LỜI CẢM ƠN

Trước hết em xin chân thành thầy Trần Ngọc Thái là giáo viên hướng dẫn em trong quá tình thực tập. Thầy đã giúp em rất nhiều và đã cung cấp cho em nhiều tài liệu quan trọng phục vụ cho quá trình tìm hiểu về đề tài "Tìm hiểu về đồ 3D Plug-in API và ứng dụng".

Thứ hai, em xin chân thành cảm ơn các thầy cô trong bộ môn công nghệ thông tin đã chỉ bảo bảo em trong quá trình học và rèn luyện trong 4 năm học vừa qua. Đồng thời em cảm ơn các bạn sinh viên lớp CT1001 đã gắn bó với em trong quá trình rèn luyện tại trường.

Cuối cùng em xin chân thành cảm ơn ban giám hiệu trường Đại Học Dân Lập Hải Phòng đã tạo điều kiện cho em có kiến thức, thư viện của trường là nơi mà sinh viên trong trường có thể thu thập tài liệu trợ giúp cho bài giảng trên lớp. Đồng thời các thầy cô trong trường giảng dạy cho sinh viên kinh nghiệm cuộc sống. Với kiến thức và kinh nghiệm đó sẽ giúp em cho công việc và cuộc sống sau này.

Em xin chân thành cảm ơn!

Hải Phòng, tháng 08 năm 2010

Sinh viên

Nguyễn Hữu Toàn

MỤC LỤC

MỞ ĐẦU

Ngày nay công nghệ thông tin và đặc biệt là các ứng dụng đồ họa 3 chiều ngày càng phát triển mạnh mẽ. Ứng dụng phổ biến nhất của đồ họa 3 chiều chính là Game ,đặc biệt là Game Online-lĩnh vực công nghệ thông tin mang lại nhiều lợi nhuận nhất, ngoài ra là một số các lĩnh vực khác như là y học, xây dựng… Với mong muốn tiếp cận và phát triển lĩnh vực đồ họa 3 chiều để giải quyết một số bài toán trong thực tế, em đã tìm hiểu về thư viện đồ họa O3D Plug-In API.

Đồ án được chia làm 4 chương:

- Chương 1: Đưa ra cái nhìn tổng quan về kỹ thuật đồ họa.
- Chương 2: Tìm hiểu kỹ hơn về một số kỹ thuật ứng dụng trong đồ họa 3D.
- Chương 3: Tìm hiểu tổng quan về về thư viện đồ họa mã nguồn O3D Plug-in API.
- Chương 4: Xây dựng chương trình để giải quyết bài toán đặt ra.

CHƯƠNG 1: TỔNG QUAN VỀ KỸ THUẬT ĐỒ HỌA

1.1 Các khái niệm tổng quan của kỹ thuật đồ họa máy tính

Definition (ISO): Phương pháp và công nghệ chuyển đổi dữ liệu từ thiết bị đồ hoạ sang máy tính.

Computer Graphics là phương tiện đa năng và mạnh nhất của giao tiếp giữa con người và máy tính.

Computer Graphics (Kỹ thuật đồ hoạ máy tính) là một lĩnh vực của Công nghệ thông tin mà ở đó nghiên cứu, xây dựng và tập hợp các công cụ (mô hình lý thuyết và phần mềm) khác nhau: kiến tạo, xây dựng, luu trữ, xử lý các mô hình (model) và hình ảnh (image) của đối tượng. Các mô hình (model) và hình ảnh này có thể là kết quả thu được từ những lĩnh vực khác nhau của rất nhiều ngành khoa học (vật lý, toán học, thiên văn học…)

Computer graphics xử lý tất cả các vấn đề tạo ảnh nhờ máy tính.

1.2 Các kỹ thuật đồ họa

1.2.1 Kỹ thuật đồ họa điểm

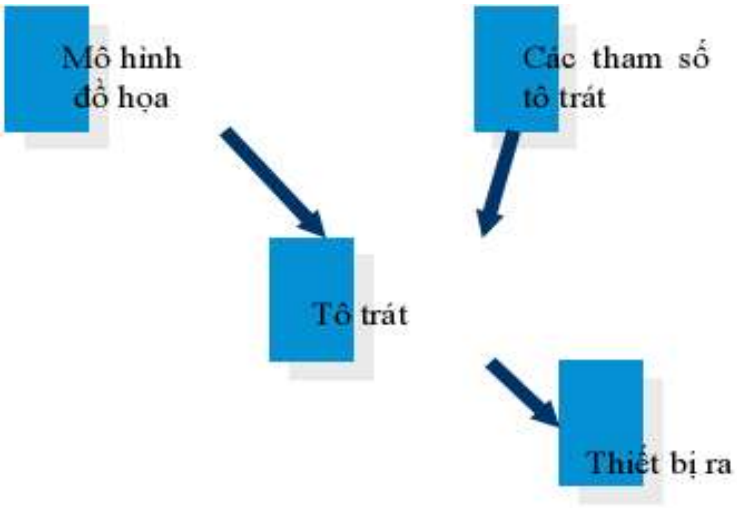
- Các mô hình , hình ảnh của các đối tượng được hiển thị thông qua từng pixel (từng mẫu rời rạc).
- Đặc điểm: có thể thay đổi thuộc tính
  + Xoá đi từng pixel của mô hình và hình ảnh các đối tượng.
  + Các mô hình hình ảnh được hiển thị như một lưới điểm (grid) các pixel rời rạc.
  + Từng pixel đều có vị trí xác định, được hiển thị với một giá trị rời rạc (số nguyên) các thông số hiển thị (màu sắc hoặc độ sáng)
  + Tập hợp tất cả các pixel của grid cho chúng ta mô hinh, hình ảnh đối tượng mà chúng ta muốn hiển thị.

Hình 1.1 Ảnh đồ họa điểm

Phương pháp để tạo ra các pixel

- Phương pháp dùng phần mềm để vẽ trực tiếp từng pixel một.
- Dựa trên các lý thuyết mô phỏng (lý thuyết Fractal, v.v) để xây dựng nên hình ảnh mô phỏng sự vật.
- Phương pháp rời rạc hóa (số hóa) hình ảnh thực của đối tượng.
- Có thể sửa đổi (image editing) hoặc xử lý (image processing) mảng các pixel thu được theo những phương pháp khác nhau để thu được hình ảnh đặc trưng của đối tượng.

1.2.2 Kỹ thuật đồ họa vector



Hình 1.2 Mô hình đồ họa vector

- Mô hình hình học (geometrical model) cho mô hình hoặc hình ảnh của đối tượng.
- Xác định các thuộc tính của mô hình hình học này.

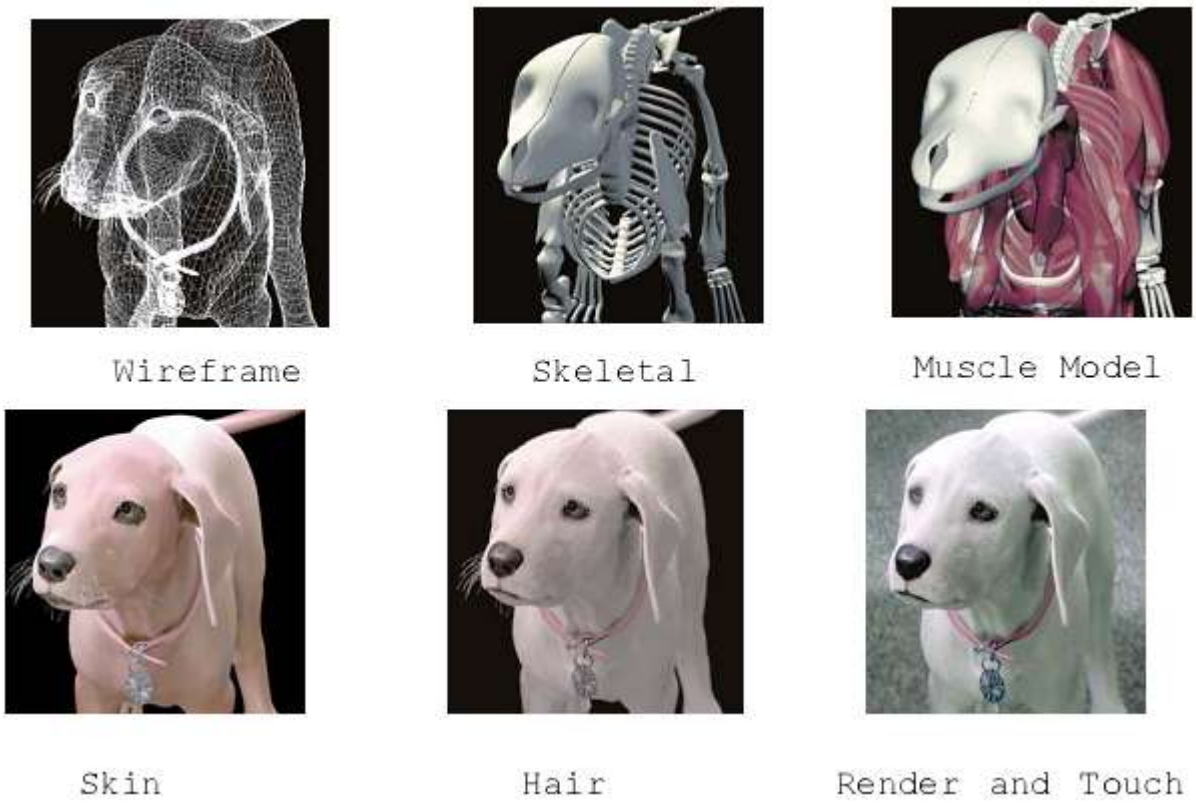
- Quá trình tô trát (rendering) để hiển thị từng điểm của mô hình, hình ảnh thực của đối tượng.

Có thể định nghĩa đồ họa vector: Đồ họa vector = geometrical model + rendering.

So sánh đồ họa điểm và đồ họa vector

Đồ họa điểm(Raster Graphics)	Đồ hoạ vector(Vector Graphics)
Hình ảnh và mô hình của các vật thể được biểu diễn bởi tập hợp các điểm của lưới (grid) Thay đổi thuộc tính của các pixel $\Rightarrow$ thay đổi từng phần và từng cùng của hình ảnh. Copy được các pixel từ một hình ảnh này sang hình ảnh khác.	Không thay đổi thuộc tính của từng điểm trực tiếp Xử lý với từng thành phần hình học cơ sở của nó và thực hiện quá trình tô trát và hiển thị lại. Quan sát hình ảnh và mô hình của hình ảnh và sự vật ở nhiều góc độ khác nhau bằng các thay đổi điểm nhìn và góc nhìn.

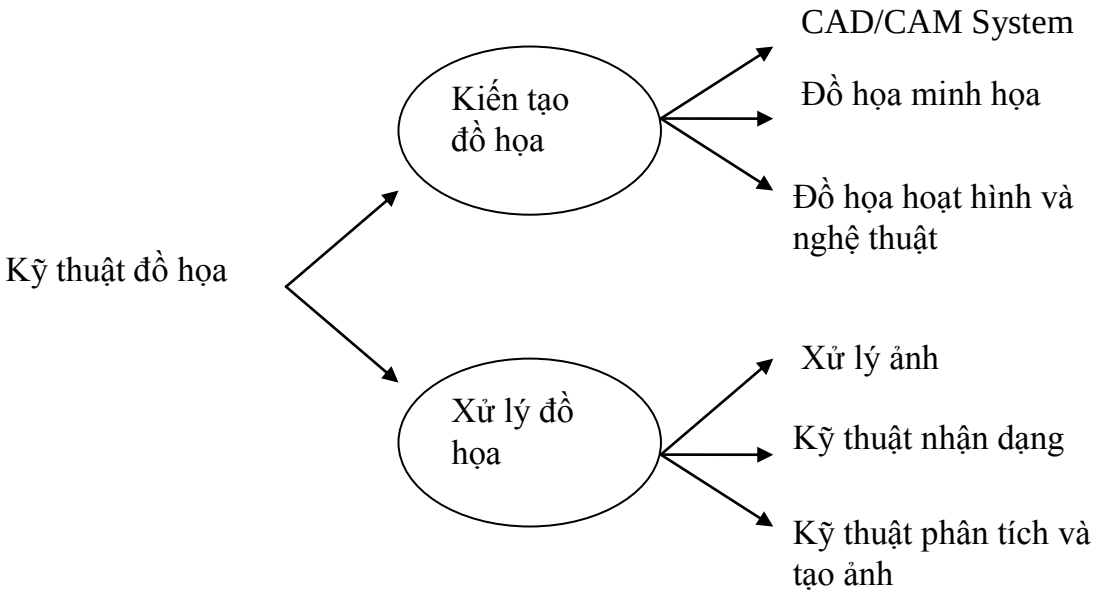
Ví dụ về hình ảnh đồ họa vector



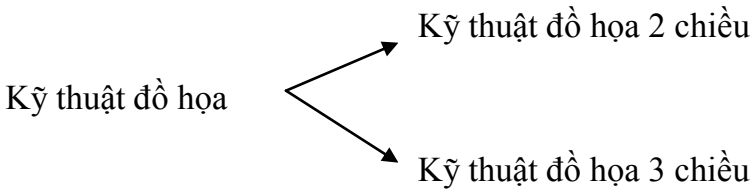
Hình 1.3 Ví dụ về đồ họa vector

1.2.3 Phân loại của đồ họa máy tính

Phân loại theo các lĩnh vực hoạt động của đồ họa máy tính



Phân loại theo hệ tọa độ



- *Kỹ thuật đồ họa 2 chiều*: là kỹ thuật đồ họa máy tính sử dụng hệ tọa độ hai chiều (hệ tọa độ thẳng), sử dụng rất nhiều trong kỹ thuật xử lý bản đồ, đồ thị.

