BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

NGUYỄN THỊ DIỄM MY

RAY TRACING CHO ĐỐI TƯỢNG ĐỒ HỌA

ĐỒ ÁN NGÀNH

CÔNG NGHỆ THÔNG TIN

TP. HỒ CHÍ MINH, 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

NGUYỄN THỊ DIỄM MY

RAY TRACING CHO ĐỐI TƯỢNG ĐỒ HỌA

Mã số sinh viên: 1851050091

ĐỒ ÁN NGÀNH

NGÀNH CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn: ThS. VÕ THỊ HỒNG TUYẾT

TP. HỒ CHÍ MINH, 2021

Mẫu số: 5

TRƯỜNG ĐẠI HỌC MỞ **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

THÀNH PHỐ HỒ CHÍ MINH **Độc lập – Tự do – Hạnh phúc**

KHOA CÔNG NGHỆ THÔNG TIN

———

GIẤY XÁC NHẬN

Tôi tên là: Nguyễn Thị Diễm My

Ngày sinh: 19/02/2000 ..... Nơi sinh: Bến Tre

Chuyên ngành: Công Nghệ Thông Tin .....Mã sinh viên: 1851050091

Tôi đồng ý cung cấp toàn văn thông tin đồ án hợp lệ về bản quyền cho Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh. Thư viện Trường Đại học Mở Thành phố Hồ Chí Minh sẽ kết nối toàn văn thông tin đồ án vào hệ thống thông tin khoa học của Sở Khoa học và Công nghệ Thành phố Hồ Chí Minh.

Ký tên

(Ghi rõ họ và tên)

Nguyễn Thị Diễm My

Mẫu số: 6

Ý KIẾN CHO PHÉP BẢO VỆ ĐỒ ÁN

CỦA GIẢNG VIÊN HƯỚNG DẪN

Giảng viên hướng dẫn: .....

Sinh viên thực hiện: ..... Lớp: .....

Ngày sinh: ..... Nơi sinh: .....

Tên đề tài: .....

.....

.....

.....

.....

Ý kiến của giảng viên hướng dẫn về việc cho phép sinh viên được bảo vệ đồ án trước Hội đồng:

.....

.....

.....

.....

.....

.....

.....

.....

Thành phố Hồ Chí Minh, ngày ... tháng ... năm ……

Người nhận xét

……

LỜI CẢM ƠN

Lời đầu tiên em xin chân thành cảm ơn toàn thể quý thầy cô trường Đại học Mở Thành phố Hồ Chí Minh đã tận tình truyền dạy kiến thức cho em trong suốt quá trình học tập tại trường. Nhờ những kiến thức và kỹ năng mà thầy cô truyền đạt đã trở thành hành trang cơ bản vững chắc để em vận dụng vào đồ án ngành.

Đặc biệt, em xin chân thành gửi lời cảm ơn sâu sắc đến Giáo Viên Hướng Dẫn - cô ThS. Võ Thị Hồng Tuyết khoa Công Nghệ Thông Tin đã trực tiếp hướng dẫn và chỉ bảo em trong suốt quá trình thực hiện và báo cáo môn đồ án ngành. Trong quá trình làm bài báo cáo em không thể tránh khỏi những thiếu sót nhưng đã được cô góp ý để hoàn thiện hơn.

Em xin chân thành cảm ơn ạ!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TÓM TẮT ĐỒ ÁN

Đề tài đồ án ngành “Ray tracing cho đối tượng đồ họa” được thực hiện trong khoảng thời gian từ tháng 8/2021 đến tháng 11/2021.Mục tiêu của đề tài là tìm hiểu về một số thuật toán ray tracing cơ bản và sử dụng kỹ thuật này để kết xuất đồ họa lại dạng hình học đơn giản. Điều này được hiện thực với thư viện hỗ trợ OpenGL cùng ngôn ngữ C++ qua Visual Studio 2019. Đối tượng này, ở đây là hình cầu có sự khác biệt về màu sắc trên bề mặt khuếch tán ở phần nhận được ánh sáng trực tiếp và phần bị khuất sáng khi thay đổi vị trí nguồn sáng.

Đề tài đạt đạt những kết quả cụ thể như sau: nắm bắt được một số thuật toán cơ bản về ray tracing trong kết xuất đồ họa hình cầu. Nhìn thấy được sự thay đổi màu trên bề mặt hình cầu khi sử dụng phím để thay đổi vị trí đặt nguồn sáng.

TÓM TẮT ĐỒ ÁN
(PHIÊN BẢN TIẾNG ANH NẾU CÓ)

MỤC LỤC

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Từ đầy đủ	Diễn giải
1	API	Application Programming Interface	phương thức trung gian kết nối các thư viện và ứng dụng khác nhau
2	OpenGL	Open Graphics Library	một API đa nền tảng, đa ngôn ngữ cho render đồ họa vector 2D và 3D
3	BVH	Bounding Volume Hierarchy	hệ thống phân cấp khối lượng giới hạn
4	NDC	Normalized Device Coordinates	tọa độ thiết bị chuẩn hóa
5	SBR	Shooting And Bouncing	phương pháp chụp và cắt
6	GPU	Graphics Processing Unit	bộ xử lý các tác vụ có liên quan đến đồ họa
7	AABB	Axis-Aligned Bounding Box	hộp giới hạn được căn chỉnh theo trục
8	OBB	Oriented Bounding Boxes	hộp giới hạn định hướng

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

MỞ ĐẦU

Phương pháp lập trình thường được sử dụng trong đồ họa máy tính đòi hỏi sinh viên phải học và tìm hiểu rất nhiều để có thể tạo ra được những hình ảnh căn bản nhưng lại không thực tế. Điều này dẫn đến việc dễ dàng bị lạc trong rừng của những lập trình ban đầu và những kỳ vọng cao sẽ dần biến mất. Cùng với đó, phương pháp tiếp cận lập trình phụ thuộc vào API đồ họa dựa trên các mô hình chiếu sáng cục bộ mà không hỗ trợ các mô hình chiếu sáng tổng quát. Điều này có nghĩa là không thể đổ bóng, phản xạ và khúc xạ chúng. Hơn nữa, một API đồ họa điển hình có khả năng mô hình hóa hạn chế.

Hãy thử một cách tiếp cận mới đó là kết hợp ray tracing với lập trình cùng máy tính và hình học. Sử dụng ray tracing là một cách tiếp cận tốt để giới thiệu các nguyên tắc cơ bản về đồ họa cho những người không có nền tảng sâu. Hơn thế nữa, lợi thế quan trọng nhất của việc sử dụng ray tracing là giúp tạo ra hình ảnh rực rỡ và truyền cảm hứng làm được nhiều hơn.

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu sơ lược

Với sự phát triển của ngành công nghiệp giải trí như điện ảnh và trò chơi, ngày càng có nhiều nhu cầu về đồ họa máy tính thực tế. Trong đó, kết xuất đồ họa là một chủ đề con riêng biệt nhưng lại có quan hệ chặt chẽ với các chủ đề con khác. Có nhiều kỹ thuật kết xuất để đạt được hình ảnh cuối cùng như rasterization (tạo điểm ảnh), radiosity (tính va đập của ánh sáng), ray tracing (dò tia sáng),… Trong đó, ray tracing là phương pháp có thể đạt đến cấp độ chân thực rất cao. Ray tracing dò đường đi của ánh sáng qua các điểm ảnh trên một mặt phẳng và mô phỏng các hiệu ứng quang học như phản xạ, khúc xạ, tán xạ, bóng tối,… để tạo ra hình ảnh.

Đề tài "Ray tracing cho đối tượng đồ họa" chủ yếu tìm hiểu về công nghệ ray tracing, các thuật toán liên quan và các bước cơ bản để thực hiện. Cụ thể là, ray tracing được áp dụng để xử lý bề mặt bị khuất sáng của hình cầu.

1.2. Nội dung và mục tiêu đề tài

Đề tài có những yêu cầu về nội dung và mục tiêu cụ thể như sau:

- Một phương pháp kết xuất hình ảnh.
- Tìm hiểu chi tiết các thuật toán và cách thực hiện.
- Kết xuất đồ họa một dạng hình học cơ bản (hình cầu): có sự khác biệt về mức độ sáng/tối trên bề mặt khi thay đổi vị trí nguồn sáng, có sự đổ bóng giữa các hình cầu.

1.3. Phương pháp thực hiện

Tham khảo các công trình nghiên cứu liên quan để nhận thấy ưu và nhược điểm chung của kỹ thuật ray tracing qua trình tự tiến hành. Dựa trên các tài liệu kết xuất đồ họa bằng kỹ thuật ray tracing để nắm rõ các bước thực hiện rồi lựa chọn các thuật toán phù hợp để áp dụng.

Đề tài "Ray tracing trên đối tượng đồ họa" sẽ được hiện thực trên Visual Studio 2019 với ngôn ngữ C++. Kết quả hình ảnh sẽ được hiển thị trên màn hình cửa sổ với sự hỗ trợ của thư viện OpenGL.

1.4. Giới hạn đề tài

Về không gian: tìm hiểu về công nghệ kết xuất hình ảnh hiện nay được sử dụng rộng rãi trong đồ họa máy tính là ray tracing.

Về nội dung: đề tài tìm hiểu và áp dụng thuật toán ray tracing để tính toán xử lý và dựng hình cầu. Hình cầu này có sự khác nhau về màu sắc trên bề mặt nhận ánh sáng trực tiếp và bề mặt khuất sáng. Tính toán đổ bóng hình cầu này lên hình cầu khác trong cảnh.

Về công nghệ: sử dụng ngôn ngữ lập trình C++ với thư viện OpenGL với Visual Studio 2019.

1.5. Bố cục báo cáo

Nội dung báo cáo gồm 4 chương, cụ thể như sau:

- Chương 1. Giới thiệu đề tài: giới thiệu sơ lược về đề tài, nêu lên mục tiêu cần và nội dung chính, phương pháp thực hiện và giới hạn đề tài.
- Chương 2. Cơ sở lý thuyết: chi tiết các khái niệm liên quan đến đề tài và sơ lược một số công trình nghiên cứu liên quan đến ray tracing.
- Chương 3. Phương pháp đề xuất: sơ đồ tổng quát và nêu chi tiết các bước hiện thực.
- Chương 4. Kết quả: kết quả đạt được, rút ra ưu nhược điểm và cho biết định hướng phát triển trong tương lai.

