thuật sẽ sử dụng 3 luật như sau:

- p có thể là nhánh con của một nhóm g
- p có thể kết hợp với một hình cơ bản p' nhóm g', g' sẽ là một nhánh con của g
- p có thể được thêm vào một nhóm g' thuộc nhóm đệ quy của g

Phương pháp nêu trên có thể được sử dụng để tạo một hệ bao xấp xỉ tuy nhiên nó có một số hạn chế. Hệ được tạo ra dựa trên yêu cầu thêm vào của các nút. Và yêu cầu này là không mong muốn do phải dựa trên cảm quan của người xây dựng hệ bao.

Trong một số trường hợp giá trị của cây sẽ không tối ưu và mỗi nhóm mới chỉ chứa hai hình cơ bản. Điều này được cải thiện hơn trong thuật toán được đưa ra bởi Haber [17] sử dụng hai cách tiếp cận như sau:

- Thêm lại thành công: Loại bỏ những nút không tốt và thêm lại chúng vào hệ bao
- Giới hạn các nhóm xấu: Tìm các nhóm không tốt và cố gắng chia chúng ra.

2) Giải thuật chia nhỏ

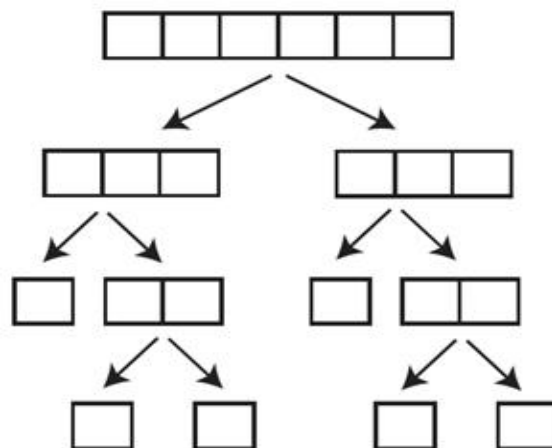
Thuật toán này được xây dựng bởi Muller [18]. Thuật toán chia nhỏ một tập hợp các hình cơ bản một cách đệ quy thành hai tập con không trùng phần tử. Quá trình này được dừng lại khi đạt đến một mức ngưỡng.

Thuật toán thực hiện theo các bước như sau:

- Cây phân hệ vùng bao được xây dựng bởi việc sắp xếp các hình cơ bản theo các trục tọa độ chính và lấy mốc là tâm của các hình cơ bản.
- Sau đó chức năng lựa chọn giá trị nhỏ nhất của cây hoạt động trên việc xem xét tất cả các điểm phân chia có thể.
- Thuật toán sẽ tiếp tục chia đến khi các cây chứa toàn các hình cơ bản tại các lá.

Giải thuật này cũng được Gottschalk [4,5] sử dụng cho phân hệ vùng bao sử dụng các hộp bao OBB.

Trong đó, việc chia nhỏ sẽ tiến hành dọc theo trục dài nhất, sử dụng các điểm trung tâm.



Hình 15. Giải thuật chia nhỏ

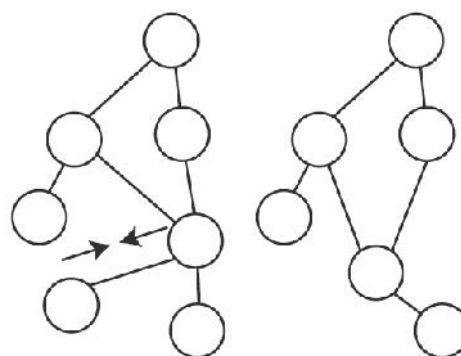
Hình 15 mô tả giải thuật chia nhỏ, trong đó cây được xây dựng bằng cách phân chia dọc theo một trong ba trục tại các điểm có giá trị nhỏ nhất.

Điểm hạn chế duy nhất của giải thuật này là chỉ xây dựng được các phân hệ vùng bao nhị phân. Tuy nhiên có thể khắc phục bằng cách chia nhiều lần tại cùng mỗi cấp. Độ cân bằng của cây phụ thuộc và chức năng giá trị được sử dụng.