- *Kỹ thuật đồ họa 3 chiều*: là kỹ thuật đồ họa máy tính sử dụng hệ tọa độ ba chiều, đòi hỏi rát nhiều tính toán và phức tạp hơn nhiều so với kỹ thuật đồ họa hai chiều.

Các lĩnh vực của đồ họa máy tính:

- Kỹ thuật xử lý ảnh (*Computer Imaging*): sau quá trình xử lý ảnh cho ta ảnh số của đối tượng, Trong quá trình xử lý ảnh sử dụng rất nhiều các kỹ thuạt phức tạp: kỹ thuật khôi phục ảnh, kỹ thuật làm nối ảnh, kỹ thuật xác định biên ảnh.

- Kỹ thuật nhận dạng (*Pattern Recognition*): từ những ảnh mẫu có sẵn ta phân loại theo các trúc, hoặc theo các tiêu trí được xác định từ trước và bằng các thuật toán chọn lọc để cso thể phân tích hay tổng hợp cá ảnh gốc, các ảnh gốc này được lưu trong một thư viện và căn cứ vào thư viện này ta xây dựng được các thuật giải phân tích và tổ hợp ảnh.

- Kỹ thuật tổng hợp ảnh (*Image Synthesis*): là lĩnh vực xây dựng mô hình và hình ảnh của các vật thể dựa trên các đối tượng và mối quan hệ giữa chúng.

- Các hệ CAD/CAM (Computer Aided Design/Computer Aided Manufacture System): kỹ thuạt đồ họa tập hợp các công cụ, các kỹ thuật trợ giúp cho thiết kế các chi tiết và các hệ thống khác nhau: hệ thống cơ, hệ thống điện, hệ thống điện tử…

- Đồ họa minh họa (Presentation Graphics): gồm các công cụ giúp hiển thị các số liệu thí nghiệm một cách trực quan, dựa trên các mẫ đồ thị hoặc các thuật toán có sẵn.

- Đồ họa hoạt hình và nghệ thuật: bao gồm các công cụ giúp cho các họa sĩ, các nhà thiết kế phim hoạt hình chuyên nghiệp làm các kỹ xảo hoạt hình, vẽ tranh… ví dụ: phần mềm Studio, 3D Animation, 3D Studio Max.

1.2.4 Các ứng dụng tiêu biểu của kỹ thuật đồ họa

Đồ họa máy tính là một trong những linh vực lý thú nhất và phát triển nhanh nhất của tin học. Ngay từ khi xuất hiện nó đã có sức lôi cuốn mãnh liệt, cuốn hút rất nhiều người ở nhiều lĩnh vực khác nhau như khoa học nghệ thuật, kinh doanh, quản lý...Tính hấp dẫn của nó có thể được minh họa rất trực quan thông qua các ứng dụng của nó.

- Xây dựng giao diện người dùng (User Interface):

Giao diện đồ hoạ thực sự là cuộc cách mạng mang lại sự thuận tiện và thoải mái cho người dùng ứng dụng. Giao diện WYSIWYG và WIMP đang được đa số người dùng ưa thích nhừ tính thân thiện, dễ sử dụng của nó.

- Tạo các biểu đồ trong thương mại, khoa học, kỹ thuật

Các ứng dụng này thường được dùng để tóm lược các dữ liệu về tài chính, thống kê, kinh tế, khoa học, toán học… giúp cho nghiên cứu, quản lý… một các có hiệu quả.

- Tự động hóa văn phòng và chế bán điện tử
- Thiết kế với sự trợ giúp của máy tính (CAD_CAM)
- Lĩnh vực giải trí, nghệ thuật và mô phỏng
- Điều khiển các quá trình sản xuất (Process Control)
- Lĩnh vực bản đồ (Cartography)
- Giáo dục và đào tạo

CHƯƠNG 2: MỘT SỐ KỸ THUẬT ỨNG DỤNG TRONG ĐỒ HỌA 3D

2.1 Các phép biến đổi hình học ba chiều

2.1.1 Hệ tọa độ thuần nhất

- Hệ tọa độ thuần nhất: (Homogeneous Coordinates) : Mỗi điểm (x,y,z) trong không gian Descartes được biểu diễn bởi một bộ bốn tọa độ trong không gian 4 chiều thu gọn (hx,hy,hz,h). Người ta thường chọn h=1.

- Các phép biên đổi tuyến tính là tổ hợp của các phép biến đổi sau : tỉ lệ, quay, biến dạng và đối xứng. Các phép biến đổi tuyến tính có các tính chất sau :

  + Gốc tọa độ là điểm bất động
  + Ảnh của đường thẳng là đường thẳng
  + Ảnh của các đường thẳng song song là đường thẳng song song
  + Bảo toàn tỷ lệ khoảng cách
  + Tổ hợp các phép biến đổi có tính phân phối

Ma trận biến đổi tổng quát trong hệ tọa độ thuần nhất (4x4)

$$T = \begin{bmatrix} a & b & c & p \\ d & e & f & q \\ g & i & j & r \\ l & m & n & s \end{bmatrix} \text{ hay } T = \begin{bmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & i & j & 0 \\ dx & dy & dz & 1 \end{bmatrix}$$

2.1.2 Phép tịnh tiến

$$T(dx, dy, dz) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{bmatrix}$$

$$X' = X.T(dx, dy, dz)$$

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}.T(dx, dy, dz) = \begin{bmatrix} x+dx & y+dy & z+dz & 1 \end{bmatrix}$$

2.1.3 Phép tỷ lệ

$$Ts = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X' = X.Ts$$

$$x' \quad y' \quad z' \quad 1 = x \quad y \quad z \quad 1.Ts$$

$$= x.Sx \quad y.Sy \quad z.Sz \quad 1$$

Với Sx, Sy, Sz là các hệ số tỷ lệ trên các trục tọa độ

2.1.4 Phép biến dạng

- Ta có tất cả các phần tử nằm trên đường chéo chính bằng 1
- Các phần tử chiếu và tịnh tiến bằng 0

$$Tsh = \begin{bmatrix} 1 & b & c & 0 \\ d & 1 & f & 0 \\ g & i & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$X' = X.Tsh$$

$$x' \quad y' \quad z' \quad 1 = x \quad y \quad z \quad 1.Tsh$$

$$= x + yd + gz \quad bx + y + iz \quad cx + fy + z \quad 1$$

2.1.5 Phép quay 3 chiều

- Quay quanh trục Oz

$$Tz = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 & 0 \\ -\sin\varphi & \cos\varphi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Quay quanh trục Ox

$$Tx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & \sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Quay quanh trục Oy

$$T_y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.1.6 Phép đối xứng

- Qua mặt phẳng tọa độ

$$M_{Ox} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{Ox} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{Oy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Qua các trục

$$Mx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$My = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Mz = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Qua gốc tọa độ

$$Mo = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

2.2 Quan sát 3 chiều (Phép chiếu - Projection)

2.2.1 Các phép chiếu

Định nghĩa về phép chiếu

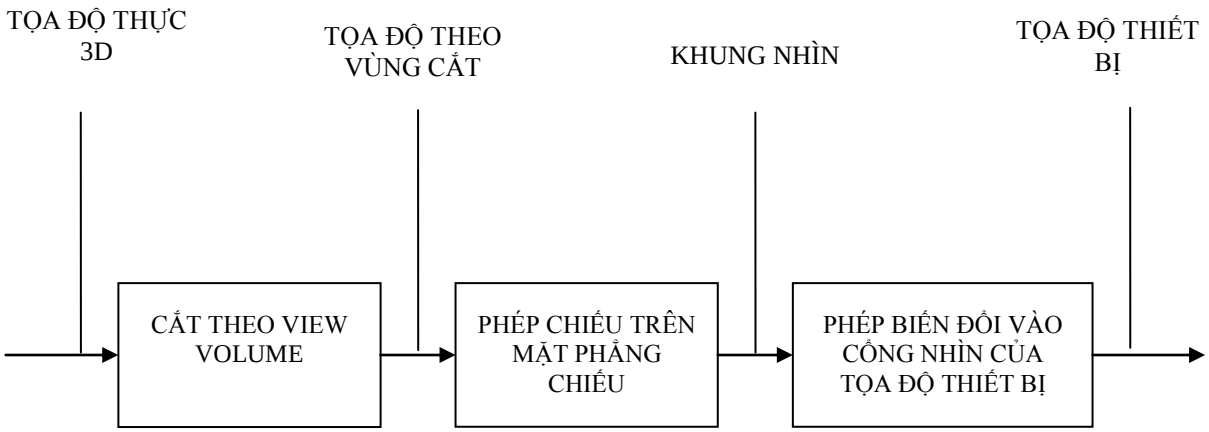
Một cách tổng quát, phép chiếu là phép chuyển đổi những điểm của đối tượng trong hệ thống tọa độ n chiều thành những điểm trong hệ thống tọa độ có số chiều nhỏ hơn n.

Định nghĩa về hình chiếu

Ảnh của đối tượng trên mặt phẳng chiếu được hình thành từ phép chiếu bởi các đường thẳng gọi là tia chiếu (projection) xuất pháp từ một điểm gọi là tâm chiếu (center of projection) đi qua các điểm của đối tượng giao với mặt chiếu (projection plan)

Các bước xây dựng hình chiếu

- Đối tượng trong không gian 3D với tọa độ thực được cắt theo một không gian xác định gọi là view volume.
- View volume dược chiếu lên mặt phẳng chiếu. Diện tích choản bởi view volume trên mặt phẳng chiếu đó sữ cho chúng ta khung nhìn.
- Là việc ánh xạ khung nhìn vào trong một cổng nhìn bất kỳ cho trước trên màn hình để hiển thị hình ảnh.



Hình 2.1 Mô hình nguyên lý của tiến trình biểu diễn đối tượng 3D

2.2.1.1 Phép chiếu song song (Parallel Projections)

Phép chiếu song song (Parallel Projections) là phép chiếu mà ở đó các tia chiếu song song với nhau hay xuất phát từ điểm vô cùng.

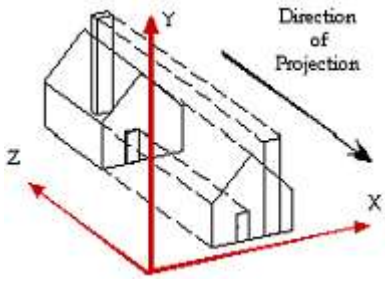
Phân loại phép chiếu song song dựa trên hướng của tia chiếu (Direction Of Projection) và mặt phẳng chiếu (projection plane).

2.2.1.1.1 Phép chiếu trực giao (Orthographic projection)

Là phép chiếu song song và tia chiếu vuông góc với mặt phẳng chiếu. Về mặt toán học, phép chiếu trực giao là phép chiếu với một trong các mặt phẳng tọa độ có giá trị bằng 0. Thường dùng mặt phẳng z=0, ngoài ra x=0 và y=0.

Ứng với mỗi mặt phẳng chiếu ta có một ma trận chiếu tương ứng.

$$\underline{Ty} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \underline{Tx} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \underline{Tz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Hình 2.2 Phép chiếu trực giao

Thông thường thì người ta không sử dụng cả 6 mặt phẳng để suy diễn ngược hình của một đối tượng mà chỉ sử dụng một trong số chúng như: hình chiếu bằng, đứng, cạnh.

Cả sáu góc nhìn đều có thể thu được từ một mặt phẳng chiếu thông qua các phép biến đổi hình học như quay, dịch chuyển hay lấy đối xứng.

Ví dụ: giả sử chúng ta có hình chiếu bóng trên mặt phẳng z=0, với phép quay đối tượng quanh trục một góc 90 sẽ cho ta hình chiếu cạnh.

Đối với các đối tượng mà các mặt của chúng không song song với một trong các mặt phẳng hệ tọa độ thì phép chiếu này không cho hình dạng thật của vật thể. Muốn nhìn vật thể chính xác hơn người ta phải hình thành phép chiếu thông qua viếc quay và dịch chuyển đối tượng sao cho mặt phẳng đó song song với các trục toạ độ.

Hình của đối tượng quá phức tạp cần thiết phải biết các phần bên trong của đối tượng đôi lúc chúng ta phải tạo mặt cắt đối tượng.

2.2.1.1.2 Phép chiếu trục lượng (Axonometric)

Phép chiếu trục lượng là phép chiếu mà hình chiếu thu được sau khi quay đối tượng sao cho ba mặt của đối tượng được trông thấy rõ nhất (thường mặt phảng chiếu là z=0).