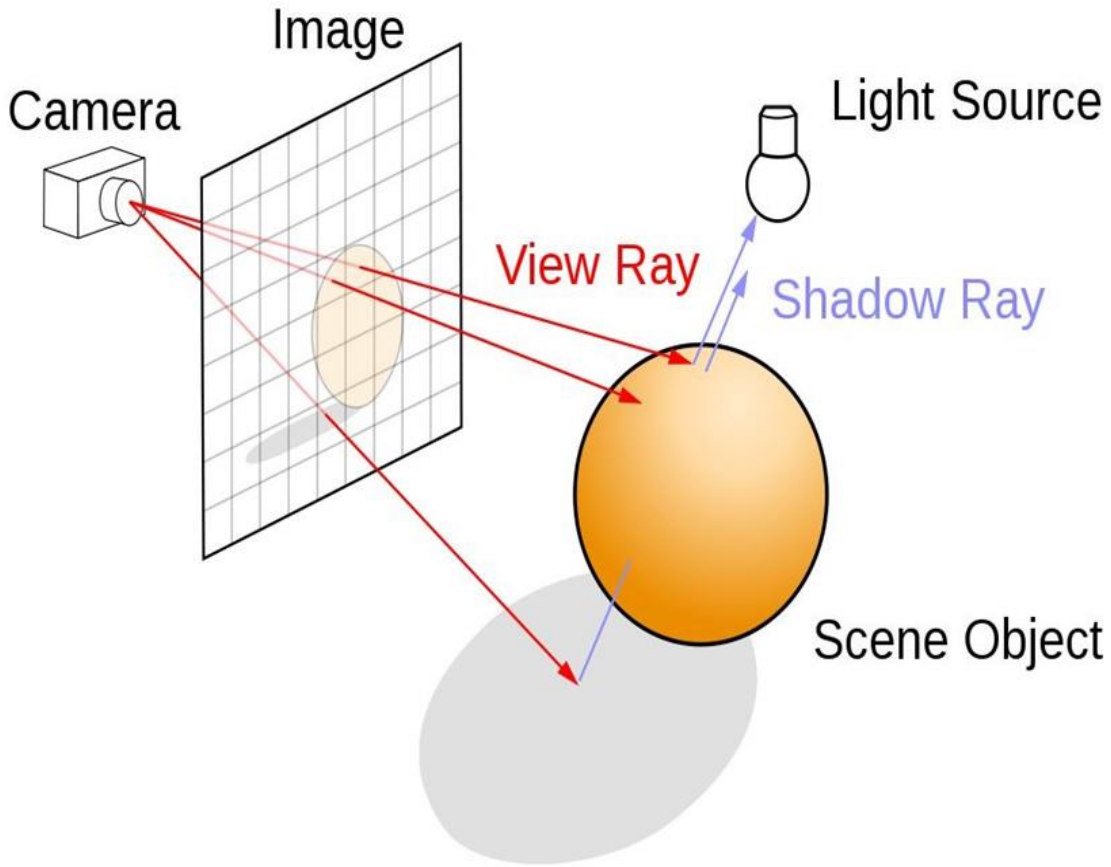
Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Kiến thức cơ bản

2.1.1. Tổng quan ray tracing

2.1.1.1. Định nghĩa

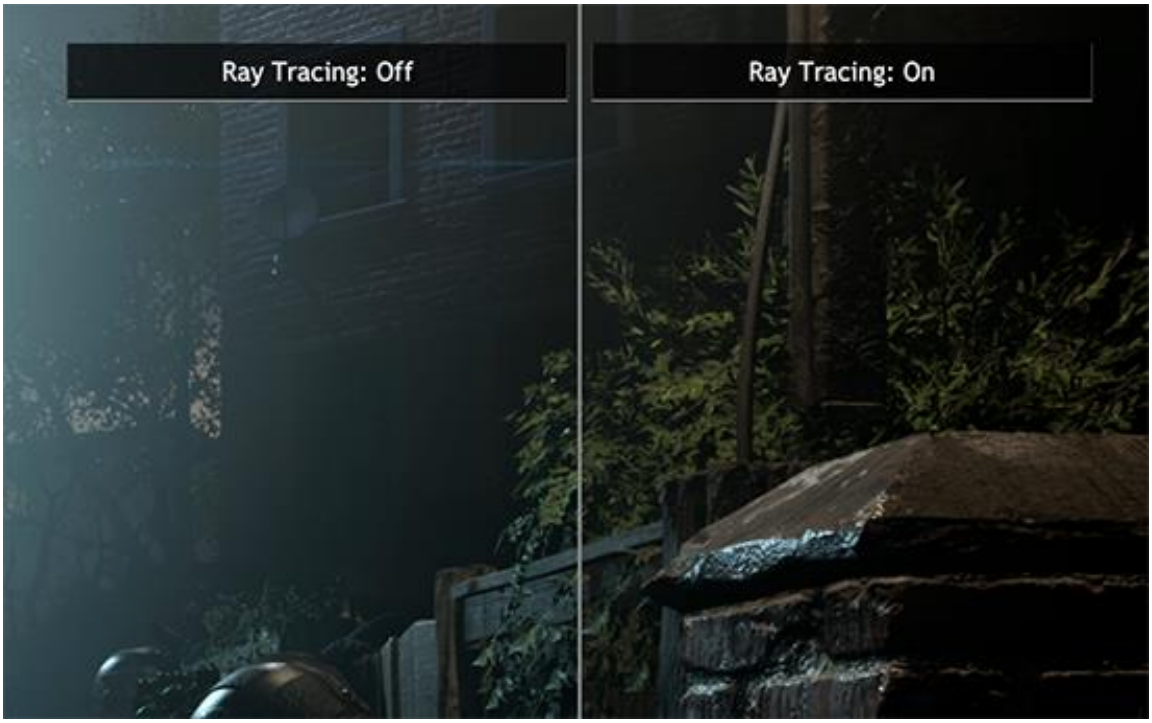
Ray tracing là một kỹ thuật dựng có thể mô phỏng thực tế ánh sáng của cảnh và các đối tượng trong cảnh bằng cách hiển thị các phản xạ, khúc xạ, bóng đổ và ánh sáng gián tiếp chính xác về mặt vật lý. Nói cách khác, ray tracing tạo ra một hình ảnh đồ họa máy tính bằng cách theo dõi đường đi của ánh sáng từ máy ảnh quan sát qua mặt phẳng xem 2D, ra cảnh 3D rồi quay trở lại nguồn sáng. Khi đi ngang qua khung cảnh, ánh sáng có thể phản xạ từ đối tượng này sang đối tượng khác (gây ra phản xạ), bị đối tượng chặn lại (gây ra bóng) hoặc đi qua đối tượng có tính chất trong suốt hay bán trong suốt (gây khúc xạ). Tất cả những tương tác này được kết hợp để tạo ra màu sắc và độ chiếu sáng cuối cùng của một điểm được hiển thị sau đó trên màn hình [1].



Hình 2-1 Ray tracing cơ bản [1]

2.1.1.2. Mục tiêu chính

Mục tiêu chính của ray tracing là tạo ra hiệu ứng chiếu sáng trông chân thực hơn, đổ bóng tự nhiên hơn,… Đây là kỹ thuật thường được ứng dụng vào game và hoạt động theo thời gian thực tạo nên các khung cảnh với độ chân thực cao. Ray tracing còn được sử dụng khi phát triển hình ảnh đồ họa máy tính trong một số chương trình truyền hình và phim ảnh. Cùng với tiến bộ gần đây trong công nghệ, nhiều khả năng ray tracing sẽ đóng vai trò lớn hơn trong các ứng dụng đồ họa tương tác ở tương lai. Hình ảnh phía dưới sẽ cho thấy sự khác biệt khi bật và tắt ray tracing trong một cảnh game. Qua đó nhận thấy rằng, những nơi có ánh sáng trực tiếp chiếu vào trong cảnh sẽ sáng hơn so với những nơi bị vật cản sáng. Ngoài ra, đây còn là mảnh ghép làm những mảng đổ bóng chuyển biến nhẹ nhàng giữa những mảng sáng và tối.



Hình 2-2 Off/On ray tracing trong một cảnh game

2.1.1.3. Các nguyên tắc cơ bản

Các nguyên tắc cơ bản về ray tracing:

- Ray casting (đúc tia): là một quá trình trong thuật toán ray tracing. Ray casting bắn một hoặc nhiều tia từ mắt/máy ảnh qua mỗi điểm ảnh trong mặt phẳng hình ảnh. Tiến hành kiểm tra xem các tia có giao nhau với bất kỳ

gốc nào (hình tam giác) trong cảnh hay không. Nếu tia đi qua một điểm ảnh rồi đi ra ngoài cảnh 3D va chạm phải một điểm gốc thì khoảng cách dọc theo tia từ điểm gốc của mắt/máy ảnh đến điểm gốc được xác định. Dữ liệu màu từ điểm gốc sẽ đóng góp vào màu cuối cùng của điểm ảnh. Tia cũng có thể dội lại và đập vào các đối tượng khác và nhận thông tin về ánh sáng và màu sắc từ đối tượng đó [1].

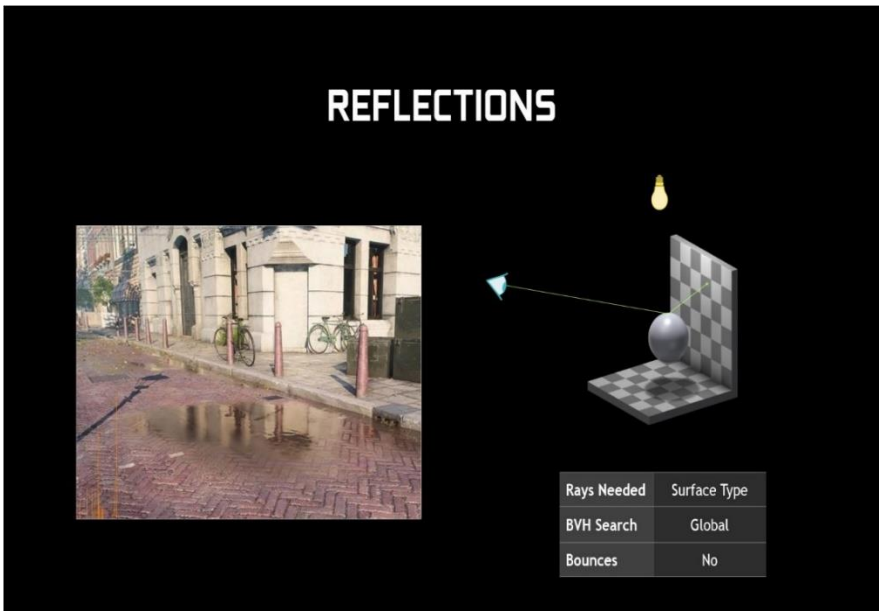
- Path tracing (theo dõi đường dẫn): là một loại của ray tracing. Trong path tracing, các tia chỉ tạo ra một tia duy nhất cho mỗi lần bị trả lại. Các tia này không theo một đường định sẵn cho mỗi lần trả lại mà bắn ra theo một hướng ngẫu nhiên. Thuật toán path tracing sau đó sẽ lấy mẫu ngẫu nhiên của tất cả các tia để tạo ra hình ảnh cuối cùng [2].
- BVH: là một kỹ thuật gia tốc ray tracing phổ biến. BVH sử dụng "cấu trúc gia tốc" dựa trên cây chứa nhiều hộp giới hạn được sắp xếp theo thứ bậc (khối lượng giới hạn) bao gồm hoặc bao quanh các lượng hình học trong cảnh hoặc nguyên thủy khác nhau [3].
- Denoising Filtering (lọc giảm nhiễu): là phần không thể thiếu của đường ống ray tracing. Trong cài đặt thời gian thực, người ta chỉ có thể tạo ra một số ít tia trên mỗi điểm ảnh, thường được phân phối ngẫu nhiên nên rất huyên náo. Bằng cách kết hợp các đánh giá trên mỗi điểm ảnh thưa thớt (nhưng đúng) này với các bộ lọc không gian, phương sai được giảm đáng kể với chi phí tăng độ chệch. Hơn nữa, mỗi tia là một mẫu điểm có thể giới thiệu răng cưa. Bằng cách lọc trước các thuật ngữ nếu có thể như tra cứu kết cấu thì răng cưa có thể được giảm bớt.