3) Giải thuật kết hợp

Giải thuật được xây dựng bởi Erleben [19, 20] và có thể thấy được áp dụng trong OpenTissue [21]. Các bước của giải thuật như sau:

- Giải thuật này bắt đầu với việc xây dựng cấu trúc đồ thị dữ liệu, trong đó mỗi nút thuộc đồ thị liên quan đến các hình cơ bản và các đỉnh có quan hệ lân cận.
- Một đỉnh trong đồ thị nghĩa là hai nút trong hệ bao có thể kết hợp tốt với nhau.
- Các đỉnh được xác định bằng một chức năng phỏng đoán trong đó phóng đại hộp bao cơ bản và ghi nhận va chạm.
- Một va chạm có nghĩa là một đỉnh giữa hai đồ thị nút vừa va chạm cần được thêm vào đồ thị.
- Quá trình trên được lặp đi lặp lại cho đến khi một nút duy nhất tồn tại.



Hình 16. Một đỉnh sụp đổ thành một nút

Hình 16 mô tả giải thuật kết hợp, trong đó một đỉnh sụp đổ thành một nút.

Sau khi một đỉnh sụp đổ trong đồ thị, các nút thuộc phân hệ vùng bao được kết hợp thành một nhóm mới khi một trong hai điều kiện sau thỏa mãn:

- Đồ thị nút bao phủ lượng lớn hơn một nhánh cố định.
- Có ít đỉnh hơn trong một đồ thị so với một nhánh cố định.

D. Phương pháp lựa chọn hộp bao phù hợp

Như đã trình bày ở trên, việc xây dựng phân hệ vùng bao đối tượng có thể thông qua các phương pháp chính là: sử dụng hệ bao cầu (Sphere); hệ bao hộp chữ nhật (AABB); hệ bao hộp chữ nhật có hướng (OBB); và hệ bao đa diện có hướng rời rạc (k-DOP).

Để tận dụng lợi thế của hai dạng hộp bao: tính đơn giản của các hộp bao AABB và Sphere; tính chính xác của các hộp bao OBB và k-DOP, ta có thể xây dựng một cây phân hệ vùng bao được xây dựng bằng nhiều dạng hộp bao trên mỗi nút. Trong đó, tại mỗi nút sẽ có một hộp bao dạng đơn giản và một hộp bao dạng chính xác.

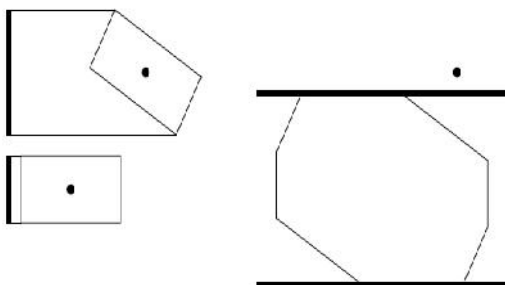
Trong bài báo này, ta lựa chọn sử dụng hai dạng hộp bao là AABB và OBB để xây dựng cây phân hệ vùng bao cho đối tượng 3D.

Cấu trúc cây về cơ bản được xây dựng dựa trên cấu trúc cây OBB đã được đưa ra bởi Gottschalk [4, 5].

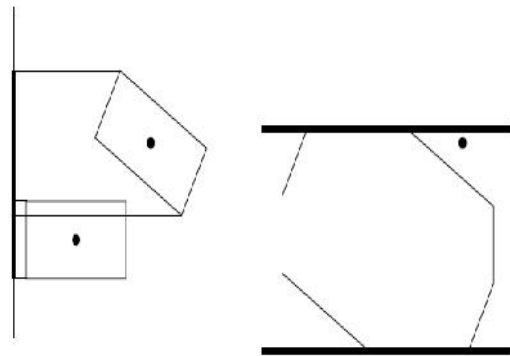
Với mỗi nút trên cây OBB được xây dựng, cấu trúc hai hộp bao sẽ được xây dựng bao gồm thêm một hộp bao AABB bao các thành tố của mặt phẳng tại nút đó.