Có 3 phép chiếu

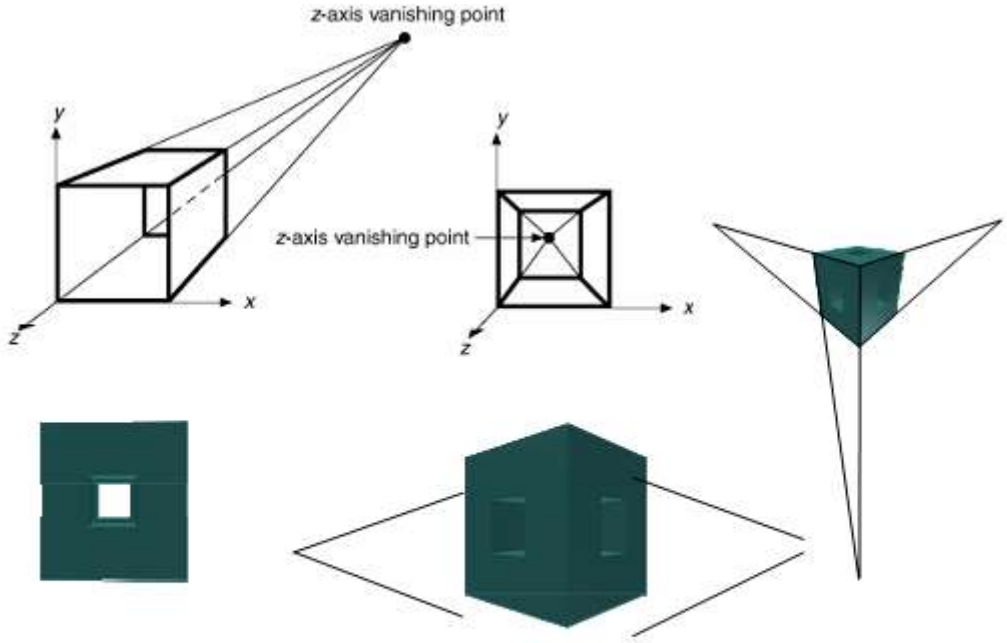
- Phép chiếu Trimetric
- Phép chiếu Dimetic
- Phép chiếu Isometric

2.2.1.2 Phép chiếu phối cảnh (Perspective Projection)

Phép chiếu phối cảnh là phép chiếu mà các tia chiếu không song song với nhau mà xuất phát từ một điểm gọi là tâm chiếu. Phép chiếu phối cảnh tạo ra hiệu ứng về luật xa gần tạo cảm giác về độ sâu của đối tượng trong thế giới thật mà phép chiếu song song không lột tả được.

Các đoạn thẳng song song của mô hình 3D sau phép chiếu hội tụ tại một điểm gọi là điểm triệt tiêu (vanishing point).

Phân loại phép chiếu phối cảnh dựa vào tâm chiếu - *Centre Of Projection* (COP) và mặt phẳng chiếu - projection plane



Một tâm chiếu	Hai tâm chiếu	Ba tâm chiếu
1 point Perspective	2 point Perspective	3 point Perspective

Hình 2.3 Phép biến đổi phối cảnh

2.2.1.2.1 Phép chiếu phối cảnh một tâm chiếu

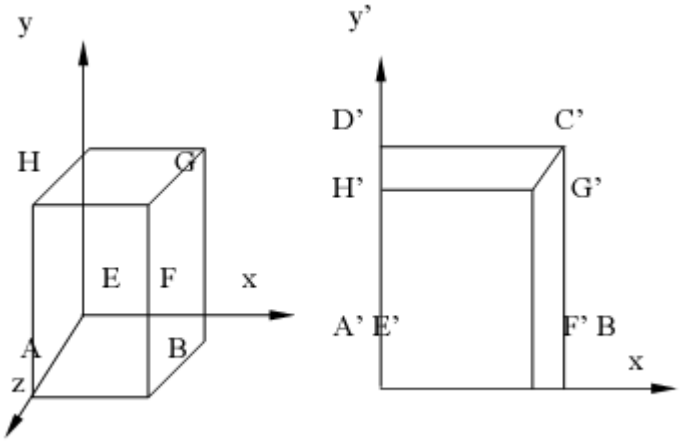
Giả sử khi mặt phẳng được đặt tại z = 0 và tâm phép chiếu nằm trên trục z, cách trục z một khoảng zc = -1/r.

Nếu đối tượng cũng nằm trên mặt phẳng z = 0 thì đối tượng sẽ cho hình ảnh thật.

Phương trình biến đổi:

[x y z 1][Tr] = [x y z rz+1]

Ma trận biến dổi một điểm phối cảnh [Tr] có dạng:



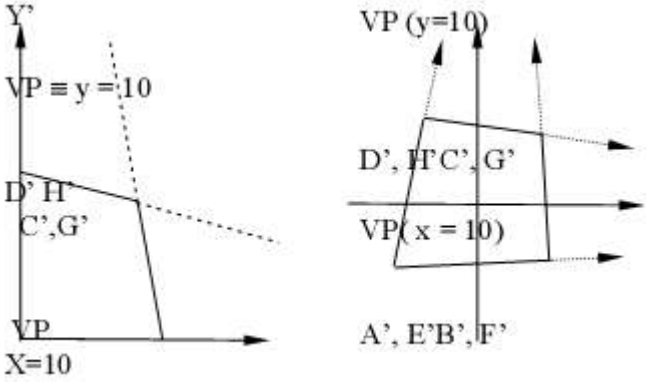
Hình 2.4 Phép chiếu phối cảnh một tâm chiếu

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{vmatrix} x & y & z & 1 \end{vmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{vmatrix} x & y & 0 & rz+1 \end{vmatrix}$$

$$\begin{vmatrix} x' & y' & z' & 1 \end{vmatrix} = \begin{bmatrix} \frac{x}{rz+1} & \frac{y}{rz+1} & 0 & 1 \end{bmatrix}$$

2.2.1.2.2 Phép chiếu phối cảnh hai tâm chiếu



Hình 2.5 phép chiếu phối cảnh hai tâm chiếu

$$T_c = T_{pq} \cdot T_z$$

$$= \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{pq} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

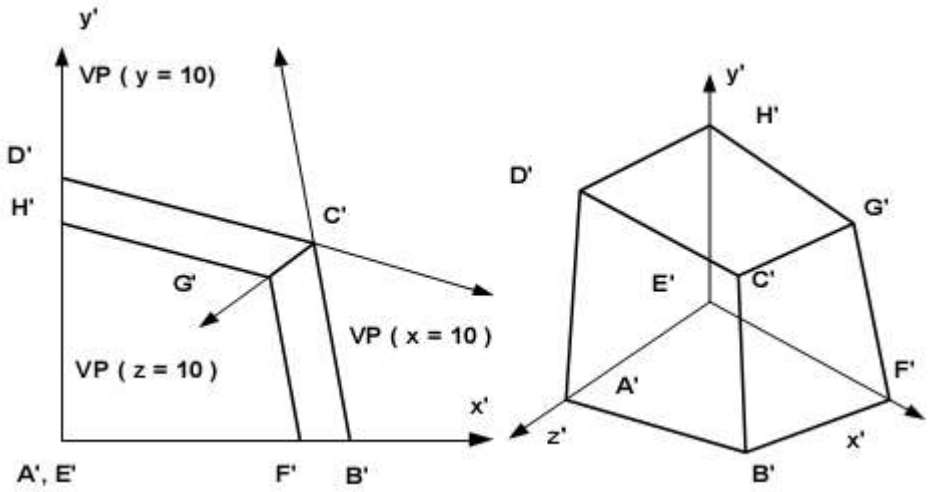
$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & px+qy+1 \end{bmatrix}$$

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} \frac{x}{px+qy+1} & \frac{y}{px+qy+1} & \frac{z}{px+qy+1} & 1 \end{bmatrix}$$

Hai tâm chiếu: [-1/p 0 0 1] và [0 -1/q 0 1]

Điểm triêu tiêu (VP – Vanishing point) tương ứng trên 2 trục x và y là điểm: [1/p 0 0 1] và [0 1/q 0 1].

2.2.1.2.3 Phép chiếu phối cảnh ba tâm chiếu



Hình 2.6 Phép chiếu phối cảnh ba tâm chiếu

$$T_{pqr} = T_p \cdot T_q \cdot T_r$$

$$= \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_c = T_{pqr} \cdot T_z = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 0 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & (px+qy+rz+1) \end{bmatrix}$$

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} \frac{x}{(px+qy+rz+1)} & \frac{y}{(px+qy+rz+1)} & \frac{z}{(px+qy+rz+1)} & 1 \end{bmatrix}$$

Ba tâm chiếu: trên trục x tại điểm [-1/p 0 0 1], y tại điểm [0 -1/q 0 1] và z tại điểm [0 0 -1/r 1]

Điểm triệt tiêu – VP sẽ tương ứng với các giá trị:

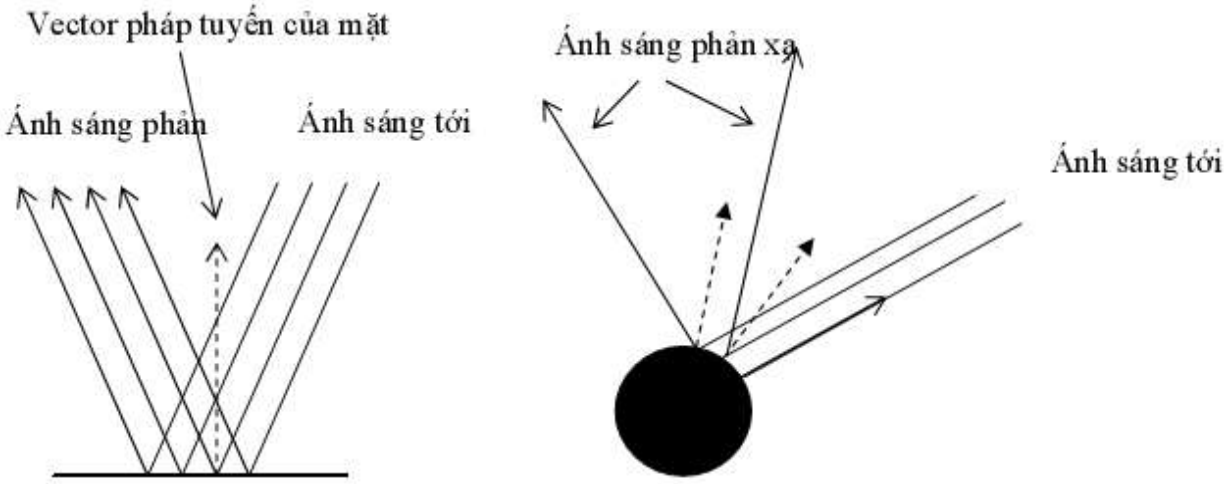
[1/p 0 0 1], [0 1/q 0 1] [0 0 1/r 1]

2.2.2 Chiếu sáng và tô bóng

Khi biểu diễn các đối tượng 3 chiều, một yếu tố không thể bỏ qua để tăng tính thực của đối tượng đó là tạo bóng sáng cho vật thể. Để thừjc hiện được điều này, chúng ta cần phải lần lượt tìm hiểu các dạng nguồn sáng có trong tự nhiên, cũng như các tính chất đặc trưng khác nhau của mỗi loại nguồn sáng.

2.2.2.1Nguồn sáng xung quanh

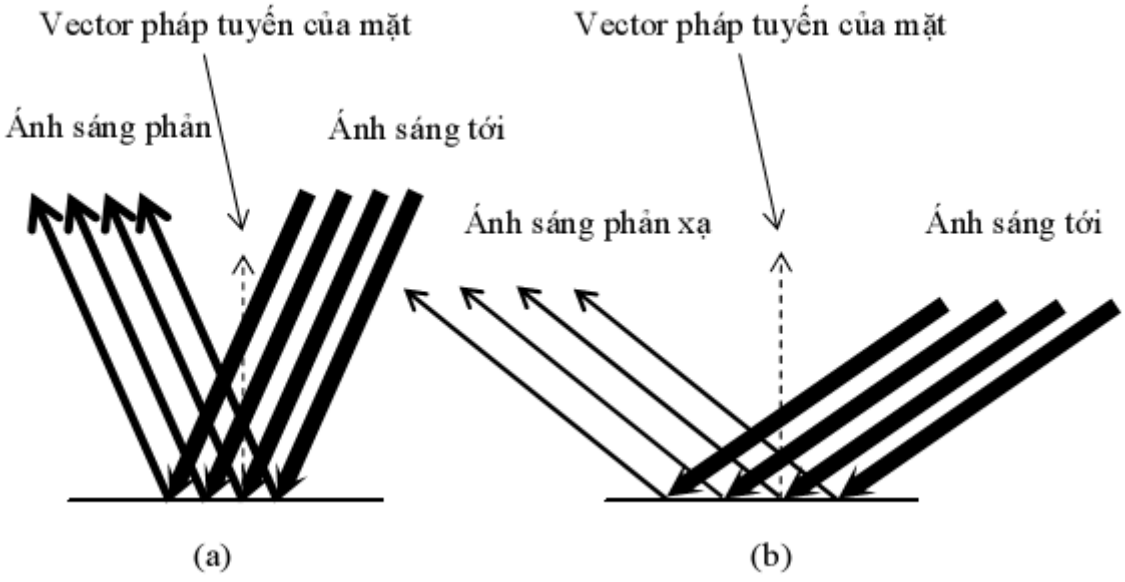
Ánh sáng xung quanh là mức ánh sáng trung bình, tồn tại trong một vùng khôn gian. Một không gian lý tưởng là khôn gian mà tại đó mọi vật đều được cung cấp một lượng ánh sáng lên bề mặt là như nhau, từ mọi phía ở mọi nơi. Thông thường ánh sáng xung quanh được xác định với một mức cụ thể gọi là mức sáng xung quanh của vùng không gian mà vật thể đó cự ngụ, sau đó ta cộng với cường độ sáng có được từ các nguồn sáng khác để có được cường độ sáng cuối cùng lên một điểm hay một mặt của vật thể.



Hình 2.7. Sự phản xạ của ánh sáng

2.2.2.2Nguốn sáng định hướng

Nguồn sáng định hướng giống như những gì mà mặt trời cung cấp cho chúng ta. Nó bao gồm một tập các tia sáng song song, bất kể cường độ của chúng có giống nhau hay không. Có hai loại kết quả của ánh sáng định hướng khi chúng ta chiêu đến bề mặt là: khuếch tán và phản chiếu. Nếu bề mặt phản xạ toàn bộ (giống như trong gương) thì các tia phản xạ sẽ có hướng ngược với hướng của góc tới. Trong trường hợp ngược lại, nếu bề mặt là không phản xạ toàn phần (có độ xám, xù xì) thì một phần các tia sáng sẽ bị tỏa đi các hướng khác hay bị hấp thụ, phần còn lại thì phản xạ lại, và lượng ánh sáng phản xạ lại này tỷ lệ với góc tới. Ở đây chúng ta sẽ quan tâm đến hiện tượng phản xạ không toàn phần vì đây là hiện tượng phổ biến.