2.1.1.4. Các tính năng chính

Công nghệ Ray Tracing thường được chia thành nhiều bộ phận nhỏ và đóng gói, thường bao gồm bốn tính năng chính sau đây: [4]

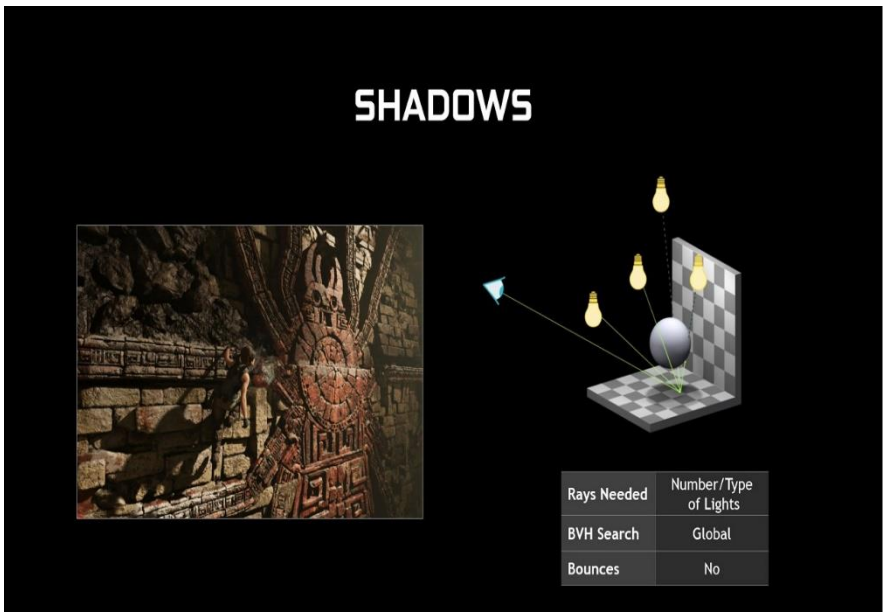
- Ray-traced reflections (phản chiếu): khi ray tracing được ứng dụng thì mới thực sự tạo ra các hình ảnh phản chiếu đúng nghĩa. Trước đây, phản chiếu được thực hiện bằng việc sử dụng những thủ thuật đồ họa để dựng thêm một hình ảnh giống hệt. Ray-traced reflection chiếu lại chi tiết ngoài màn hình, chi tiết xem ở góc thấp và chi tiết bị tắc trên màn hình. Tính năng

này còn tạo ra các phản chiếu thời gian thực, toàn cảnh, có độ phân giải cao mà các kỹ thuật trước đó không làm được.



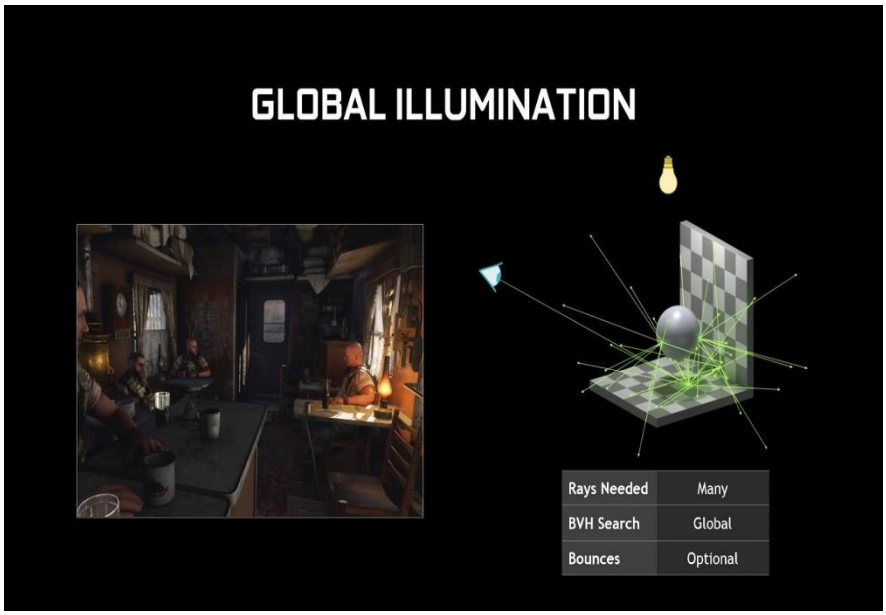
Hình 2-3 Ray-traced reflections [4]

- Ray-traced shadows (đổ bóng): tuân theo định luật vật lý dò tất cả các nguồn sáng có mặt trong cảnh. Sau đó, xác định độ che khuất của đối tượng so với nguồn sáng dẫn đến những bóng đổ giống như thật. Tính năng này không những tạo ra những bóng đổ chính xác mà còn có thể tạo ra bóng mờ ở mức chi tiết trong thời gian thực. Đặc biệt, dù là một điểm duy nhất thì các nguồn sáng ở các góc khác nhau cũng phải được dò riêng biệt để hiển thị bóng chính xác.



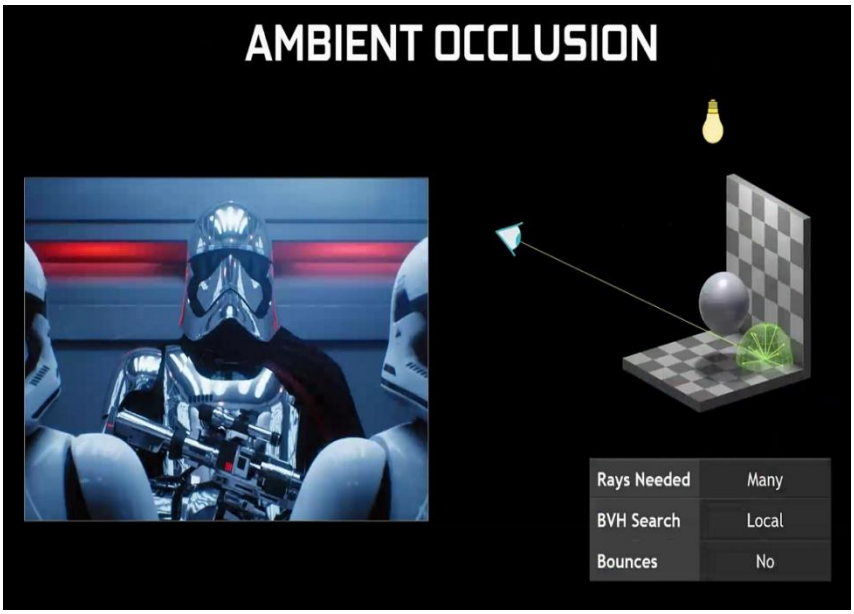
Hình 2-4 Ray-traced shadows [4]

- Ray-traced global illumination (chiếu sáng tổng thể): mô hình hóa chính xác hơn ánh sáng khuếch tán phản xạ bởi một hay nhiều tia trả lại gián tiếp từ các bề mặt trong cảnh. Các tia tạo ra bởi các bề mặt trong vùng tối lấy mẫu các bề mặt khác để tính toán lượng ánh sáng bị phản xạ theo hướng của chúng. Thuật toán hoạt động bằng cách mỗi điểm ảnh gửi tập hợp các tia của nó theo phân bố ngẫu nhiên giả trong một mái vòm phía trên bề mặt mà điểm ảnh đó thuộc về.



Hình 2-5 Ray-traced global illumination [4]

- Ray-traced ambient occlusion (đổ bóng môi trường): tính toán lượng điểm ảnh trên cảnh tiếp xúc với nguồn sáng và áp dụng đổ bóng. Các điểm ảnh được đổ bóng theo tính toán luồng ánh sáng đem đến cảm giác nổi khối chân thực hơn với các luồng sáng động thì các khu vực đổ bóng cũng tự nhiên hơn. Thay vì theo dõi tia sáng riêng lẻ và xác định bề mặt giao nhau thì ước tính có bao nhiêu tia có thể tiếp cận một bề mặt dựa trên hình dạng của môi trường xung quanh.



Hình 2-6 Ray-traced ambient occlusion [4]

2.1.2. Kỹ thuật kết xuất ray tracing

Kỹ thuật kết xuất ray tracing là cách tạo ra hình ảnh của cảnh 3D bằng cách sử dụng thuật toán ray tracing. Để tạo ra hình ảnh của cảnh 3D thật thì cần trượt hình ảnh raster (được tạo ra từ các điểm ảnh) này dọc theo mặt phẳng hình ảnh của máy ảo. Bắn các tia qua mỗi điểm ảnh trong hình để tìm phần nào của cảnh mà điểm ảnh bao phủ. Nói cách khác đơn giản hơn, truyền một tia đi từ mắt/máy ảnh qua giữa điểm ảnh. Kiểm tra xem tia giao nhau với đối tượng nào trong cảnh. Phương tia được xác định bằng cách đi theo một đường từ điểm gốc của mắt/máy ảnh đến tâm điểm ảnh và kéo dài vào cảnh. Lặp lại quá trình này với tất cả điểm ảnh trong hình ảnh. Thiết lập màu điểm ảnh với màu đối tượng mà tia đi qua giữa điểm ảnh giao nhau để tạo ra hình ảnh của cảnh từ một góc nhìn cụ thể. Đối với những tia không giao với bất kỳ đối tượng nào trong cảnh thì thiết lập điểm ảnh thành màu đen hoặc màu nền [5].

Trong đồ họa máy tính, việc xác định màu thật trên bề mặt của đối tượng tại bất kỳ điểm nào còn tùy thuộc vào hình dáng và độ sáng. Về hình dáng, các đối tượng trong thế giới thật không bằng phẳng mà có ngoại hình nhìn chung rất phức tạp. Về độ sáng, màu sắc thay đổi còn tùy thuộc vào lượng ánh sáng nhận được và cường độ sáng.

Tóm lại, kết xuất hình ảnh của các đối tượng 3D bằng cách sử dụng thuật toán ray tracing có thể được chia làm 3 bước cơ bản. Các bước này sẽ được nêu chi tiết hơn trong những mục tiếp theo, khái quát như sau:

- Truyền tia vào cảnh: tạo một tia máy ảnh hay còn gọi là tia sơ cấp – tia đầu tiên chiếu vào cảnh cho mỗi pixel trong cảnh. Có thể sinh ra nhiều tia hơn từ tia sơ cấp gọi là tia thứ cấp. Những tia này dùng để xem một điểm nhất định trong cảnh có ở trong bóng tối không để tính các hiệu ứng đổ bóng.
- Kiểm tra các giao điểm tia – hình học: kiểm tra xem một tia có giao với đối tượng hay không. Việc này cần phải lặp lại qua tất cả các đối tượng có trong cảnh và kiểm tra tia so với đối tượng hiện tại.
- Đổ bóng: nếu tia sáng và đối tượng giao nhau thì sẽ tính toán tìm xem đối tượng đó trông như thế nào. Điều này còn dựa vào thuộc tính bề mặt, vật cản.

Ray tracing chỉ là một kỹ thuật được dùng để tính khả năng hiển thị giữa các điểm với ý tưởng đằng sau là tính toán giao điểm của tia với hình học. Tính toán màu sắc tại bất kỳ điểm nào trên bề mặt đối tượng cần phải mô phỏng cách ánh sáng phản xạ khỏi bề mặt vật thể đó. Nhiệm vụ này được gọi là thuật toán vận chuyển ánh sáng. Trong đó, thuật toán vận chuyển ánh sáng dựa trên ray tracing thường được gọi là Whitted ray tracing. Thuật toán này áp dụng các định luật quang học vật lý để tính toán phương phản xạ và khúc xạ của tia khi giao nhau với bề mặt có tính chất phản xạ hoặc trong suốt. Theo dõi đường đi để tìm ra màu của đối tượng xảy ra va chạm. Tại giao điểm của các tia phản xạ và khúc xạ - còn gọi là tia thứ cấp có thể xảy ra những trường hợp sau:

- Nếu bề mặt mờ đục hoặc khuếch tán thì chiếu một tia tối theo phương của ánh sáng trong cảnh để xem điểm đó có nằm trong vùng tối không.