Ta có thể sử dụng hai phương thức để xây dựng hộp bao AABB trong trường hợp này.

- Phương thức thứ nhất sẽ tìm ra hộp bao AABB nhỏ nhất cho đối tượng. Phương thức thứ hai sẽ đặt tâm của hộp AABB trùng với tâm của hộp OBB.
- Phương thức thứ hai sẽ cho giải thuật đơn giản hơn và việc tính toán sẽ nhanh hơn. Trong khi đó phương án thứ nhất sẽ cho hộp bao AABB khít hơn đối với đối tượng. Theo một số thực nghiệm, việc chọn khối hộp AABB khít sẽ cho kết quả của các phép thử tốt hơn.



Hình 17. Sử dụng hai hộp bao không chồng lấn



Hình 18. Sử dụng hai hộp bao có chồng lấn

Việc kiểm tra khi phân tách nút đối với cây phân hệ vùng bao sử dụng hai dạng hộp bao sẽ được thực hiện như sau:

- Hệ hộp bao AABB sẽ được kiểm tra trước, nếu chúng cần phải chia nhỏ thì phân hệ vùng bao chung sẽ chia nhỏ.
- Nếu hệ hộp bao AABB bị chồng lấn, khi đó hộp bao OBB sẽ được xem xét tiếp.

Những ưu điểm của phương pháp xây dựng hộp bao đã nêu trên bao gồm:

- Tăng cường độ khít của hộp bao so với các phương pháp AABB đơn lẻ. Điều này đạt được do độ khít của hộp OBB tốt hơn so với hộp AABB
- Giảm độ phức tạp của phép thử so với phương pháp sử dụng hộp OBB. Do chỉ phải thực hiện phép thử với hệ hộp AABB trước, nếu xảy ra chồng lấn thì mới cần xét tiếp đến hệ hộp OBB nên số lượng tính toán của phương pháp kép sẽ giảm thiểu.

E. Nhận xét

Phương pháp xây dựng phân hệ vùng bao (BVH) tự động cho các đối tượng 3D nêu trên có những ưu điểm và nhược điểm chính như sau:

- Ưu điểm chính của phương pháp này là việc không làm giảm độ chính xác của các phép thử do sử dụng hệ bao đảm bảo chính xác (OBB) làm cơ sở. Một ưu điểm khác là có khả năng tăng tốc tính toán do sử dụng hệ bao đảm bảo tính đơn giản (AABB) để tính toán trước, khi va chạm xảy ra tại nhánh nào thì mới khoanh vùng để tính chính xác.
- Hạn chế của phương pháp này là thời gian xây dựng phân hệ vùng bao sẽ tăng lên nhiều so với phương pháp sử dụng phân hệ vùng bao sử dụng một dạng hộp bao. Ngoài ra do có hai dạng hộp bao trên một vật thể nên kích thước của đối tượng được xem xét cũng sẽ tăng lên.

F. Thuật toán xây dựng phân hệ bao

Các bước xây dựng thuật toán có thể được mô tả tóm tắt như sau:

- Bước 1: Xây dựng cây dữ liệu phân hệ vùng bao theo phương pháp xây dựng phân hệ vùng bao tự động sử dụng cho dạng hộp bao là AABB theo giải thuật của Gottschalk như đã nêu trên.
- Bước 2: Tại mỗi nút trên cây phân hệ vùng bao đã xây dựng, thực hiện tái tạo một cây mới có cấu trúc cây giống cây cũ. Dạng hộp bao được sử dụng sẽ được thay thế bằng OBB.
- Bước 3: Xây dựng giải thuật và tính toán dựa trên cơ sở việc phát hiện các va chạm xảy ra với phân hệ vùng bao.
- Nếu không xảy ra va chạm: Phân hệ vùng bao cho đối tượng sẽ là phân hệ vùng bao sử dụng dạng hộp bao là AABB.
- Nếu xảy ra va chạm tại một nút nào đó thuộc hệ vùng bao: Phân hệ vùng bao cho đối tượng sẽ là phân hệ vùng bao sử dụng dạng hộp bao là OBB.