Hình 2.8. Sự phản xạ không toàn phần của ánh sáng

Trong hình 2.8 thể hiện sự phản xạ ánh sáng không toàn phần. Độ đậm nét của các tia ánh sáng tới thể thiện cường độ sáng cao, độ mảnh của các tia phản xạ thể hiện cường độ sáng thấp. Nói chung, khi bề mặt là không phản xạ toàn phần thì cường độ của ánh sáng phản xạ luôn bé hơn so với cường độ của ánh sáng tới, và cường độ của tia phản xạ còn tỷ lệ với góc giữa tia tới với vector pháp tuyến của bề mặt, nếu góc này càng nhỏ thì cường độ phản xạ càng cao. Ở đây ta chỉ quan tâm đến thành phần ánh sáng khuếch tán và tạm bỏ qua hiện tượng phản xạ toàn phần.

Nếu gọi θ là góc giữa tia tới với vector pháp tuyến của bề mặt thì Cos(θ) phụ thuộc vào tia tới a và vector pháp tuyến của mặt n theo công thức:

$$Cos\,\theta = \frac{\vec{a}.\vec{n}}{|\vec{a}|.|\vec{n}|} \qquad (*)$$

Trong công thức trên Cos(θ) bằng tích vô hướng của a và n chia cho tích độ lớn của chúng. Nếu ta đã chuẩn hóa độ lớn của các vector a và n về 1 từ trước thì ta có thể tính giá trị trên một cách nhanh chóng như sau:

Cos(θ) = tích vô hướng của $\vec{a}$ và $\vec{n} = a.x * n.x + a.y * n.y + a.z * n.z$

Vì Cos(θ) có giá trị từ +1 đến -1 nên ta có thể suy ra công thức tính cường độ của ánh sáng phản xạ là:

Cường độ AS phản xạ = Cường độ AS định hướng * $|Cos\,\theta + 1/2|$ (**)

Trong đó $|Cos\,\theta + 1/2|$ có giá trị trong khoảng từ 0 đến 1. Vậy qua công thức (*) và (**) chúng ta có thể tính được cường độ của ánh sáng phản xạ trên bề mặt khi biết được cường độ của ánh sáng định hướng cũng như các vector pháp tuyến của mặt và tia tới.

2.2.2.3Nguồn sáng điểm

Nguồn sáng định hướng là tương đương với nguồn sáng điểm đặt ở vô tận. Nhưng khi nguồn sáng điểm được mang đến gần đối tượng thì các tia sáng từ nó phát ra không còn song song nữa mà được tỏa ra theo mọi hướng theo dạnh hình cầu. Vì thế, các tia sáng sẽ rơi xuống các điểm trên bề mặt dưới các góc khác nhau. Giả sử vector pháp tuyến của mặt là n = (xn, yn, zn), điểm đang xét có tọa độ là

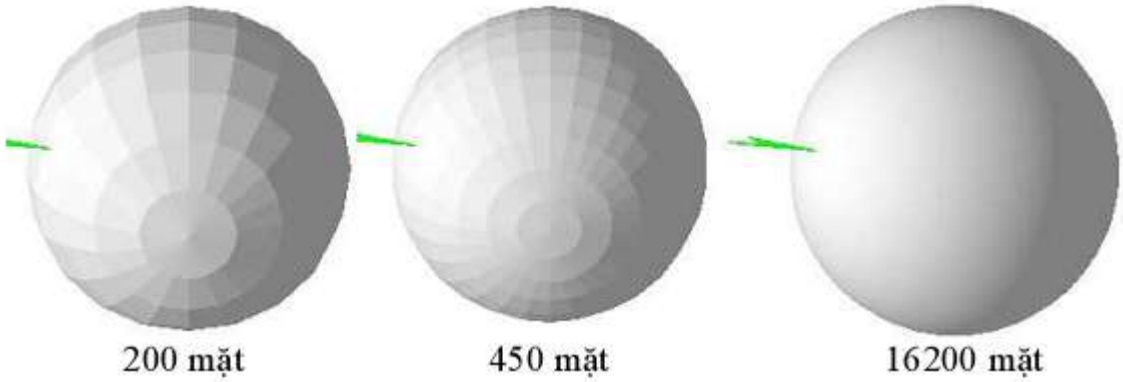
(x_0, y_0, z_0) và nguồn sáng điểm có tọa độ là (plx, ply, plz) thì ánh sáng sẽ rọi đến điểm đang xét theo vector $(x_0 - plx, y_0 - ply, z_0 - plz)$ hay tia tới:

$$a = (plx - x_0, ply - y_0, plz - z_0)$$

Từ đó cường độ sáng tại điểm đang xét sẽ phụ thuộc vào Cos(θ) giữa n và a như đã trình bày trong nguồn sáng định hướng.

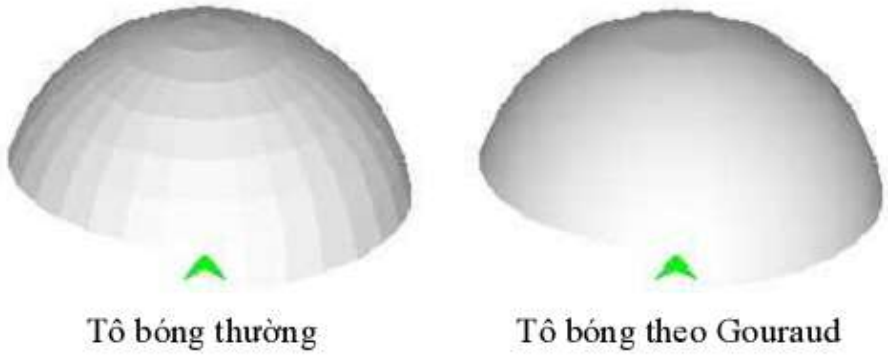
2.2.2.4Mô hình bóng Gouraud

Mô hình bóng Gouraud là một phương pháp vẽ bóng, tạo cho đối tượng 3D có hình dáng cong có một cái nhìn có tình thực hơn. Phương pháp này đặt cơ sở trên thực tế sau: đối với các đối tượng 3D có bề mặt cong thì người ta thường xấp sỉ bề mặt cong của đối tượng bằng nhiều mặt đa giác phẳng, ví dụ như một mặt cầu cso thể xấp sỉ bởi một tập các mặt đa giác phẳng có kích thước nhở sắp xếp lại, khi số đa giác xấp xỉ tăng lên thì tính thực của mặt cầu sẽ tăng, sẽ cho ta cảm giác mặt cầu trông tròn trịa hơn, mịn và cong hơn. Tuy nhiên, khi số đa giác xấp xỉ một mặt cong tăng thì khối lượng tính toán và lưu trữ cũng tăng theo tỷ lệ thuận theo số mặt, điều đó dẫn đến tốc độ thực hiện sẽ trở nên chậm chạp hơn. Vấn đề thứ 2 nảy sinh là khi ta phóng lớn hay thu nhỏ vật thể. Nếu ta phóng lớn thì rõ ràng các đa giác cũng được phóng lớn theo cùng tỷ lệ, dẫn đến hình ảnh về các mặt đa giác lại hiện rõ và gây ra cảm giác không được trơn mịn. Ngược lại, khi ta thu nhỏ thì nếu số đa giác xấp xỉ lớn thì sẽ dẫn đến tình trạng các đa giác nhỏ, chồng chất lên nhau không cần thiết.



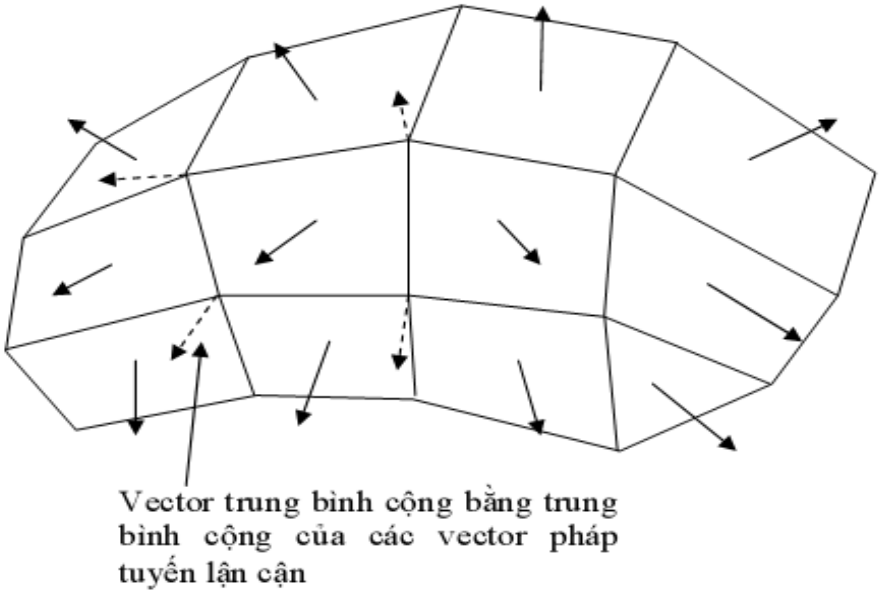
Hình 2.9. So sánh vật thể với số mặt đa giác tăng dần

Để giải quyết vấn đề trên, chúng ta có thể tiến hành theo phương pháp tô bóng Gouraud. Mô hình bóng Gouraud tạo cho đối tượng một cái nhìn giống như là nó có nhiều mặt đa giác bằng cách vẽ mỗi mặt không chỉ với một cường độ sáng mà vẽ với nhiều cường độ sáng khác nhau trên các vùng khác nhau, làm cho mặt phẳng nơm như bị cong. Bởi thực chất ta cảm nhận được độ cong của các mặt cong do hiệu ứng ánh sáng khi chiếu lên mặt, tại các điểm trên mặt cong sẽ có vector pháp tuyến khác nhau nên sẽ đón nhận và phản xạ ánh sáng khác nhau, từ đó ta sẽ cảm nhận được các độ sáng khác nhau trên cùng một mặt cong.



Hình 2.10. So sáng tô bóng thường và tô bóng Gouraud

Thường thì mỗi mặt đa giác có một vector pháp tuyến, và như phần trên đã trình bày, vector pháp tuyến đó được dùng để tính cường độ của ánh sáng phản xạ trên bề mặt của đa giác từ đó suy ra cường độ sáng của mặt. Tuy nhiên mô hình Gouraud lại xem một đa giác không chỉ có một vector pháp tuyến, mà mỗi đỉnh của mặt đa giác lại có một vector pháp tuyến khác nhau, và từ vactor pháp tuyến của các đỉnh chúng ta sẽ nội suy ra được vector pháp tuyến của từng điểm trên mặt đa giác, từ đó tính được cường độ sáng của điểm. Như thế, các điểm trên cùng một mặt của đa giác sẽ có cường độ sáng khác nhau và cho ta cảm giác mặt đa giác không phải là mặt phẳng mà là mặt cong.



Hình 2.11. Mô tả vector trung bình cộng của các mặt

CHƯƠNG 3: GIỚI THIỆU VỀ 03D PLUG-IN API

3.1 Giới thiệu tổng quan về O3D Plug-In

3.1.1 Một số khái niệm và đặc điểm về O3D

O3D là một mã nguồn mở JavaScript API cho việc tạo đồ họa 3D tương tác các ứng dụng chạy trong một cửa sổ trình duyệt, trò chơi, quảng cáo, người xem mô hình 3D, trình diễn sản phẩm, thế giới ảo. O3D mở rộng phần mềm client-side của một ứng dụng web bằng cách cung cấp tính năng theo các mức sau:

+ *Hệ thống:* O3D cung cấp một plug-in trình duyệt thể hiện khả năng đồ họa bên trong trình duyệt web tiêu chuẩn trên Windows, Macintosh, và Linux (TBP) nền tảng.

+ *Nội dung:* Nội dung của web cho ngày hôm nay là ở dạng HTML, tập tin hình ảnh, và các tập tin video. Các phát triển cung cấp thông tin về cách để tạo ra một công cụ chuyển đổi tập tin và bộ nạp cho bất kỳ nội dung 3D. O3D cung cấp một mẫu COLLADA Converter, có thể được sử dụng để nhập khẩu các tập tin ở định dạng COLLADA, một tiêu chuẩn mở đối với tài sản 3D được hỗ trợ bởi các ứng dụng tạo ra phổ biến nội dung như SketchUp, 3ds Max, và Maya. Sử dụng mẫu chuyển đổi trực tiếp, hoặc viết công cụ chuyển đổi của chính ứng dụng và bộ nạp cho các định dạng khác.

+ *Mã số:* O3D mở rộng ứng dụng mã JavaScript với một API cho đồ họa 3D.Nó sử dụng tiêu chuẩn chế biến sự kiện JavaScript và các phương pháp gọi lại.