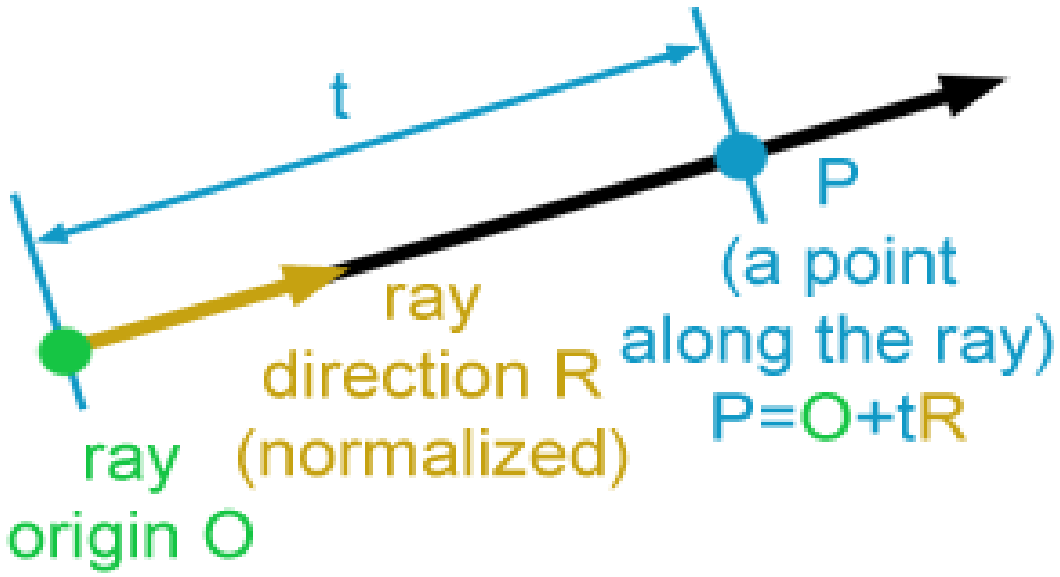
- Nếu bề mặt có tính chất phản chiếu như gương thì chiếu thêm một tia phản xạ khác tại giao điểm.
- Nếu bề mặt có tính chất trong suốt thì chiếu một tia phản xạ khác và một tia khúc xạ tại giao điểm.

2.1.3. Tạo ra các tia sơ cấp

2.1.3.1. Định nghĩa

Như đã đề cập ở phần trên, ray tracing là một kỹ thuật tính toán giao điểm giữa tia và bề mặt. Ray tracing còn được sử dụng để tính toán khả năng hiển thị bằng cách truyền một tia qua mỗi điểm ảnh của hình ảnh rồi xem tia đó giao với đối tượng gần nhất nào. Quá trình này còn được gọi là truyền tia. Trong đó, tia sơ cấp hay còn gọi là tia máy ảnh vì đây là những tia đầu tiên được truyền vào cảnh, có nguồn gốc từ máy ảnh. Thông thường, một tia có hai biến duy nhất là một điểm và một vectơ. Điểm biểu diễn gốc còn vectơ biểu diễn phương của tia. Về mặt toán học, bất kỳ điểm nào nằm trên nửa đường thẳng khi kết hợp điểm và phương đó lại với nhau có thể được định nghĩa theo phương trình tham số sau

$$P = O + tD \quad (1)$$



Hình 2-7 Thành phần của tia [6]

Trong đó:

- O: gốc tia
- D: phương tia
- t: khoảng cách từ điểm gốc đến điểm trên nửa đường thẳng

Xét các trường hợp với t:

- nếu $t < 0$ thì điểm P nằm sau gốc tia
- nếu $t > 0$ thì điểm P nằm phía trước gốc tia

Lưu ý trong ray tracing, với $t > 0$ thì giao điểm giữa tia và bề mặt được coi là hợp lệ. Do kỹ thuật ray tracing chỉ quan tâm việc tìm giao điểm giữa tia và các bề mặt nằm phía trước gốc tia. Quá trình tìm giao điểm tia và hình học (nếu tìm thấy) luôn trả về giao điểm theo tham số t. Nói cách khác, kết quả trả về bao gồm khoảng cách từ gốc tia đến giao điểm đó [6].

2.1.3.2. Tạo ra các tia sơ cấp

Hình ảnh được tạo ra bắt đầu từ việc xây dựng các tia sơ cấp. Trong các ứng dụng 3D, máy ảnh được tạo ra có vị trí mặc định là điểm gốc của thế giới xác định tại điểm có tọa độ $(0, 0, 0)$. Do nguồn gốc của máy ảnh là khẩu độ của máy ảnh lỗ kim nên phim của máy ảnh này trong thế giới thực đặt phía sau khẩu độ. Điều này khiến các tia sáng bằng cấu trúc hình học tạo ra hình ảnh đảo ngược của cảnh. Để tránh sự đảo ngược này, mặt phẳng phim phải nằm cùng phía với cảnh. Theo quy ước trong ray tracing, cụ thể như sau:

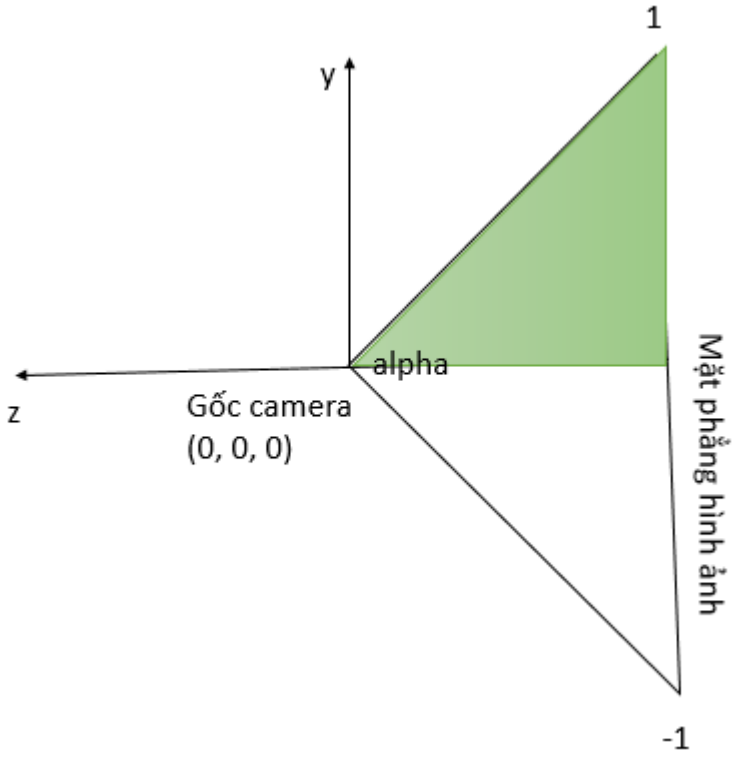
- Mặt phẳng đặt cách điểm gốc của máy ảnh đúng bằng một đơn vị.
- Máy ảnh dọc theo trục z âm rồi giả định hình ảnh được kết xuất có chiều ngang và chiều cao tính bằng điểm ảnh là như nhau.

Nhiệm vụ chính là tạo một tia sơ cấp cho mỗi điểm ảnh của khung. Tiến hành lần theo một đường thẳng bắt đầu từ điểm gốc của máy ảnh đi qua giữa mỗi điểm ảnh. Đường thẳng được biểu diễn dưới dạng một tia có gốc là gốc của máy ảnh, phương là vectơ từ gốc của máy ảnh đến tâm điểm ảnh. Chuyển tọa độ điểm ảnh được biểu thị ban đầu trong không gian raster đến không gian thế giới qua một vài bước:

- Tọa độ điểm được thể hiện trong không gian raster là tọa độ điểm ảnh cộng thêm độ lệch $0{,}5$.

- Tọa độ được ánh xạ lại thành phạm vi $[0, 1]$ chuyển đổi thành không gian NDC.
- Tọa độ NDC được ánh xạ lại thành phạm vi $[-1, 1]$ chuyển đổi thành không gian màn hình.
- Tọa độ trong không gian màn hình biến đổi sang không gian thế giới.

Những trường hợp hình ảnh có chiều rộng và chiều cao khác nhau thì các điểm ảnh bị móp méo. Để tránh vấn đề trên, chia tỷ lệ mặt phẳng hình ảnh dọc theo trục x với tỷ lệ khung tính từ chiều ngang chia cho chiều cao của hình.



Hình 2-8 Trường xem [7]

Tính toán trường nhìn bằng cách vẽ một tam giác được nối cạnh trên và cạnh dưới của mặt phẳng phim với gốc của máy ảnh được nhìn từ chế độ mặt bên của thiết lập máy ảnh. Khoảng cách từ gốc máy ảnh đến mặt phẳng phim là 1. Chiều cao mặt phẳng phim là 2 do nó đi từ $y = 1$ đến $y = -1$. Tính góc của tam giác vuông (nửa trên tam giác) bằng một số hàm lượng giác đơn giản. Nói cách khác, xác định trường xem của máy ảnh theo góc α rồi nhân với tọa độ của màn hình với kết quả của tiếp tuyến của góc này chia cho 2. Lúc này, tọa độ điểm ảnh ban đầu được biểu thị theo mặt phẳng hình ảnh của

máy ảnh được chuẩn hóa và ánh xạ lại giữa $[-1,1]$ nhân với tan trường góc xem $\frac{\alpha}{2}$ và nhân với tỷ lệ khung hình. Máy ảnh được căn dọc theo trục z âm nên tọa độ cuối cùng của điểm trên mặt phẳng hình ảnh được biểu diễn là $P = (Px, Py, -1)$. Tính tia cho điểm ảnh với gốc là gốc của máy ảnh và có phương từ gốc tia đến P. Di chuyển máy ảnh trong không gian để tạo khung theo ý muốn. Tuy nhiên, tạo tia cho mỗi điểm ảnh của khung và xem có tia nào trong số đó cắt hình học từ cảnh không để tạo ra hình ảnh cuối cùng, việc tìm giao điểm này sẽ được nêu chi tiết ở mục tiếp theo [7].

2.1.4. Thể hiện hình dạng

Trong đồ họa máy tính, các đối tượng hình học 3D được mô tả theo nhiều cách khác nhau như toán học, phương trình,… Có thể sử dụng các phương trình để tính toán phân tích về giao điểm giữa hình học với tia để tạo ra một hình dạng. Xác định về mặt toán học theo hai cách khác nhau là theo tham số và ngầm định.

Với bề mặt tham số [8], như đã trình bày ở mục 2.1.3.1 thì tia được xác định bằng phương trình tham số (1) với t là một tham số. Khi thay đổi giá trị của t thì có thể tùy thích thêm bao nhiêu điểm trên nửa đường thẳng tia và tập hợp chúng tạo thành nửa đường thẳng. Tương tự, một hình dạng có thể được mô tả bằng cách sử dụng phương trình. Các đối tượng 3D được xác định bằng các phương trình như vậy được gọi là các bề mặt tham số. Ví dụ, phương trình tham số của hình cầu là

$$P.x = cos(\theta)sin(\emptyset) \quad (2)$$

$$P.y = cos(\theta) \quad (3)$$

$$P.z = sin(\theta)sin(\emptyset) \quad (4)$$

Trong đồ họa máy tính, các tham số trong các phương trình tham số trong trường hợp chung có thể lấy sử dụng làm tọa độ kết cấu của một điểm trên bề mặt 3D của vật thể.