IV. KẾT QUẢ THỬ NGHIỆM

Trong phần này, bài báo tóm tắt một số kết quả tính toán với các mẫu thử. Việc tính toán thời gian xử lý được áp dụng cho các dạng bề mặt khác nhau, với các cấu hình khác nhau.

Mẫu thử được dùng trong thử nghiệm là hình khối tượng Phật Di lặc. Mẫu thử hệ bao được thể hiện dưới dạng hệ lưới bao gồm 15.536 tam giác. Va chạm xảy ra với hai đối tượng giống nhau sẽ có 229,824 cách cấu hình vị trí và hướng mẫu thử.

Bảng 1 biểu thị các mẫu thử sử dụng 6 dạng khoảng cách khác nhau: 0%, 1%, 2%, 3%, 4% và 5% cho kích thước mẫu thử đưa vào. Mỗi khoảng cách được xác định bởi bán kính của hộp bao. Cách cấu hình vị trí và hướng mẫu thử được đưa ra bởi Trenkel [22].

Bảng 1. So sánh thời gian tính toán với các mẫu thử

		RAPID	DUAL
Mẫu thử	0%	27.2540	20.6053
	1%	14.0696	10.1924
	2%	8.6457	5.8939
	3%	6.2860	4.0741
	4%	4.9193	3.0381
	5%	4.0032	2.3816

Giải thuật RAPID là giải thuật cho phép nhận dạng va chạm trên cơ sở sử dụng các hộp bao OBB. Giải thuật này xây dựng trên cơ sở sử dụng giải thuật OBB từ trang web:

<http://www.cs.unc.edu/~geom/OBB/OBBT.html>

Trên cơ sở thay đổi mã nguồn mở của giải thuật này chúng tôi đã xây dựng giải thuật cho việc nhận dạng va chạm sử dụng hai dạng hộp bao.

Kết quả về thời gian tính toán trong Bảng 1 cho thấy: việc sử dụng hai hộp bao với giải thuật DUAL

cho thời gian nhỏ hơn so với giải thuật chỉ sử dụng một hộp bao RAPID.

V. KẾT LUẬN

Việc xây dựng các phân hệ vùng bao (Bounding volume hierarchy - BVH) là vấn đề cần giải quyết nhằm đạt được hiệu quả về tính đơn giản, tính chính xác và thời gian tính toán ngắn.

Phương pháp xây dựng hệ bao BVH tự động cho một đối tượng 3D được đề xuất trong bài thể hiện nhiều ưu điểm. Phương pháp này được xây dựng trên cơ sở sử dụng nhiều dạng hộp bao khác nhau phù hợp với thực tế hoạt động của đối tượng. Ưu điểm của phương pháp đã đề xuất là cho kết quả thời gian tính toán ngắn hơn so với phương pháp sử dụng một hộp bao như đã mô tả trong [4,5]. Các kết quả thử nghiệm với hai dạng hộp bao và tỏ ra hiệu quả đối với các mô hình đối tượng 3D được xây dựng theo phương pháp sử dụng một hộp bao.