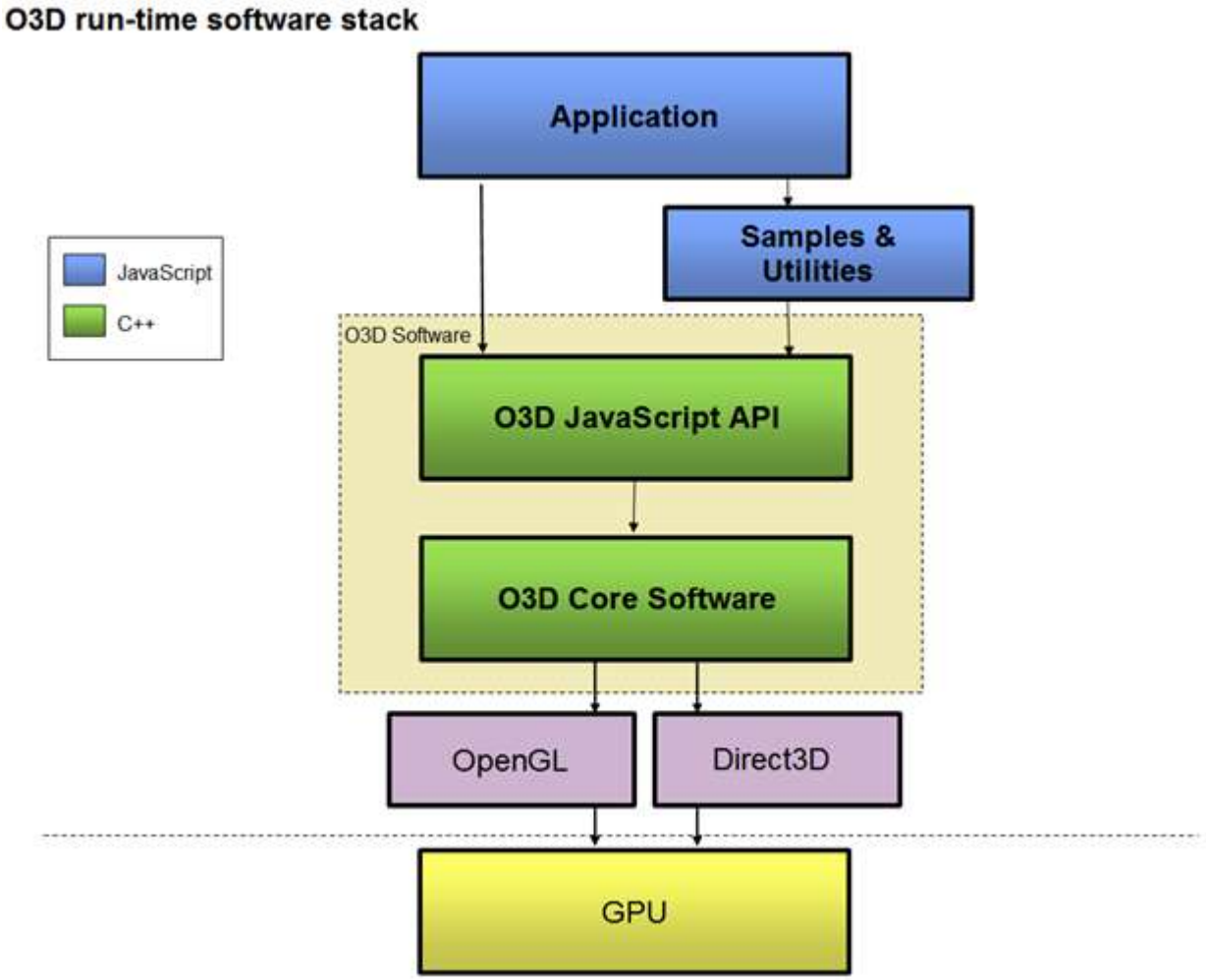
+ Mạng.

+ Nguồn vào (mô hình đồ họa, các thực thể, thông tin giữa nhiều người chơi…v.v.).

+ Va chạm (chuyên xử lý về vật lý trong game).

3.1.2 Cấu trúc quản lý O3D Plugin

Hình dưới đây cho thấy một cái nhìn đơn giản của phần mềm O3D stack:



Các thành phần chính của phần mềm này như sau:

+ O3D ứng dụng JavaScript.

+ Tiện ích JavaScript được cung cấp như mẫu mã với nhiệm vụ lập trình chung.

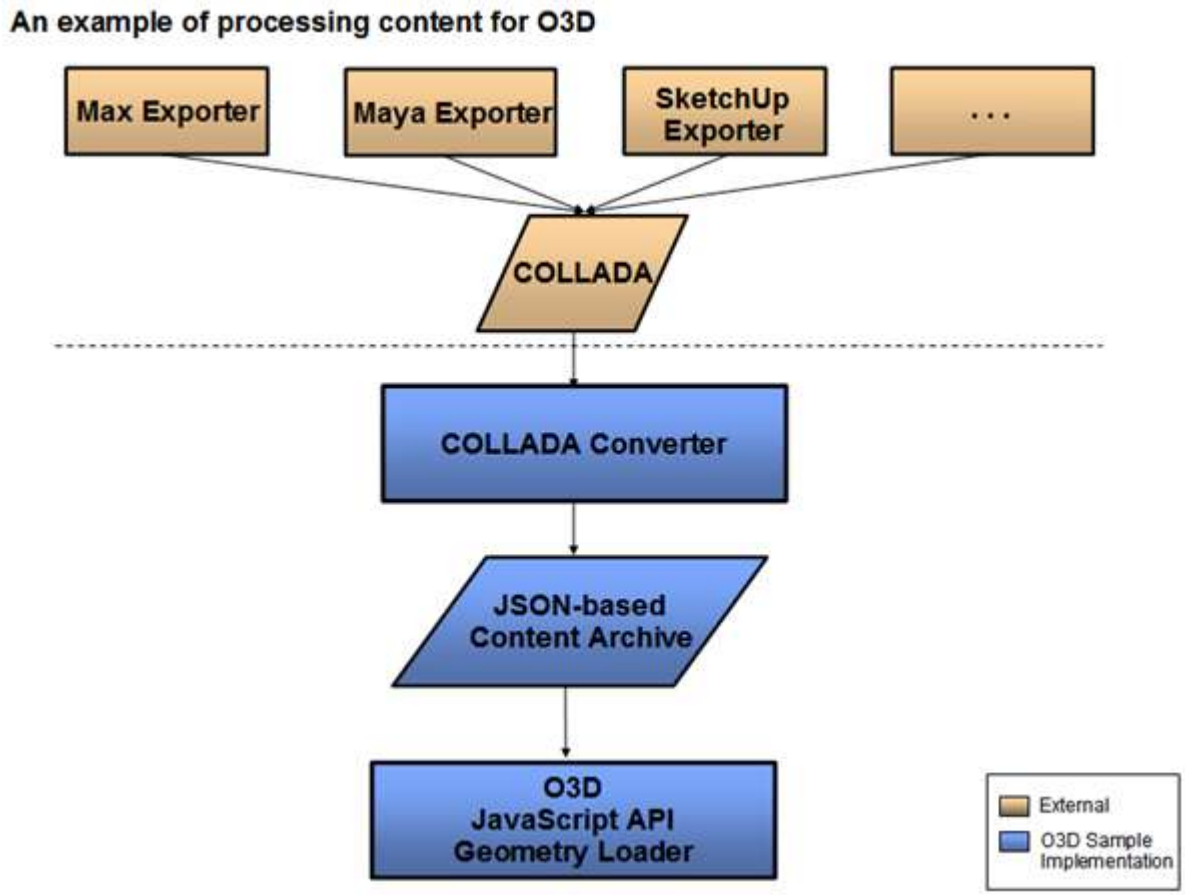
+ API, trong đó có các lớp học và chức năng sử dụng trong các ứng dụng của bạn. Điều này mã nguồn, viết bằng C + +, là mã nguồn mở và có thể được xem trong khu vực tải của dự án O3D .

Các mã JavaScript O3D ứng dụng là hoàn toàn có trong một tài liệu HTML đó là nạp vào một trình duyệt web. Để phát triển một ứng dụng O3D, chỉ cần có các O3D plug-in và soạn thảo một văn bản để viết mã JavaScript.

Theo kiến trúc cho thấy sơ đồ, các O3D giao tiếp phần mềm với phần cứng đồ họa của hệ thống (đơn vị của *nó-GPU* xử lý đồ họa) hoặc thông qua các thư viện OpenGL hoặc Direct3D.

3.2 Nội dung nhập khẩu

Thư viện O3D cung cấp bản vẽ nguyên thủy để tạo ra hình dạng trực tiếp trong ứng dụng (danh sách điểm, danh sách các dòng, danh sách tam giác, dải hình tam giác, hình tam giác fan hâm mộ). Ví dụ, O3D cung cấp mã mẫu để hiển thị như thế nào, có thể nhập nội dung từ một tập tin COLLADA, nhập khẩu nội dung từ các ứng dụng sáng tạo nội dung như Autodesk 3ds Max, Maya, và Google SketchUp, như thể hiện trong hình này:



Như thể hiện trong sơ đồ trên, "nguyên liệu" COLLADA tập tin xuất khẩu từ 3ds Max, Maya, và SketchUp được chuyển đổi bởi các COLLADA mẫu Converter (hộp màu xanh) để sử dụng bởi các JavaScript O3D API.

3.3 Các đồ thị cảnh API là gì?

Quang cảnh O3D đồ thị API được sử dụng để tạo ra một biến đổi đồ thị và biểu đồ vẽ lại. Việc chuyển đổi các cửa hàng đồ thị các thông tin về vị trí, kích thước, hình dạng, vật liệu, và shaders mà bao gồm các dữ liệu cơ bản về ứng dụng 3D trên thế giới. Đồ thị vẽ lại cửa hàng thông tin về cách thức các đối tượng 3D được chuyển thành các điểm ảnh thực tế được hiển thị trên màn hình của những người sử dụng. Đồ thị vẽ có trách nhiệm sau đây:

- Chứa thông tin về những hình dạng 3D là không thấy được.
- Thông qua các biến đổi đồ thị để lắp ráp nguyên thủy được rút ra.
- Xử lý các tính toán liên quan cho các hiệu ứng vẽ đặc biệt như tính minh bạch, nhiều quan điểm giống nhau của thế giới, và hiển thị heads-up.

3.3.1 Chuyển đồ thị

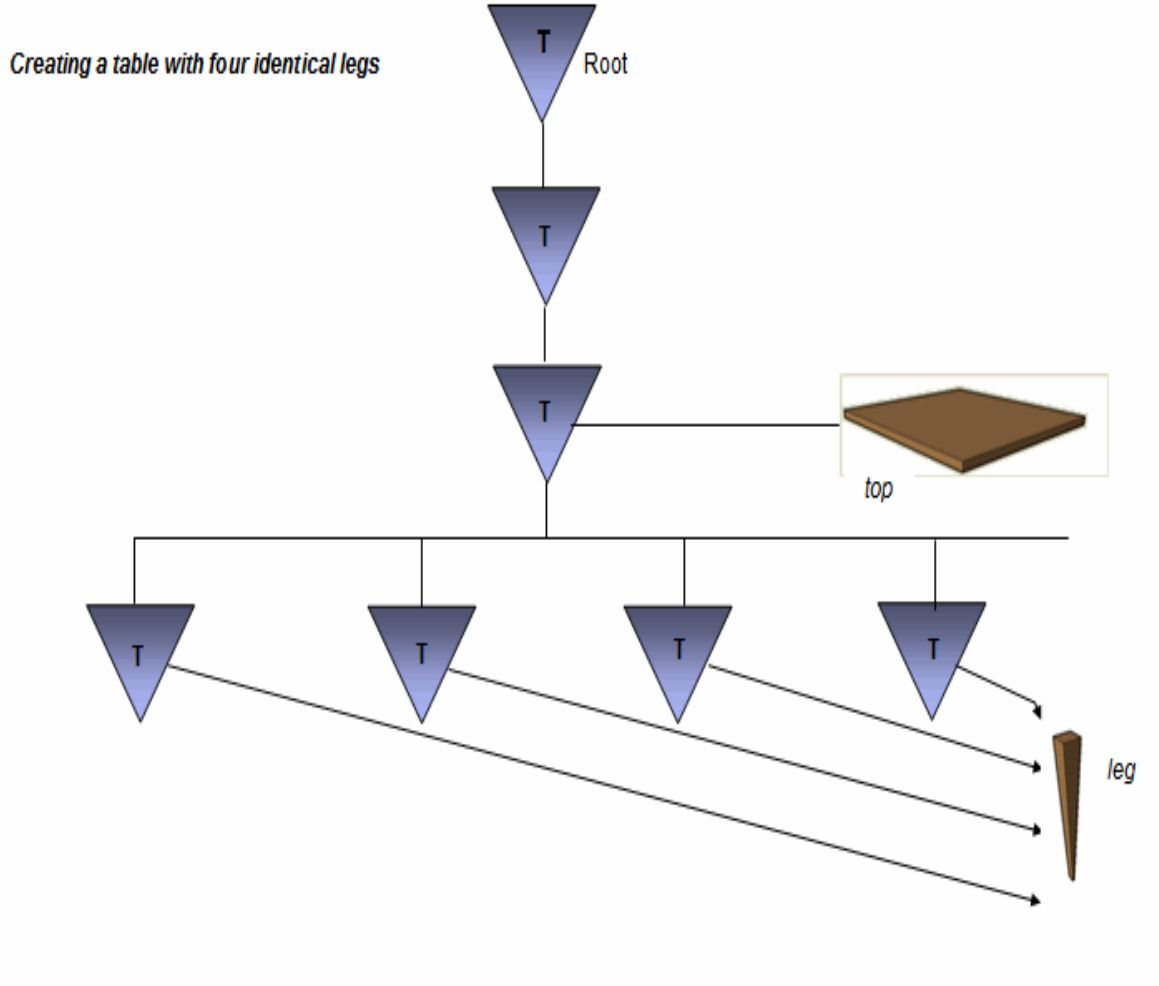
Một biến đổi chứa một ma trận có quy định cụ thể như thế nào liên quan đến hình dạng và kích cỡ được định vị trong không gian 3D. Một biến đổi đồ thị là một bộ sưu tập đã ra lệnh các biến đổi đó được sắp xếp theo một thứ bậc cha mẹ trẻ em . Các ứng dụng của biến đổi đồ thị có một gốc biến đổi ở trên cùng của cây và bất kỳ số nào của con biến đổi bố trí trong các ngành dưới gốc biến đổi.