Với bề mặt ngầm định [8], một số hình học có thể được xác định bằng phương trình dạng ngầm định. Ví dụ, phương trình ngầm định của hình cầu là

$$x^2 + y^2 + z^2 = R^2 \quad (5)$$

Nhiều hình dạng được xác định bằng các phương trình như hình cầu, mặt phẳng, hình nón,… Một số hình dạng được sử dụng để biểu thị thể tích giới hạn của các đối tượng khác. Nếu đối tượng được bao bọc rất phức tạp thì sẽ rất tốn kiếm về mặt tính toán khi

kiểm tra xem một tia có cắt đối tượng này không. Vì lí do trên, trước tiên cần kiểm tra xem tia có cắt hình dạng giới hạn nó hay không. Nếu chúng không cắt nhau thì chắc chắn rằng tia không cắt đối tượng bên trong giúp tiết kiệm thời gian kiểm tra việc giao nhau giữa tia và chính đối tượng đó.

Tóm lại, cả hai cách biểu diễn trên đều khá đơn giản và cần thiết. Tuy nhiên, bề mặt tham số hữu ích hơn để tính toán tọa độ kết cấu của một điểm nằm trên bề mặt đối tượng. Trong khi đó, bề mặt ngầm định lại hữu ích hơn để tính toán kiểm tra giao điểm hình học với tia.

2.1.5. Giao của tia và hình cầu

Có rất nhiều phương trình truyền tia hiển thị hình ảnh của các hình cầu. Giao giữa tia và hình cầu được xem là dạng đơn giản nhất trong trong việc kiểm tra giao giữa tia và hình học. Như đã đề cập ở mục 2.1.3.1, một tia được biểu diễn dưới dạng sử dụng tham số $O + tD$. Trong đó, phương tia thường được chuẩn hóa. Có thể xác định bất kỳ vị trí nào dọc theo tia khi thay đổi giá trị của t:

- nếu $t > 0$ thì điểm nằm trước gốc của tia
- nếu $t = 0$ thì điểm sẽ trùng với gốc của tia
- nếu $t < 0$ thì điểm nằm sau gốc của tia.

Có hai giải pháp cơ bản thường được sử dụng là hình học và phân tích.

Với giải pháp hình học [9], các phép toán đơn giản như hình học, định lý Pytago và lượng giác được sử dụng. Khoảng cách từ gốc đến điểm giao nhau gần nhất

$$t_0 = t_{hc} - t_{ca} \tag{6}$$

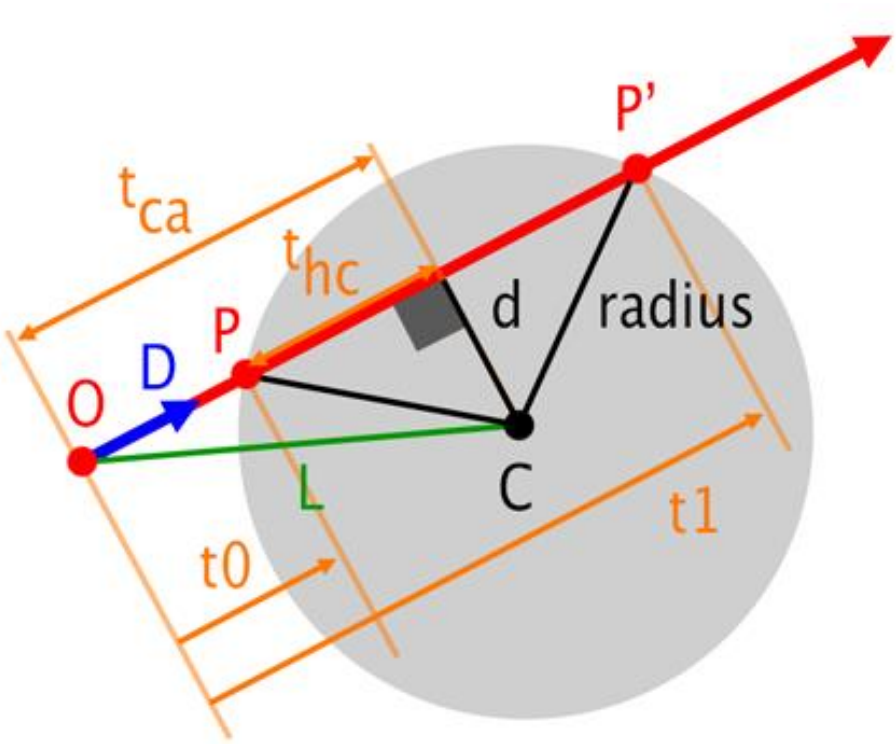
Khoảng cách từ gốc đến điểm giao nhau xa nhất

$$t_1 = t_{hc} + t_{ca} \tag{7}$$

Sử dụng phương trình tham số tia để tính hai điểm giao nhau trên

$$P = O + t_0 D \tag{8}$$

$$P' = O + t_1 D \tag{9}$$



Hình 2-9 Một tia giao hình cầu [8]

Tính tích vô hướng của $\vec{L}$ và $\vec{D}$ để được t_{ca}. Tia và hình cầu chỉ giao nhau nếu $t_{ca} > 0$ vì khi $t_{ca} < 0$ thì $\vec{L}$ và $\vec{D}$ ngược chiều nhau. cạnh L, t_{ca} và d tạo thành một tam giác vuông, áp dụng định lý Pytago tính d. Tương tự, tính t_{hc} trong tam giác vuông thứ hai được tạo bởi bán kính hình cầu $radius$, t_{hc} và d. Có giá trị của t_{hc} tính được t_0 và t_1.

Với giải pháp phân tích [9], sử dụng một dạng đại số để kiểm tra giao điểm. Xác định phương trình của một hình cầu: phương trình (5). Trong đó x, y, z là tọa độ một điểm cartesian, R là bán kính hình cầu có tâm tại điểm gốc. Coi x, y, z là tọa độ của điểm P, lúc này phương trình có dạng

$$P^2 - R^2 = 0 \tag{10}$$

Trong toán học và đồ họa máy tính, phương trình (10) là điển hình của một hàm ngầm định. Hình cầu được thể hiện dưới dạng này được gọi là một hình dạng hay bề mặt ngầm định. Tiếp theo kết hợp phương trình tham số tia (1) và phương trình (10)

$$|O + tD|^2 - R^2 = 0 \tag{11}$$

Triển khai tiếp

$$O^2 + (tD)^2 + 2OtD - R^2 = O^2 + t^2D^2 + 2OtD - R^2 \quad (12)$$

bản thân nó là một phương trình có dạng

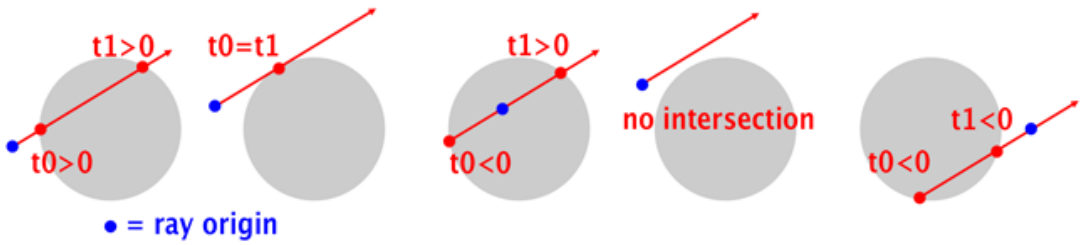
$$f(x) = ax^2 + bx + c \quad (13)$$

với $a = D^2, b = 2OD, c = O^2 - D^2$. Từ đây, dễ dàng tìm được các nghiệm nguyên bằng cách sử dụng phương trình

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (14)$$

$$\Delta = b^2 - 4ac \quad (15)$$

- Nếu $\Delta > 0$ thì tia cắt hình cầu tại hai điểm $\frac{-b+\sqrt{\Delta}}{2a}$ và $\frac{-b-\sqrt{\Delta}}{2a}$
- Nếu $\Delta = 0$ thì tia cắt hình cầu tại một điểm duy nhất $\frac{-b}{2a}$. Nếu $\Delta < 0$ thì tia không cắt hình cầu.



Hình 2-10 Các trường hợp xét tia với hình cầu [8]

Đối với những hình cầu không có tâm tại điểm gốc, phương trình được viết như sau

$$|P - C|^2 - R^2 = 0 \quad (16)$$

$$|O + tD - C|^2 - R^2 = 0 \quad (17)$$

điều này mang lại $a = 1$, b= $2D(O - C)$ và $c = |O - C|^2 - R^2$

Tiếp theo, tính điểm giao nhau sau khi đã biết được t_0 bằng cách sử dụng phương trình tham số tia

$$P_{hit} = O + t_0D \quad (18)$$

Tính toán pháp tuyến tại điểm giao nhau với phương pháp đơn giản bằng cách lấy vị trí điểm trừ đi tâm hình cầu rồi chuẩn hóa vectơ kết quả

$$N = \|P - C\| \quad (19)$$

Tọa độ kết cấu là tọa độ cầu của điểm trên hình cầu được ánh xạ lại thành phạm vi $[0, 1]$. Tọa độ Descartes tính từ tọa độ cầu của nó theo phương trình (2), (3), (4). Trong đó, θ và $\emptyset$ được tính từ tọa độ Descartes bằng phương trình

$$\theta = \text{atan}\,(z, x) \quad (20)$$

$$\emptyset = \frac{P.y}{R} \quad (21)$$

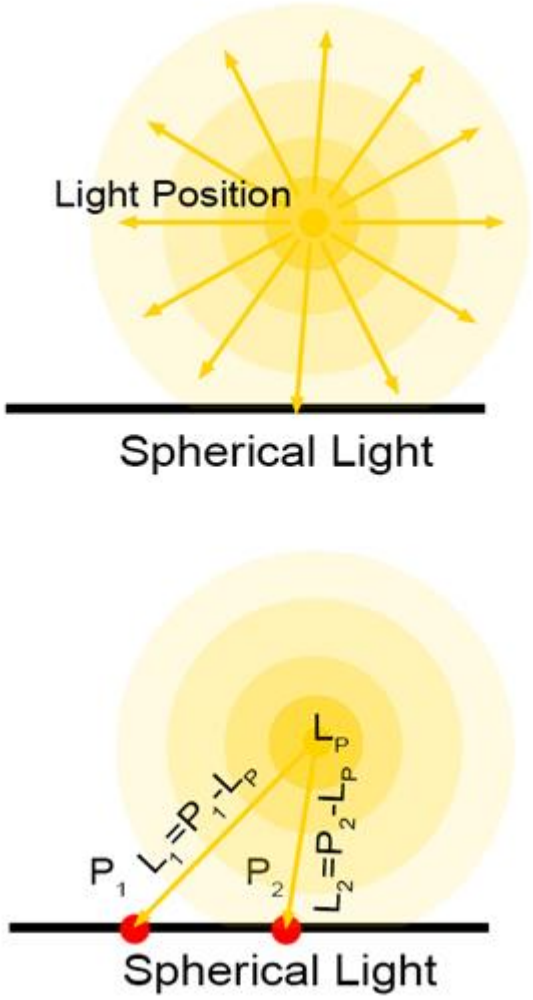
2.1.6. Đổ bóng và ánh sáng

2.1.6.1. Ánh sáng

Về cơ bản khi bắt đầu một tia tới vật thể và gửi đến nguồn sáng theo đường đi của các photon ngược. Nếu tia ngược này chạm tới nguồn sáng mà không chạm phải bất kỳ đối tượng nào trên đường đi thì chắc chắn một số các photon sẽ đi về phía trước dọc theo tia này từ ánh sáng để chiếu sáng vật. Tuy nhiên, nếu có bất kỳ đối tượng nào cản đường thì ánh sáng sẽ không thể nhận được thông qua đối tượng can thiệp. Sau đó nó sẽ ở trong bóng tối so với nguồn sáng đó. Khi một tia bóng tối tiếp cận nguồn sáng mà không bị gián đoạn thì khi đi qua sẽ là một tia sáng. Những tia sáng đến trực tiếp từ nguồn sáng chiếu sáng đối tượng lớp tia sáng đầu tiên góp phần tạo nên màu của ánh sáng khỏi đối tượng [10].