TÀI LIỆU THAM KHẢO

- [1] Hamzah A.S., Abdullah Bade. Bounding Volume Hierarchies for Collision Detection. InTech (www.intechopen.com) Mar. 2012, DOI: 10.5772/35555. pp.39-54.
- [2] Kassper A.Andersen,, Christian Bay. A survey of algorithms for construction of optimal Heterogeneous Bounding Volume Hierarchies. Technical Report. Copenhagen, Denmark: Department of Computer Science, University of Copenhagen. 2006.
- [3] Herman J. Haverkort, Introduction to bounding volume hierarchies. PhD Thesis Chapter 1, 2004.
- [4] Gottschalk. Collision Queries using Oriented Bounding Boxes. PhD thesis, Department of Computer Science, University of North Carolina, 2000.
- [5] Lin, M.C., Gottschalk, S.: Collision detection between geometric models: a survey. In: Proc. IMA Conference on the Mathematics of Surfaces, pp. 37–56, 1998
- [6] Herman J. Haverkort, Introduction to bounding volume hierarchies. PhD Thesis Chapter 1, 2004.
- [7] Akenine-Moller, T., Hains, E.: Real-Time Rendering. A K Peters, 2002
- [8] Gottschalk, S., Lin, M.C., Manocha, D.: OBB-Tree: a hierarchical structure for rapid interference detection. In: ACM SIGGRAPH 1996, pp. 171–180, 1996.
- [9] Hubbard, P.M.: Collision detection for interactive graphics applications. IEEE Trans. on Visualization and Computer Graphics 1(3), 218–230, 1995.
- [10] Beckmann, N., Kriegel, H.-P., Schneider, R., Seeger, B.: The R -Tree: an efficient and robust access method for points and rectangles. In: ACM SIGMOD Conf. on the Management of Data, pp. 322–331, 1990.
- [11] Palmer, I., Grimsdale, R.: Collision detection for animation using sphere-trees. Computer Graphics Forum 14(2), 105–116, 1995
- [12] K. Erleben. An Introduction to Approximating Heterogeneous Bounding Volume Hierarchies . Technical Report DIKU-TR-02/04, DIKU, 2002
- [13] Klosowski, J.T., Held, M., Mitchell, J.S.B., Sowizral, H., Zikan, K.: Efficient collision detection using bounding volume hierarchies of k-DOPs. IEEE Trans. on Visualization and Computer Graphics 4(1), 21–37, 1998.
- [14] Van den Bergen, G.: Efficient collision detection of complex deformable models using AABB trees. J. Graphics Tools 2(4), 1–14, 1997.

- [15] Jepprey Goldsmith, John Salmon, Automatic creation of Object Hierarchy for Ray tracing. IEEE CG&A, 1987.
- [16] J. Goldsmith and J. Salmon. Automatic Creation of Object Hierarchies for Ray Tracing . IEEE CGA, 1987.
- [17] J. Haber, M. Staminger, and H. Seidel. Enhanced Automatic Creation of Multi- Purpose Object Hierarchies . IEEE CGA, 2000.
- [18] G. Müller, S. Schafer, and D. W. Fellner. Automatic Creation of Object Hierarchies for Radiosity Clustering . Technical Report TUBS-CG-1999-06, TU Braunschweig, 1999.
- [19] K. Erleben, J. Sparring, K. Henriksen, and H. Dohlmann. Physics-Based Animation . Charles River Media, 2005.
- [20] K. Erleben. An Introduction to Approximating Heterogeneous Bounding Volume Hierarchies . Technical Report DIKU-TR-02/04, DIKU, 2002.
- [21] Opentissue: Opensource Project, Physics-Based Animation and Surgery Simulation. <http://www.opentissue.org>.
- [22] Trenkel, S., Weller, R., Zachmann, G.: A Benchmarking Suite for Static Collision Detection Algorithms. In: International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG), 2007

ON THE METHOD FOR BUILDING A BOUNDING VOLUME HIERARCHY AUTOMATICALLY FOR 3-D OBJECTS

Abstract: Bounding Volume Hierarchy (BVH) is a tree-like structure for a set of geometric objects. The selection of bounding areas is usually defined on the basis of matching objects and often follows the top-down or bottom-up models or the add-in models for specific bounding boxes. For 3D objects, collisions may occur between objects. This paper discusses how to build a Bounding Volume Hierarchy (BVH) automatically for 3D objects. The proposed method is based on the use of many different types of bounding boxes suitable for the actual operation of the objects. The method has been tested and proved effective for 3D object models built on a continuous manner.

Keyword: Bounding Volume Hierarchy, multiple bounding boxes, collision detection.



Nguyễn Đức Hoàng, Nhận học vị Thạc sỹ năm 2013

Hiện công tác tại Học viện Công nghệ Bưu chính Viễn thông. Lĩnh vực nghiên cứu: Công nghệ trí thức, điện toán đám mây, khai phá dữ liệu, xử lý ảnh, học máy.