Biến đổi có tác dụng tích lũy, với biến đổi quy định cao hơn trong cây áp dụng cho lớp con các biến đổi về chi nhánh dưới của biến đổi đồ thị.

3.3.2 Shapes

Một biến có thể có một hoặc nhiều hình dạng liên kết với nó. Một định nghĩa một hình dạng mảnh hình học đó là vị trí và kích thước như một đơn vị. Một hình dạng, lần lượt, bao gồm các tài nguyên, mỗi trong số đó có thể có một loại vật liệu khác nhau được giao. Hình dạng được định nghĩa một cách độc lập và sau đó liên kết với một biến đổi hình dạng mà các vị trí riêng của mình tại địa phương phối hợp không gian. . Ví dụ, nếu tạo ra một bảng với bốn chân giống hệt nhau, ta sẽ mô hình hình dạng chân và sau đó tham chiếu nó bốn lần trong bốn biến đổi đó chỉ định các vị trí cho bàn chân bốn. Sau đó, tạo ra đầu bảng và tham chiếu nó trong một biến đổi mà đặt nó trên đầu trang của bốn chân của nó. Cuối cùng, tạo ra một "phụ

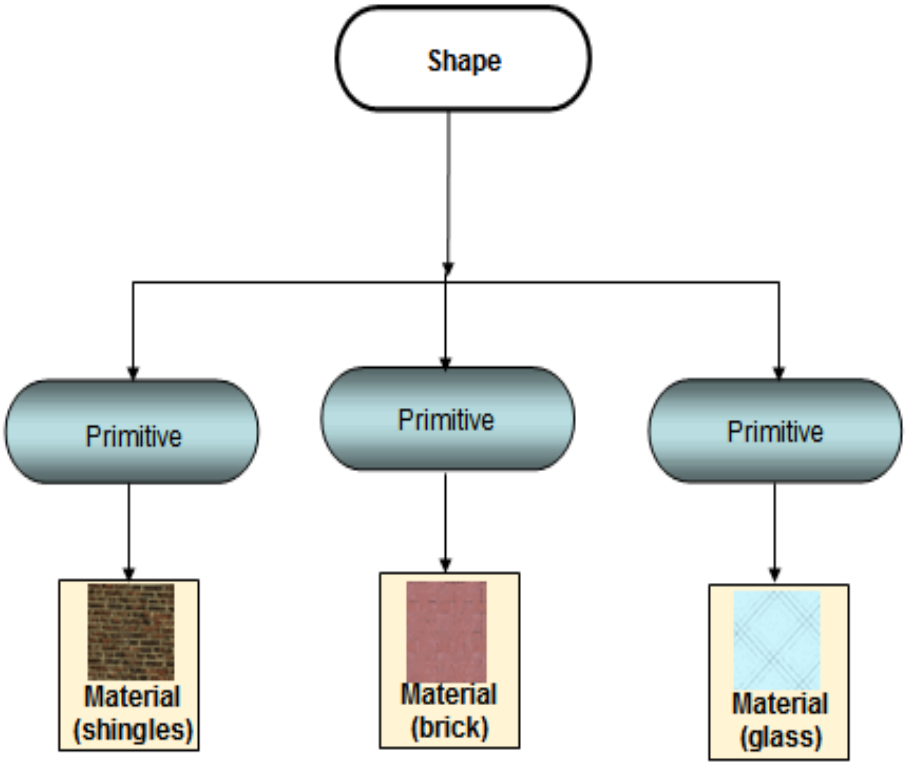
huynh" chuyển cho các đơn vị bảng để di chuyển toàn bộ nhóm đến vị trí mong muốn của mình.Việc chuyển đổi cơ bản cho đồ thị biến đổi và hình dạng tạo nên bảng này sẽ như sau:



3.3.3 Materials

Mỗi tài nguyên có chứa một tham số vật liệu. vật liệu có thể được chia sẻ giữa nhiều tài nguyên. Nếu sử dụng các mô hình từ một ứng dụng bên ngoài mô hình 3D, tài nguyên được tạo ra tự động khi cần thiết cho các vật liệu khác nhau.

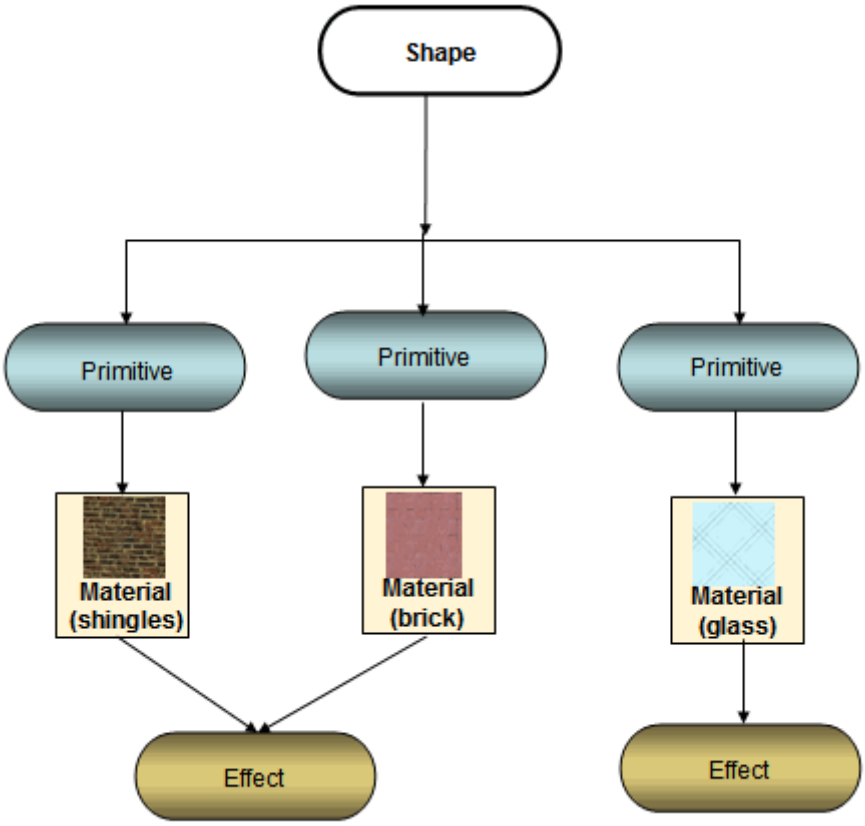
Một tai nguyên có chứa một tham số vật liệu



3.3.4 Hiệu ứng

Một vật liệu có chứa một tham số cho hiệu ứng và tham số tuỳ chọn. Hiệu ứng này, lần lượt, có chứa một đoạn đổ bóng đỉnh và một (pixel) đổ bóng, mà cùng nhau xác định làm thế nào để các điểm ảnh màu sắc tạo nên hình dạng. Các thông số của vật liệu, chẳng hạn như màu khuếch tán của nó, màu sắc specular, màu sắc xung quanh, được sử dụng bởi các hiệu ứng nó đề cập đến.

Một vật liệu có chứa một tham số cho hiệu ứng



3.4 Tạo chuyển đồ thị

Với O3D có thể tạo ra những biến đổi đồ thị theo một trong hai cách:

Sử dụng các mô hình 3D và thế giới tạo ra trong các ứng dụng bên ngoài. Nội dung sáng tạo các ứng dụng như SketchUp, 3ds Max, Maya và dữ liệu được xuất ra bằng cách sử dụng định dạng COLLADA, và O3D bao gồm một công cụ chuyển đổi mẫu mà có thể được sử dụng với các tập tin COLLADA. Google 3D Warehouse cũng sử dụng định dạng này. Sử dụng công cụ chuyển đổi này như là một mô hình mẫu, cũng có thể viết "nhập khẩu" của riêng và sử dụng cho các tập tin trong bất kỳ định dạng file khác.) Các ví dụ đã tạo ra khung cảnh bên ngoài nhập khẩu là Hello, World và thủ tục Texture ví dụ trong trang web http//…này cũng như O3D trình diễn.

Xây dựng biểu đồ biến đổi từ đầu, cung cấp dữ liệu đỉnh vào chức vụ, normals, màu sắc, và các hiệu ứng, và sau đó xác định rõ ràng cách chức các đối tượng trong không gian 3D. Cách tiếp cận này được sử dụng trong hầu hết các ứng ụng web3D.

Cũng có thể sử dụng một cách tiếp cận kết hợp, nhập khẩu một số mô hình và những người khác tạo ra từ đầu. Trong đa số trường hợp, O3D xây dựng các khung cảnh (phối cảnh), bằng cách sử dụng xem và chiếu ma trận dữ liệu được cung cấp bởi ứng dụng. Ngoài ra, O3D cung cấp hỗ trợ cho việc kiểm soát dựng hình tiên tiến, bao gồm đổ bóng, minh bạch, ánh sáng, độ sâu-of-field tính toán, và nhiều quan điểm đồng thời của cùng một cảnh.

3.5 Gói quản lý bộ nhớ

Khi tạo đối tượng trong O3D, nó là tự động thêm vào một gói, mà đảm bảo rằng các đối tượng không phải là vô tình xóa. Mỗi lần một đối tượng được tham chiếu bởi đối tượng khác, số tham chiếu của nó là tăng thêm 1. Các gói chính giữ một tham chiếu đến từng đối tượng. Chức năng phát hành một tham chiếu đến một đối tượng cá nhân, và pack.destroy phát hành tất cả các tài liệu tham khảo trong gói đó. Nếu gọi đó là loại bỏ các tham chiếu cuối cùng để một tài sản cụ thể, tài sản sẽ được loại bỏ khỏi bộ nhớ.

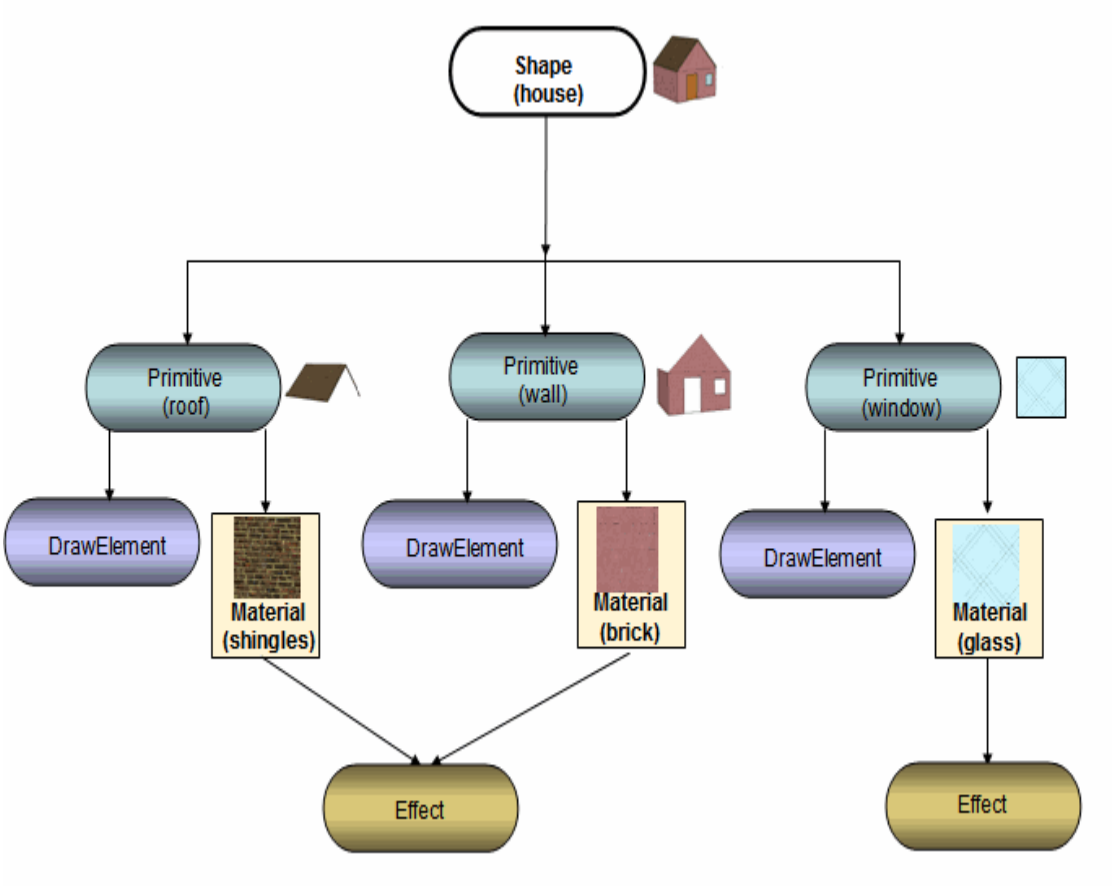
3.6 Tạo đồ thị Render

O3D cung cấp một DrawContext đối tượng được sử dụng để xác định xem ma trận và chiếu ma trận. Quan điểm đại diện cho một ma trận chuyển đổi có thể chuyển đổi từ đỉnh thế giới phối hợp để xem tọa độ. Các chiếu ma trận là một biến đổi có thể chuyển đổi tọa độ để xem clipspace tọa độ. Bất kỳ nội dung 3D rơi bên ngoài của hình cụt xem là bị loại bỏ, hoặc cắt bớt. DrawContext được chia sẻ bởi DrawPass đối tượng và các TreeTraversal đối tượng. Các TreeTraversal đối tượng sử dụng nó cho tiêu hủy, và các DrawPass đối tượng sử dụng nó trong quá trình dựng hình. Có thể chỉ định các ma trận một cách rõ ràng, hoặc nếu đang nhập khẩu các mô hình từ các nguồn khác, O3D có thể có được những thông tin máy ảnh chứa trong nội dung nhập khẩu.