Nguồn sáng hình cầu là loại nguồn sáng phổ biến trong tự nhiên. Trong đồ họa máy tính, phương của tia mà nguồn sáng hình cầu phát ra theo hướng của một điểm muốn đổ bóng trên bề mặt đối tượng được tính đơn giản. Lấy điểm tô bóng trừ đi vị trí nguồn sáng

$$L_{dir} = P - L_{pos} \quad (22)$$



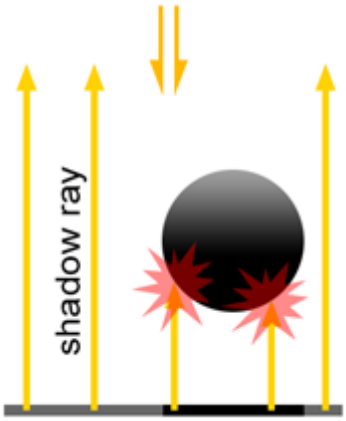
Hình 2-11 Ánh sáng hình cầu [10]

2.1.6.2. Đổ bóng

Tính toán màu sắc của các đối tượng trong cảnh 3D được gọi là đổ bóng. Trong kết xuất đồ họa, sự xuất hiện của một đối tượng phụ thuộc vào màu sắc đối tượng, hướng của bề mặt liên quan đến nguồn sáng trong cảnh và lượng ánh sáng nhận được. Tương tác vật chất ánh sáng và các thành phần thiết yếu của bóng bao gồm điểm đổ bóng, pháp tuyến tại điểm đổ bóng, phương sáng và phương xem. Pháp tuyến đóng vai trò quan trọng trong việc đổ bóng. Một vật sẽ trở nên sáng hơn khi được đặt hướng về phía nguồn sáng. Độ sáng của một điểm trên bề mặt đối tượng sẽ giảm nếu góc giữa phương sáng và pháp tuyến tăng. Với pháp tuyến có thể tạo ra một hiệu ứng đổ bóng đơn giản gọi là tỷ lệ đối mặt. Áp dụng ray tracing, phương xem được tính đơn giản là phương ngược lại

của tia giao với bề mặt tại một điểm. Tỷ lệ đối mặt được tính toán bằng tích số chấm của pháp tuyến tại điểm muốn tô bóng và phương xem [11].

Trong đồ họa máy tính, mô phỏng hành động đổ bóng của đối tượng lên đối tượng khác khá đơn giản. Với ray tracing, đổ bóng có thể được thực hiện khi hình ảnh đang hiển thị. Truyền một tia từ đối tượng có thể nhìn thấy qua một điểm ảnh cụ thể của hình ảnh từ điểm giao tới nguồn sáng. Nếu tia bóng này cắt một đối tượng trên đường tới ánh sáng thì điểm đang tô bóng nằm trong bóng của đối tượng đó. Khi đó điểm sẽ có màu đen. Cụ thể là, khi truyền tia sơ cấp vào cắt phải một đối tượng thì tính giao điểm và bắn thêm một tia bóng nữa theo hướng sáng ngược lại.



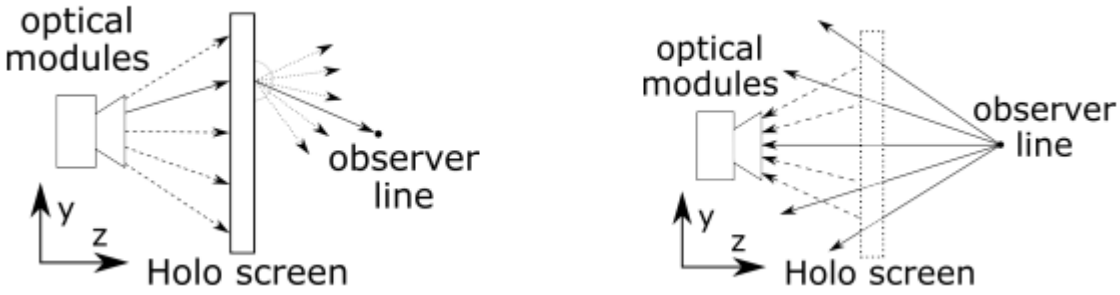
Hình 2-12 Bắn tia tối [11]

2.2. Các công trình nghiên cứu liên quan

2.2.1. Ray tracing cho màn hình hiển thị lĩnh vực ánh sáng HoloVizio

Ray tracing cho màn hình hiển thị lĩnh vực ánh sáng HoloVizio [12] mô tả việc triển khai thuật toán ray tracing cho một hệ thống kết xuất 3D. Cụ thể là cho màn hình trường ánh sáng thay đổi vị trí của một vật thể theo chiều ngang HoloVizio – hệ thống màn hình 3D thực. Công trình triển khai thuật toán như sau:

- Tạo các tia sơ cấp: các tia sơ cấp để ray tracing được tạo ra phải nghịch đảo với các tia chạm vào đường quan sát từ mô-dun quang học.



Hình 2-13 Tia sáng và tia sơ cấp [12]

Tính toán gốc và phương của tia sơ cấp trong không gian vật lý cho một vị trí (x, y) nhất định trong không gian hình ảnh.

- Triển khai các tính năng: sử dụng BVH với giới hạn theo trục hộp làm cơ cấu gia tốc. Mảng các nút BVH được tải lên bộ nhớ GPU bằng OpenGL thích hợp đệm. Giai đoạn kết xuất của thuật toán được triển khai như hai bộ đổ bóng máy tính: bộ đổ bóng truyền qua và vẽ đổ bóng. Sau khi áp dụng ánh sáng và đổ bóng làm cho kết cấu cuối cùng được sử dụng thêm bởi mô-đun quang học HoloVizio.

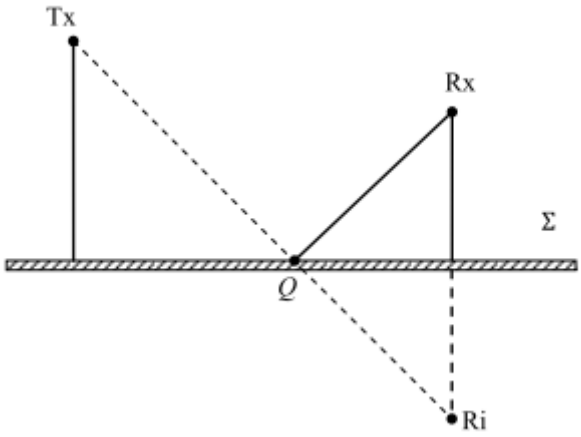
Công trình này cho thấy ưu điểm của việc áp dụng thuật toán ray tracing là nhận được thông tin vị trí chính xác cho tất cả các điểm nhìn thấy của ảnh ảo. Tuy nhiên, kích thước hầu hết của các hình học nguyên thủy trong hầu hết cảnh là nhỏ. Do đó, sự khác biệt hình ảnh trực quan mô-đun quang học thu được khó có thể nhận ra ray tracing.

2.2.2. Ray tracing để lập mô hình truyền sóng vô tuyến

Ray tracing để lập mô hình truyền sóng vô tuyến [13] xem xét các khái niệm cơ bản về tia, thuật toán ray tracing và sự lan truyền vô tuyến mô hình hóa bằng cách sử dụng phương pháp Ray Tracing. Thuật toán ray tracing đơn giản được xem xét trong công trình này như sau:

- Nguyên tắc Fermat trong thời gian ít nhất: xác định cách một tia tìm thấy đường dẫn của nó di chuyển từ vị trí này đến vị trí khác trong sự truyền bá môi trường.
- Phương pháp hình ảnh: đầu tiên xác định vị trí hình ảnh R_x, R_i đối với bề mặt phản xạ phẳng. Sau đó kết nối T_x và R_i để có đoạn thẳng với các giao điểm của

mặt phẳng tại một điểm Q. Khi đó đường đi của tia phản xạ được xác định là (T_x, Q, R_x).



Hình 2-14 Mô phỏng phương pháp hình ảnh [13]

- Phương pháp SBR: theo dõi mọi tia được phóng ra từ một vị trí nguồn để xác định có đến một trường điểm không. Bao gồm ba bước: phóng tia, dò tia và thu nhận tia.
- Phương pháp Hybrib: đầu tiên sử dụng phương pháp SBR xác định một tia hợp lệ sau đó dùng phương pháp hình ảnh để điều chỉnh quỹ đạo tia.

Tác giả hình dung mô hình lan truyền trong tương lai gần như một hệ thống thông minh, chính xác và xem hệ thống thời gian thực trong Ray Tracing nào đóng vai trò quan trọng. Đánh giá này đặc biệt hữu ích cho các chuyên gia đang phát triển các thuật toán ray tracing mới để nâng cao độ chính xác của mô hình và cải thiện tốc độ tính toán.

2.2.3. Ray tracing phân giải mắt người theo thời gian thực

Ray tracing phân giải mắt người theo thời gian thực [14] kiểm tra tính khả thi của việc chỉ sử dụng ray tracing trong hình ảnh trực quan. Đây là nơi các cảnh có thể bao gồm các thay đổi trên mỗi khung hình. Có chứa các loại đối tượng 3D khác nhau như với các hình dạng trừu tượng và những hình có dữ liệu tam giác phổ biến hơn. Bao gồm các việc triển khai liên quan như sau:

- kết xuất đường ống
- độ phân giải động
- nguồn sáng và vật liệu
- thiết lập cảnh

Trong nghiên cứu này, có các tùy chọn tia cắt đối tượng được liệt kê:

- Hình dạng trừu tượng: AABB, OBB và hình cầu có thể thực hiện một số phép toán đơn giản.
- Giao lưới tam giác: có thể sử dụng dữ liệu kết cấu để truy cập dữ liệu nhanh.
- Giao lộ Voxel (giá trị trên một lưới thông thường): cần ít giao nhau sâu hơn.