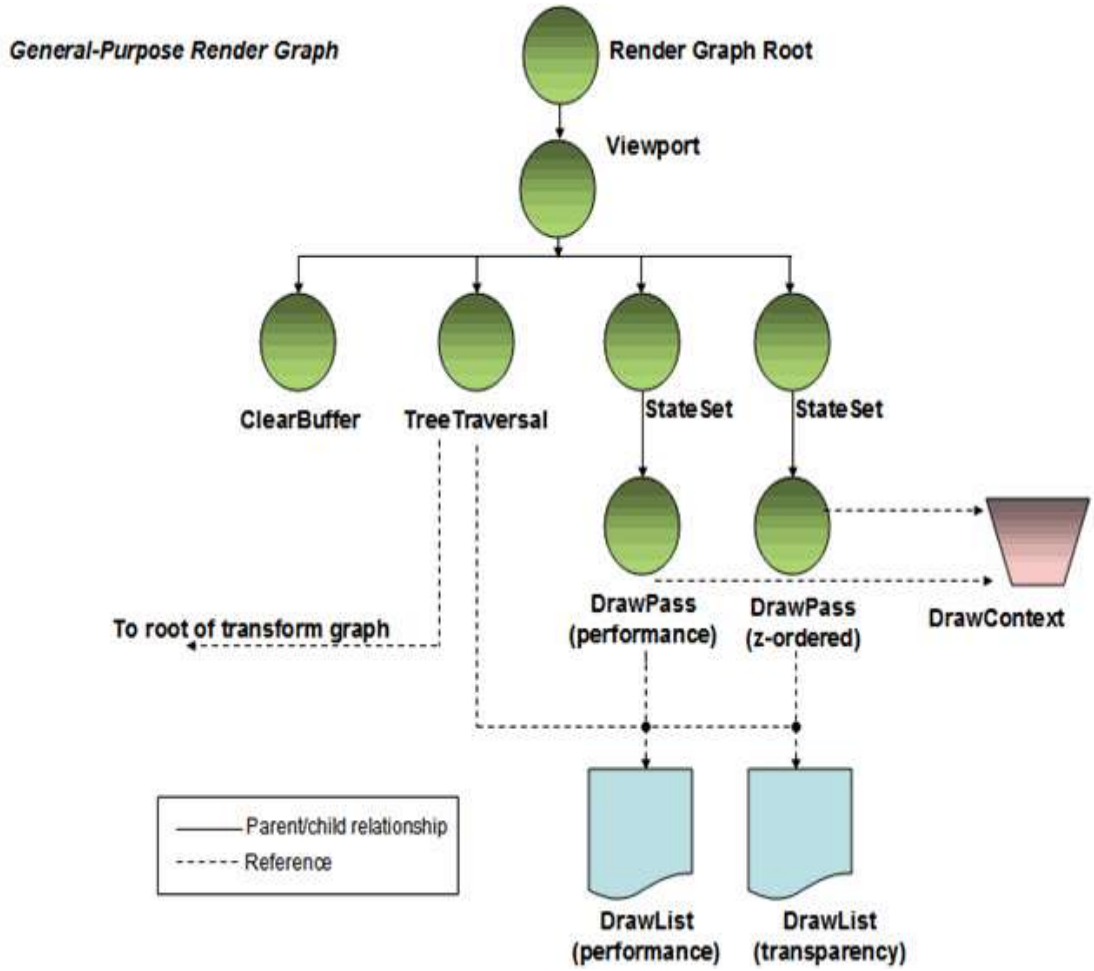
Các createDrawElements() chức năng biến đổi đồ thị và tạo ra một yếu tố thu hút đối với từng nguyên thủy trong chuyển đổi đồ thị. Một yếu tố thu hút về cơ bản là một chỉ dẫn tới "Draw nguyên thủy này." Nếu không có yếu tố thu hút, không có gì được rút ra. Vẽ các yếu tố cho phép O3D hiệu quả rút ra những nguyên sinh cùng nhiều lần (ví dụ, một lần là hình thực tế và một lần như là bóng tối cho hình đó). Trong một số trường hợp, các yếu tố thu hút sử dụng các vật liệu giao cho

nguyên thủy. Trong trường hợp khác, các yếu tố thu hút có thể có vật chất riêng của mình được giao (ví dụ, cho bóng). Trong trường hợp cả hai yếu tố nguyên thủy và các vật liệu vẽ đã được phân công, vật liệu giao cho các yếu tố thu hút các vật liệu ghi đè được giao trước đó để các nguyên thủy.

Một yếu tố thu hút là một chỉ dẫn tới "Draw này nguyên thủy" với các tài liệu quy định và có hiệu lực



Một đồ thị vẽ điển hình, được tạo ra bằng cách sử dụng chức năng tiện ích JavaScript renderGraph.createBasicView , chứa các đối tượng sau đây:



Các đối tượng trong biểu đồ vẽ lại được đi qua (có nghĩa là, đọc và thực thi) từ trên xuống dưới, từ trái sang phải (theo ưu tiên). Đây là một giải thích ngắn gọn về những tác vụ được thực hiện bởi các đối tượng trong biểu đồ này vẽ điển hình:

+ Các Viewport đối tượng thiết lập khu vực hình chữ nhật trên màn hình nơi dựng hình tiếp theo sẽ xảy ra (vẽ đồ thị có thể có nhiều đối tượng viewport). Các thiết lập của Viewport đối tượng được thừa hưởng bởi lớp con của mình trong vẽ đồ thị.

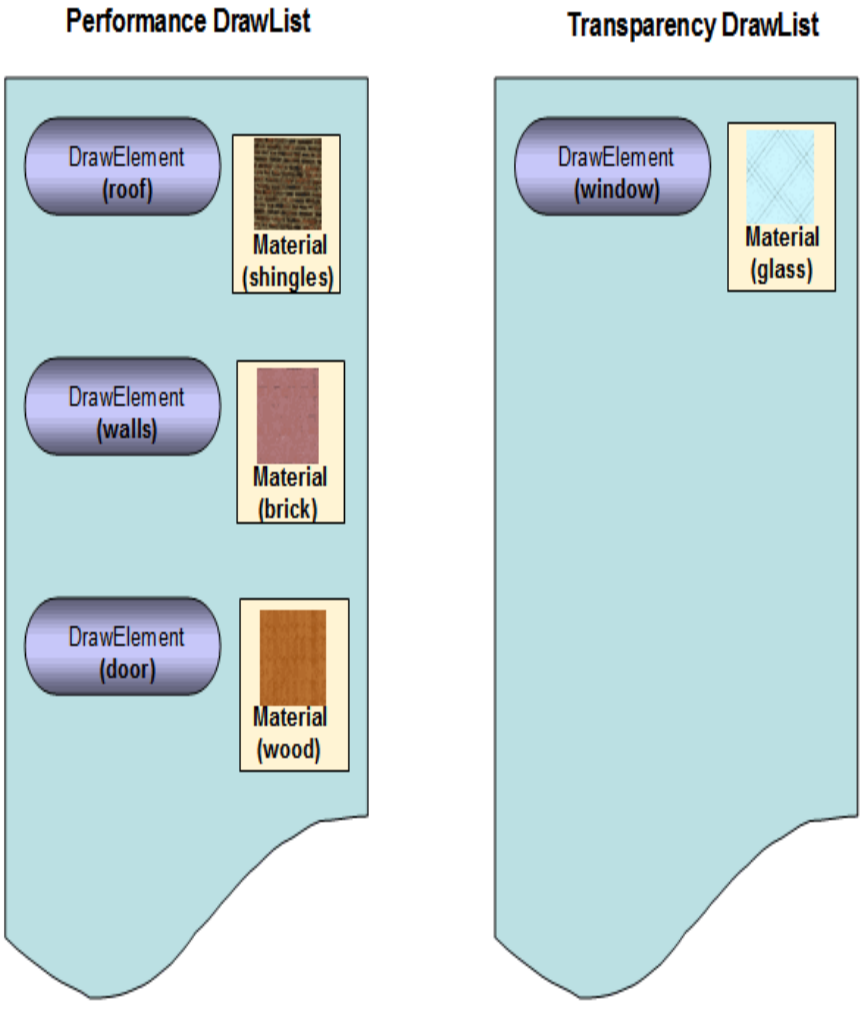
+ Các ClearBuffer đối tượng xóa sạch bộ đệm hiện tại-trong trường hợp này, màn hình.

+ Các TreeTraversal đối tượng đi qua các biến đổi đồ thị và cho biết thêm mỗi yếu tố thu hút cho một hoặc nhiều danh sách rút ra, như thể hiện trong biểu đồ dưới đây. Đồ thị vẽ tiêu chuẩn có hai danh sách rút ra: một cho hiệu năng dựng hình qua sử dụng cho các vật liệu mờ đục và một cho các z-ra lệnh dựng hình thông

qua, được sử dụng cho vật liệu trong suốt. Các TreeTraversal đối tượng thực hiện một số kiểm tra là nó đi các biến đổi đồ thị, có hiệu quả bỏ qua đối tượng mà không được trả lại. Ví dụ, nếu một biến đổi của visible thông số là FALSE, đó là bỏ qua, và không có yếu tố thu hút được tạo ra cho các đối tượng liên quan đến hình dạng của nó. . Nếu culling tham số cho các chuyển đổi được thiết lập là TRUE, các TreeTraversal ranh giới sử dụng hộp tính toán để xác định xem các biến đổi của lớp con được chứa trong khu vực xem của DrawContext cho các liên kết DrawList .

+ Các StateSet render các đối tượng thiết lập các pham vi khác nhau được thừa kế bởi các lớp con. Ví dụ, StateSet đó là cha mẹ của z-ra lệnh DrawPass đối tượng lượt về pha trộn alpha (cho minh bạch).

+ Mỗi DrawPass đối tượng của nó DrawList , do đó có tất cả các yếu tố thu hút tập hợp bởi các TreeTraversal để vượt qua điều đó.



CHƯƠNG 4: ỨNG DỤNG MÔ PHỎNG SỬ DỤNG O3D PLUGIN

4.1 Nhu cầu mô phỏng 3D

Công nghệ 3D hiện nay, đã có mặt hầu hết trong mọi ứng dụng của đời sống, 3D đã được ứng dụng trong: thiết kế kiến trúc, nội ngoại thất, gian hàng, hội chợ , Thiết kế mẫu 3D, nữ trang, thiết kế game 3d, phim hoạt hình 3d, quảng cáo, điện ảnh, giáo dục, mô phỏng thực tế ảo .....

Với công nghệ 3D giúp người thiết kế có thể xây dựng, mô tả nhiều đặc trưng của hệ thống cơ, một khi xây dựng xong mô hình, người thiết kế có thể tiến hành mô phỏng đặt các lực (ngẫu lực, trọng lực, lực tập trung, lực ma sát v.v) lên mô hình để khảo sát.

Cho phép mô phỏng nhiều dạng liên kết cơ bản như liên kết thanh, khớp quay, tời, rãnh trượt hay các liên kết phức tạp như bánh răng, motor, cơ cấu chấp hành, lò xo thẳng, lò xo xoắn, giảm chấn v.v. Thực hiện hiệu chỉnh quá trình mô phỏng với những công cụ tác động lên đối tượng với các ràng buộc cho trước. Có thể hiệu chỉnh các tham số trong những điều kiện môi trường khác nhau.

Có thể thực hiện, dừng, hiệu chỉnh lại quá trình mô phỏng tại bất kỳ thời điểm nào

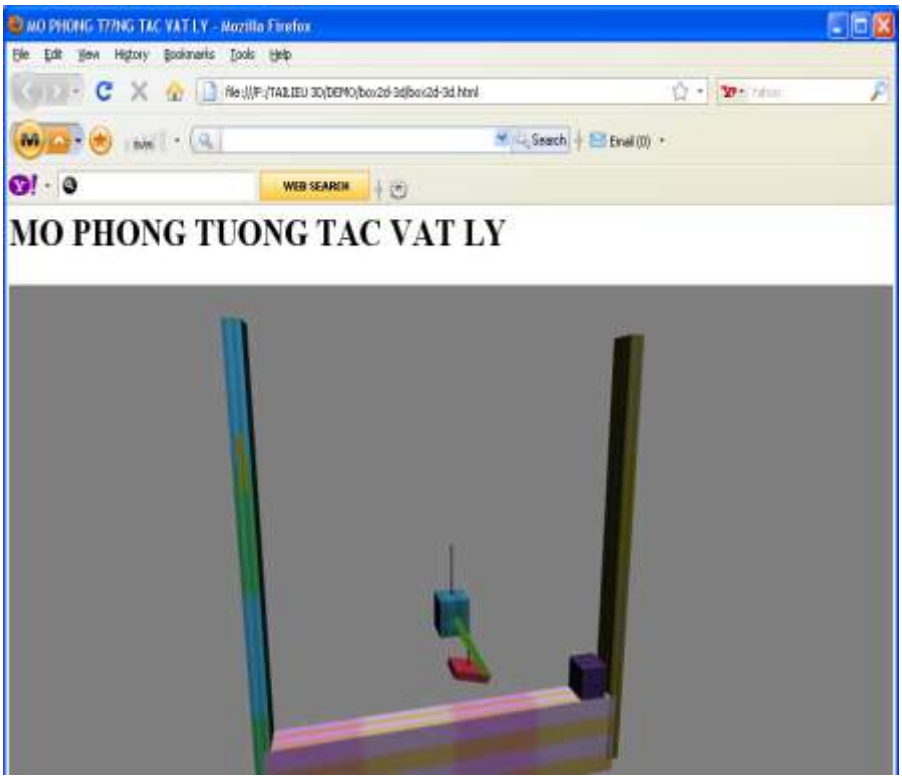
Khả năng mô hình hóa cho các phương pháp phân tích . Kết quả có thể xuất ra dưới định dạng vector, giá trị số hay đồ thị với các hệ đơn vị khác nhau.