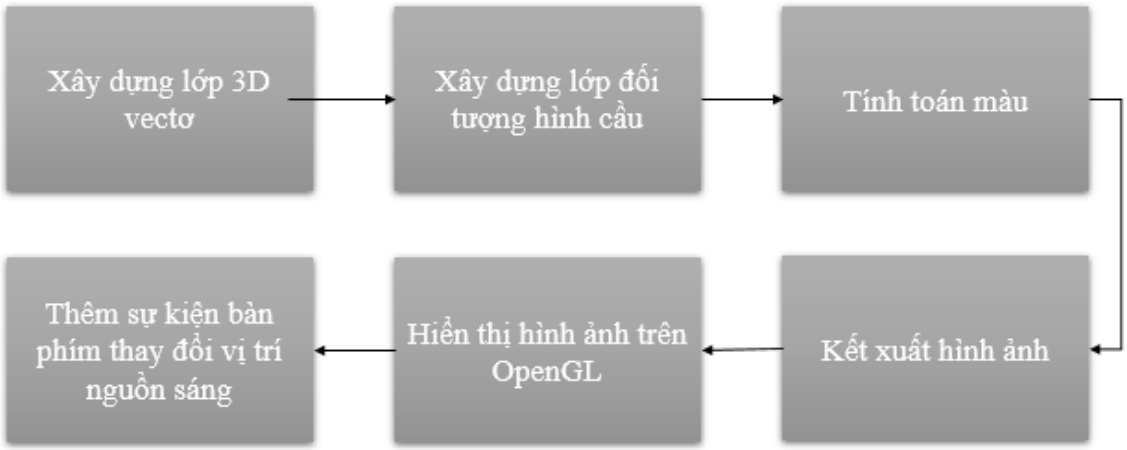
Công trình này giúp hình dung những cảnh đơn giản với các thay đổi thời gian thực mà giữ nguyên hiện trường động. Cấu trúc cảnh mỗi cơ sở khung có thể thay đổi. Tuy nhiên, đây là công trình khá phức tạp để thể hiện.

Chương 3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1. Sơ đồ các bước thực hiện

Kết xuất đồ họa bằng kỹ thuật ray tracing cho một đối tượng 3D là hình cầu. Đây là dạng hình học cơ bản và khá đơn giản để thực hiện. Hình ảnh nhận được có sự khác biệt về sắc độ trên bề mặt khuếch tán khi thay đổi vị nguồn sáng. Quá trình được hiện thực trên Visual Studio 2019 C++ với OpenGL, cụ thể qua 6 bước sau:

- Bước 1: Xây dựng một lớp 3D vectơ gồm tọa độ vectơ trên hệ trục tọa độ ba chiều để hỗ trợ tính toán.
- Bước 2: Xây dựng một lớp đối tượng là hình cầu bao gồm kiểm tra giao điểm và dữ liệu bề mặt.
- Bước 3: Tính toán màu trên bề mặt đối tượng nếu xảy ra giao giữa tia và hình cầu, nếu không thì trả về màu nền.
- Bước 4: Lặp từng điểm ảnh trong hình ảnh để tạo tia sơ cấp, truyền tia vào cảnh và nhận về màu của điểm ảnh đó ghi vào vùng nhớ.
- Bước 5: Đọc lại màu sắc các điểm đã lưu ở bước 4 và xuất lên cửa sổ màn hình đồ họa OpenGL hình ảnh cuối cùng.
- Bước 6: Thêm sự kiện bàn phím thay đổi vị trí nguồn sáng để thấy sự khác biệt về màu sắc phần bề mặt nhận sáng trực tiếp và phần bề mặt bị khuất sáng.



Hình 3-1 Sơ đồ các bước thực hiện

3.2. Chi tiết các bước

3.2.1. Xây dựng lớp 3D vectơ

Xây dựng một lớp Vec3 bao gồm các hàm thành viên công cộng:

- Các hàm xây dựng vectơ: Vec3(), Vec3(x), Vec3(x, y, z)
- Hàm chuẩn hóa vectơ: normalize
- Các hàm khai thác vectơ: tích vô hướng, tích hữu hướng và một số phép toán cơ bản

3.2.2. Xây dựng lớp hình cầu

Xây dựng một lớp đối tượng là hình cầu:

- Khai báo constructor gán các thuộc tính:
  - tâm hình cầu
  - bán kính hình cầu
  - màu bề mặt
  - màu phát xạ, mặc định bằng 0
- Phương thức kiểm tra giao điểm của hình cầu với một tia:
  - trả về true, nếu tìm thấy giao điểm
  - trả về false, nếu không tìm thấy giao điểm

  áp dụng giải pháp hình học trong giao giữa tia và hình cầu, cụ thể các bước như sau:
  - cho giá trị trả về ban đầu là true
  - tính $\vec{L}$ (vectơ từ gốc tia đến tâm hình cầu)
  - tính t_{ca} (độ lớn của $\vec{L}$)
  - nếu $t_{ca} < 0$: trả về false, phương tia đi theo hướng ngược lại nên không cần phải xét thêm nữa
  - tính d^2 (bình phương đoạn thẳng kẻ từ tâm vuông gốc với tia): áp dụng định lý Pytago
  - so sánh d^2 với $radius^2$ (bình phương bán kính hình cầu): nếu $d^2 > radius^2$ thì trả về false, phương tia không đi trúng hình cầu.
  - tính t_{hc} (độ sâu của t_{ca} đi vào hình cầu): áp dụng định lý Pytago

- tính t_0 (độ lớn từ gốc tia đến giao điểm đầu tiên): áp dụng phương trình (6)
- tính t_1 (độ lớn từ gốc tia đến giao điểm thứ hai): áp dụng phương trình (7)

o Phương thức tính toán dữ liệu bề mặt:

- tính n_{hit} (tọa độ pháp tuyến tại điểm giao nhau): áp dụng phương trình (19) lấy p_{hit} (tọa độ điểm giao giữa tia và hình cầu) trừ đi tâm hình cầu, chuẩn hóa kết quả để biến thành một đơn vị làm cho phương trình đơn giản hơn

3.2.3. Tính toán màu

Tính toán màu sắc bề mặt đối tượng thật chất là tính màu tại điểm giao nhau của tia và đối tượng và trả về màu nền nếu không giao nhau:

o Khai báo các biến:

- t_{Near} (khoảng cách từ gốc tia đến điểm giau nhau gần nhất) đặt thành một số rất lớn ($INFINITY$), t_{Near} được cập nhật khi giao điểm với đối tượng được kiểm tra gần với giá trị thực của nó
- con trỏ $sphere$ kiểu hình cầu đặt giá trị $NULL$
- $bias$ (độ lệch làm thay đổi vị trí bóng) bằng $1e-4$
- $color$ (màu sắc trả về) bằng 0
- p_{hit}
- n_{hit}

o Tiến hành vòng lặp qua tất cả các hình cầu có trong cảnh:

- khai báo t_0 và t_1 bằng $INFINITY$
- nếu hình cầu đang xét giao với tia:

  + nếu $t_0 < 0$: gốc tia nằm trong hình cầu, sử dụng t_1 thay thế

  + nếu $t_0 < t_{Near}$: t_0 ở trước t_{Near} , cập nhật lại t_{Near} và cập nhật con trỏ $sphere$ đến hình cầu đang xét

o Nếu không có giao điểm với hình cầu thì trả về màu nền $(0.8, 0.8, 0.8)$

o Cập nhật p_{hit} thành điểm giao nhau gần nhất: áp dụng phương trình (18), t_0 trong phương trình (18) đổi thành t_{Near}

- Cập nhật n_{hit}: lấy $sphere$ hiện tại tham chiếu về phương thức tính toán dữ liệu bề mặt
- Tiến hành vòng lặp qua tất cả các hình cầu có trong cảnh lần nữa: nếu màu phát xạ của hình cầu đang xét lớn hơn 0 thì hình cầu đó là nguồn sáng:
    - khai báo $light_{Dir}$ (phương tia sáng): vectơ đi từ p_{hit} đến tâm nguồn sáng, chuẩn hóa vectơ
    - khai báo tms (truyền tải) bằng $(1.0, 1.0,1.0)$
    - lặp qua tất cả các hình cầu có trong cảnh:nếu hình cầu đang xét này khác hình cầu đang xét bên ngoài:
        - khai báo t_0 và t_1 mới
        - nếu hình cầu này giao với tia có gốc là $p_{hit} + (n_{hit} \times bias)$, phương là $light_{Dir}$ thì cập nhật tms nhân thêm với màu của hình cầu đó
    - cập nhật $color$: cộng thêm với tích của màu của $sphere$, tms, max((0,0,0), tích vô hướng n_{hit} và $light_{Dir}$) và màu phát xạ của nguồn sáng
- Trả về giá trị $color$ cộng thêm màu phát xạ của nguồn sáng

3.2.4. Kết xuất hình ảnh

Ghi nội dung từng điểm ảnh vào bộ nhớ: tạo tia sơ cấp và truyền vào cảnh. Lưu ý hình ảnh không phải hình vuông nên cần kéo dài cửa sổ theo màn hình tỷ lệ khung hình của hình ảnh. Cần nhân cửa sổ màn hình theo trục x với tỷ lệ khung hình ảnh và giữ bằng 1 ngang theo trục y. Cụ thể các bước kết xuất hình ảnh như sau:

- Khai báo toàn cục:
    - vectơ con trỏ tên $image$ trỏ đến từng phần tử của một mảng có kích thước là tổng số điểm ảnh của hình ảnh (màn hình hiển thị)
    - cam_{Pos} (vị trí máy ảnh) đặt tại $(0,0,0)$
- Trong hàm kết xuất chính:
    - khai báo các biến:
        + con trỏ $pixel$ bằng với $image$
        + fov đặt bằng 25
        + img_{Asp} (tỷ lệ khung ảnh) đặt bằng tỷ lệ chiều rộng với chiều cao

+ $angle$ có giá trị bằng $\tan(PI \times 0.5 \times \frac{fov}{180})$

- vòng lặp lồng qua từng điểm ảnh với y theo chiều cao, x theo chiều ngang:
    - khai báo $xx = \left(2(x + 0.5) \times \frac{1}{WIDTH} - 1\right) \times angle \times img_{Asp}$
    - khai báo $yy = (1 - 2(x + 0.5) \times \frac{1}{HEIGHT}) \times angle$
    - khai báo ray_{Dir} (phương của tia sơ cấp hay gọi là tia máy ảnh) có giá trị $(xx, yy, -1)$, chuẩn hóa ray_{Dir}
    - $pixel$ mà tia máy ảnh truyền qua được cập nhật là kết quả ở mục 3.2.3, với tia máy ảnh có gốc cam_{Pos} và phương ray_{Dir}

3.2.5. Hiển thị hình ảnh trên OpenGL

Hiển thị hình ảnh lên màn hình cửa số OpenGL qua đọc và xuất lại từng điểm ảnh. Cụ thể như sau:

- Khai báo một biến vectơ con trỏ toàn cục $spheres$
- Gọi hàm kết xuất (mục 3.2.4)
- Khai báo biến x, y: tọa độ được ánh xạ lại thành phạm vi $[-1,1]$ chuyển sang không gian màn hình
- Khai báo biến $i = 0$ (chỉ số điểm ảnh)
- Gọi hàm vẽ nhiều điểm:
    - vòng lặp lồng với giá trị biến lặp $y = 1$, điều kiện lặp $y > -1$, cập nhật giá trị biến lặp $y-= H_{step}$ với H_{step} là độ mịn theo chiều cao, tương tự với biến lặp là x theo chiều ngang:
        - thiết lập màu vẽ: giá trị $image$ từ hàm kết xuất theo chỉ số i
        - thiết lập tọa độ ở vị trí $(x, y, 1)$
        - kiểm tra nếu i còn nhỏ hơn tổng số điểm ảnh thì tăng lên một đơn vị
- Đặt thông số cho $spheres$, gọi hàm đẩy vào vị trí sau cùng của vectơ

3.2.6. Sự kiện bàn phím

Sử dụng phím để thay đổi vị trí đặt nguồn sáng qua đó thấy được sự thay đổi sắc độ của đối tượng là hình cầu trong cảnh. Khi gọi sự kiện bàn phím sẽ:

- Gọi hàm loại bỏ tất cả các đối tượng trước đó của vùng chứa vectơ
- Gọi hàm đẩy đối tượng mới vào, hình cầu sẽ có thông số như trước riêng nguồn sáng sẽ bị thay đổi vị trí
- Tính năng các phím như sau:
    - phím ‘a’ đặt vị trí nguồn sáng ở bên trái màn hình
    - phím ‘d’ đặt vị trí nguồn sáng ở bên phải màn hình
    - phím ‘s’ đặt vị trí nguồn sáng ở bên trên màn hình gần mắt
    - phím ‘w’ đặt vị trí nguồn sáng ở bên trên màn hình xa mắt hơn ‘s’
    - phím khác để thoát

Chương 4. KẾT QUẢ

4.1. Kết quả đạt được

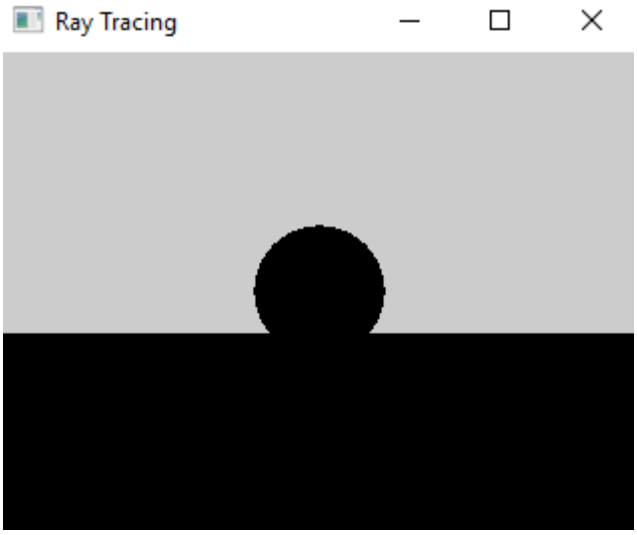
4.1.1. Các công cụ hỗ trợ

Sử dụng phần mềm Microsolf Visual Studio 2019 lập trình C++. Dùng OpenGL để hỗ trợ hiển thị hình ảnh kết quả.

4.1.2. Kết quả

Đề tài đạt được những kết quả khi chạy demo như sau:

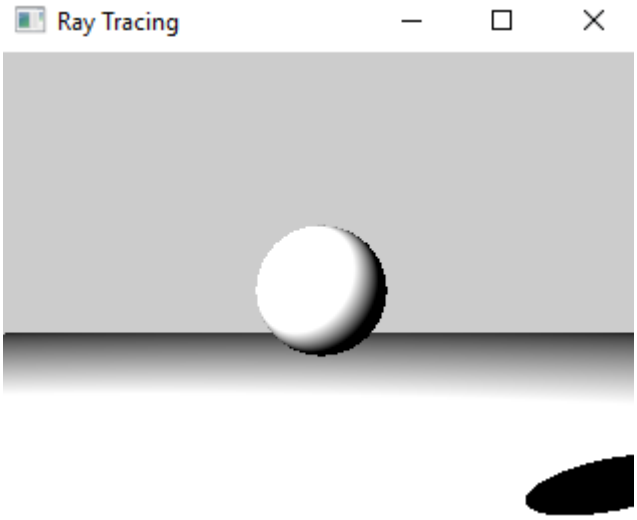
- Tiến hành chạy sẽ cho ra màn hình cửa sổ đồ họa là hai hình cầu (một vừa, một rất lớn) chưa được chiếu sáng



Hình 4-1 Màn hình hiển thị đầu tiên

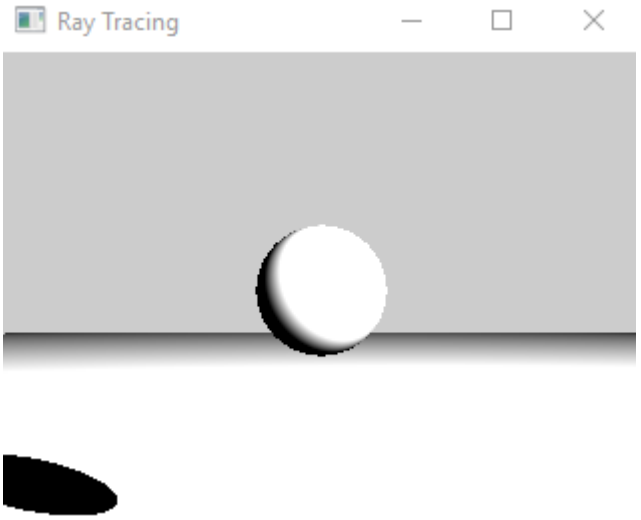
- Khi ấn phím đặt nguồn sáng vào sẽ nhận thấy: trên hình cầu nhỏ, phần bề mặt nhận ánh sáng trực tiếp sẽ có màu trắng của hình cầu và nguồn sáng, phần hoàn toàn bị khuất có màu đen. Trên hình cầu rất lớn, một phần bề mặt bị hình cầu nhỏ đổ bóng lên.

- ấn phím ‘a’ đặt nguồn sáng phía trên bên trái màn hình



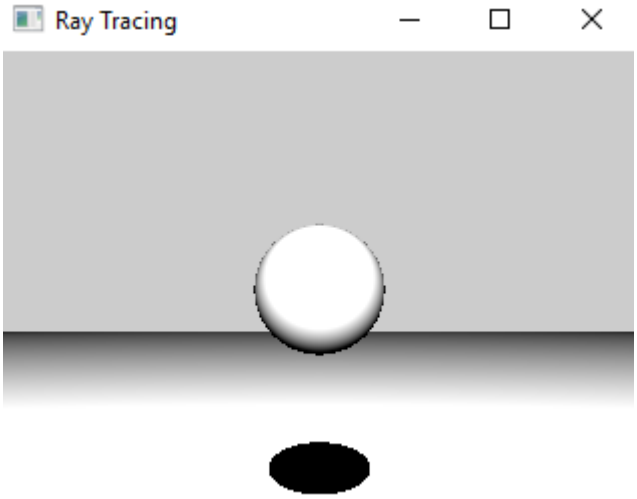
Hình 4-2 Kết quả ấn phím ‘a’

- ấn phím ‘d’ đặt nguồn sáng phía trên bên phải màn hình



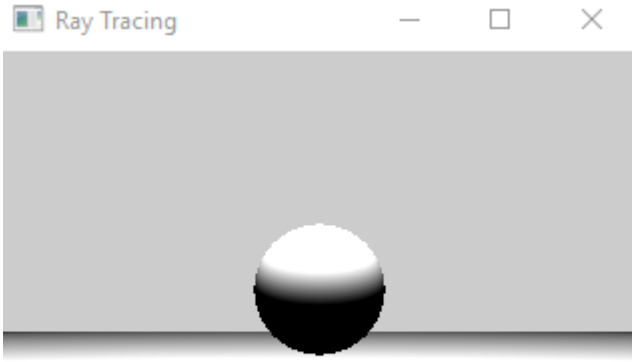
Hình 4-3 Kết quả ấn phím ‘d’

- ấn phím ‘s’ đặt nguồn sáng phía trên giữa màn hình gần mắt



Hình 4-4 Kết quả ấn phím ‘s’

- ấn phím ‘w’ đặt nguồn sáng phía trên giữa màn hình xa mắt



Hình 4-5 Kết quả ấn phím ‘w’

- ấn phím khác để thoát

4.2. Kết luận

4.2.1. Ưu điểm

Đề tài có những ưu điểm sau:

- So với các kỹ thuật khác trong kết xuất đồ họa, ray tracing có tính đúng đắn về vật lý hơn. Ray tracing có thể tính toán chính xác nâng cao các hiệu ứng quang học.
- Hiển thị màu sắc khác nhau của bề mặt đối tượng ở phần được nhận ánh sáng trực tiếp và phần bị khuất sáng.
- Hiển thị chính xác kết quả khi tự động kết hợp với các hiệu ứng tô bóng từ nhiều đối tượng theo đúng trình tự giúp cho phép xây dựng cá nhân các đối tượng và bộ đổ bóng của chúng một cách độc lập.

4.2.2. Nhược điểm

Đề tài có những nhược điểm sau:

- Việc tính toán nhiều chi tiết chiếm dụng khá nhiều tài nguyên.
- Thời gian kết xuất hình ảnh cuối cùng rất lâu khi phải tạo ra rất nhiều tia giao với hình học trong cảnh.
- Tương đối khá phức tạp khi tìm hiểu kết hợp và phân tích các giải thuật của nó.
- Sử dụng thư viện đồ họa OpenGL dù phù hợp với tính phổ biến và linh hoạt nhưng thư viện này lại kết hợp rất chặt chẽ với đường ống rasterization.

4.3. Định hướng phát triển trong tương lai

Một số định hướng phát triển trong tương lai:

- Áp dụng ray tracing để kết xuất hình ảnh với những dạng hình học phức tạp hơn.
- Tạo ra các mô hình 3D từ các hình học nguyên thủy, xử lý để trở nên mềm và chân thực như ngoài thế giới tự nhiên.
- Triển khai đối với video, theo dõi chi tiết hơn sự thay đổi sắc độ của các vật thể trong cảnh khi thay đổi hay chuyển động nguồn sáng. Xuất hiện những vật thể mới hay trong quá trình vật thể di chuyển va phải vật thể với những tính chất khác nhau.
- Tối ưu hóa chi phí tính toán và thời gian kết xuất.

TÀI LIỆU THAM KHẢO

[1] NVIDIA, "Ray Tracing," [Online]. Available: https://developer.nvidia.com/discover/ray-tracing. [Accessed 15 8 2021].

[2] J. Wang, "Future of Gaming : Rasterization vs Ray Tracing vs Path Tracing," [Online]. Available: https://medium.com/@junyingw/future-of-gaming-rasterization-vs-ray-tracing-vs-path-tracing-32b334510f1f. [Accessed 22 8 2021].

[3] Wikipedia, "Bounding volume hierarchy," [Online]. Available: https://en.wikipedia.org/wiki/Bounding_volume_hierarchy. [Accessed 22 8 2021].

[4] T. Thanh, "Ray Tracing là gì? Tổng hợp thông tin cần biết – Chuyên đề Công nghệ," [Online]. Available: https://vietgame.asia/ray-tracing-la-gi-chuyen-de-cong-nghe/. [Accessed 30 8 2021].

[5] Scratchapixel, "An Overview of the Ray-Tracing Rendering Technique," [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-overview. [Accessed 5 9 2021].

[6] Scratchapixel, "Ray-Tracing: Generating Camera Rays," [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-generating-camera-rays. [Accessed 10 9 2021].

[7] Scratchapixel, "Generating Camera Rays," [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-generating-camera-rays/generating-camera-rays. [Accessed 10 9 2021].

[8] Scratchapixel, "A Minimal Ray-Tracer: Rendering Simple Shapes (Sphere, Cube, Disk, Plane, etc.)," [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes. [Accessed 15 9 2021].

[9] Scratchapixel, "Ray-Sphere Intersection," [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-sphere-intersection. [Accessed 15 9 2021].

[10] Scratchapixel, "Lights," [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/shading-lights. [Accessed 25 9 2021].

[11] Scratchapixel, "Light and Shadows," [Online]. Available: https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/ligth-and-shadows. [Accessed 25 9 2021].

[12] M. G. M. Peter A. Kara, "RAY TRACING FOR HOLOVIZIO LIGHT FIELD DISPLAYS," 2020.

[13] M. F. I. Zhengqing Yun, "Ray Tracing for Radio Propagation Modeling: Principles and Applications," 2015.

[14] T. M. Antti Peuhkurinen, "Real-time Human Eye Resolution Ray Tracing in Mixed Reality," 2021.

PHỤ LỤC