Giảm chi phí trong thiết kế tạo mẫu.

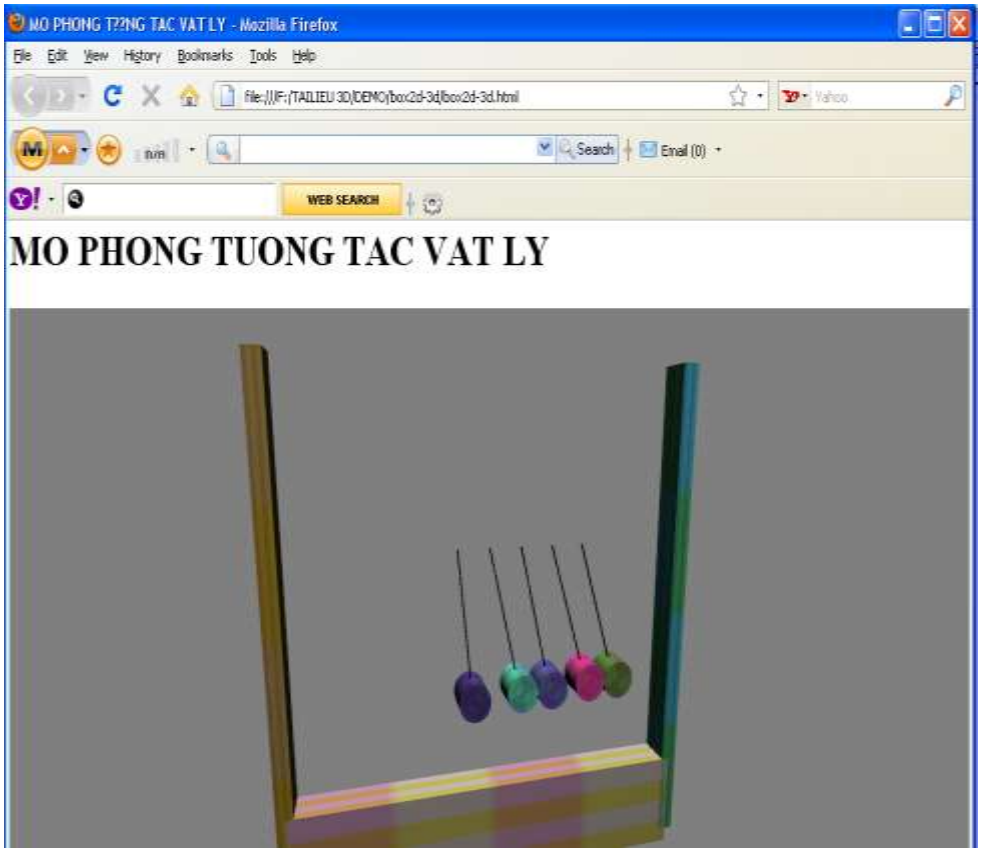
Có thể phân tích kết cấu tĩnh cho kết quả biểu đồ nội lực, mô phỏng quá trình thường gặp trong cuộc sống như tiếp xúc, va chạm, ma sát.

4.2 Xây dựng mô phỏng tương tác vật lý sử dụng O3d Plugin

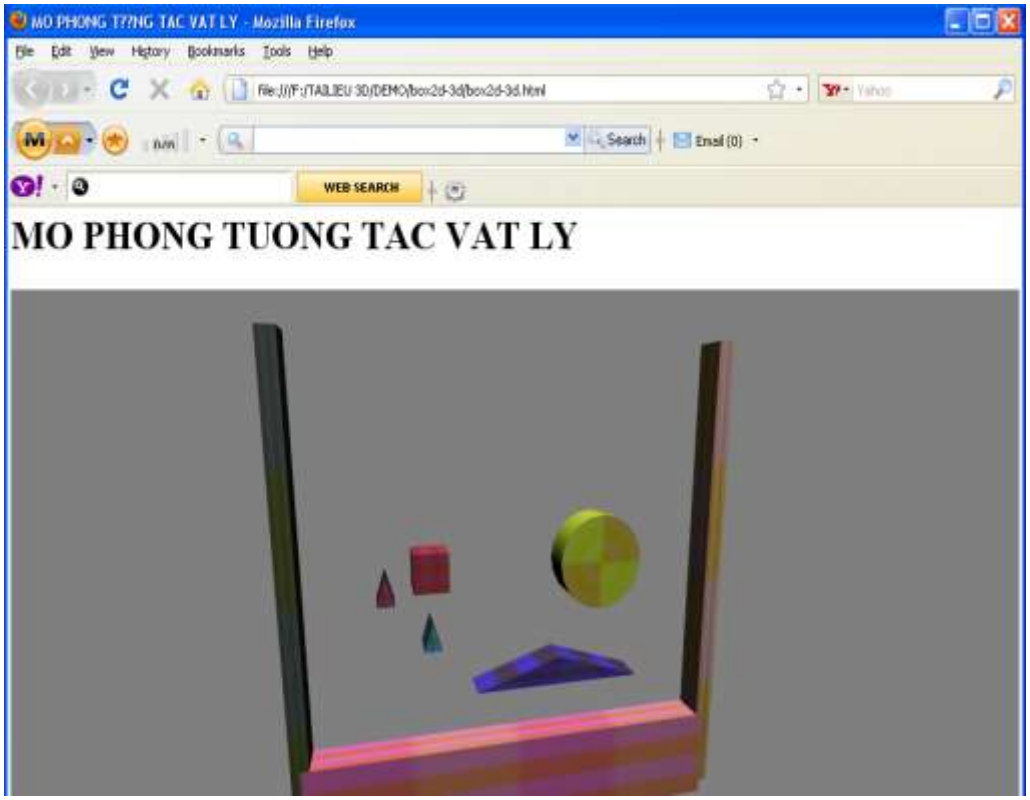
Mô phỏng về lực đẩy cuả pittông



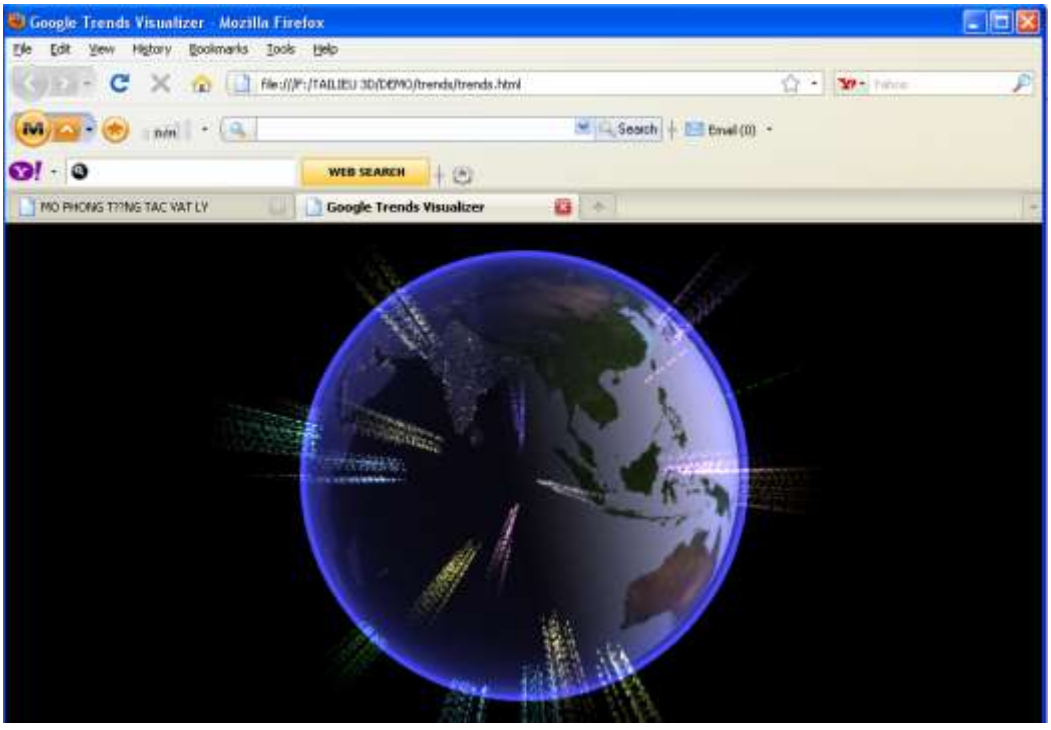
Mô phỏng về con lắc đơn



Mô phỏng về chuyển động của đối tượng



4.3 Xây dựng mô phỏng địa lý



4.3.1 Mã nguồn minh họa

Tạo tham chiếu va thiết lập kích thước cho đối tượng

```
function setClientSize() {
  var newWidth = g.client.width;
  var newHeight = g.client.height;

  if (newWidth != g.o3dWidth || newHeight != g.o3dHeight) {
    g.o3dWidth  = newWidth;
    g.o3dHeight = newHeight;

    // Create a perspective projection matrix(tao mot tham chieu diem
trong ma tran)
    g.viewInfo.drawContext.projection = g.math.matrix4.perspective(
      g.math.degToRad(45), g.o3dWidth / g.o3dHeight, 0.1, 5000);

    // Sets a new area size for arcball.(thiet lap kich thuoc cho qua?
cau)
    g.aball.setAreaSize(g.o3dWidth, g.o3dHeight);
  }
}
```

Tạo ra một đồ thị và cài đặt nền màu

```
// Create the render graph for a view.(tao ra 1do thi)
  g.viewInfo = o3djs.rendergraph.createBasicView(
      g.pack,
      g.client.root,
      g.client.renderGraphRoot);

  // Set the background color to black.(cai dat nên mau den)
  g.viewInfo.clearBuffer.clearColor = [0, 0, 0, 0];

  // Set states for shards.(thiet lap gioi han cho manh)
  g.viewInfo.zOrderedState.getStateParam('CullMode').value =
      g.o3d.State.CULL_NONE;

g.viewInfo.zOrderedState.getStateParam('DestinationBlendFunction').value
=
      g.o3d.State.BLENDFUNC_ONE;
  g.viewInfo.zOrderedState.getStateParam('ZWriteEnable').value = false;

  g.viewInfo.performanceDrawPass.sortMethod = g.o3d.DrawList.BY_PRIORITY;

  g.lastRot = g.math.matrix4.identity();
  g.thisRot = g.math.matrix4.identity();

  var root = g.client.root;
```

Tạo ra tham số cho các vị trí mặt trời và mắt của chúng ta có thể nhìn được

```
// Create a param for the sun and eye positions that we can bind(tao
1tham so cho cac vi tri mat troi va mat cua chng ta co the nhin dc) )
  // to auto update a bunch of materials.(de tu dong thiet lap cac nguyen
vat lieu)
  g.globalParams = g.pack.createObject('ParamObject');
  g.sunPosParam = g.globalParams.createParam('sunPos', 'ParamFloat3');
  g.sunPosParam.value = [1000, 200, 100];
  g.eyePosParam = g.globalParams.createParam('eyePos', 'ParamFloat3');

  updateViewFromCamera();

  g.aball = o3djs.arcball.create(100, 100);
  setClientSize();

  g.client.setRenderCallback(onRender);
```

Tạo ra vật liệu

```
// Create Materials.(tao vat lieu)
  var effectNames = [
    "noTexture",
    "dayOnly",
    "nightAndDay",
    "mask",
    "energy",
    "atmosphere"
    ];
  g.materials = [];
  for (var ii = 0; ii < effectNames.length; ++ii) {
    var effectName = effectNames[ii];
    var effect = g.pack.createObject('Effect');
    effect.loadFromFXString(document.getElementById(effectName).value);
// Create a Material for the effect.(tao ra 1vat lieu cho hieu luc)
    var material = g.pack.createObject('Material');

    // Apply our effect to this material. The effect tells the 3D
hardware(ap dung hieu ung)
    // which shader to use.(do bong)
    material.effect = effect;

    // Set the material's drawList(thiet lap cac tai lieu cua drawlist)
    material.drawList = g.viewInfo.performanceDrawList;

    // This will create the effects's params on the material.(dieu nay
tao ra cac hieu ung tren vat lieu)
    effect.createUniformParameters(material);
```

Tạo ra nặng lượng kết cấu

```
// Create energy texture(s)(tao ra nang luong ket cau)
  {
    var dots = [ 0, 1, 0, 1, 0, 0, 1, 0,
                 1, 0, 0, 1, 0, 1, 0, 0,
                 1, 0, 1, 0, 0, 0, 1, 0,
                 0, 1, 0, 1, 0, 0, 1, 0 ];
    var texture = g.pack.createTexture2D(3,
                                         dots.length,
                                         g.o3d.Texture.XRGB8,
                                         1,
                                         false);
    var pixels = [];
    for (var yy = 0; yy < dots.length; ++yy) {
      for (var xx = 0; xx < 3; ++xx) {
        var pixelOffset = (yy * 3 + xx) * 3;
        var color = (xx == 1) ? dots[yy] : 0;
        for (var cc = 0; cc < 3; ++cc) {
          pixels[pixelOffset + cc] = color;
        }
      }
    }
    texture.set(0, pixels);
    g.energySampler.texture = texture;
  }
```

Tạo một hình cầu cho trái đất

```
// Create a sphere at the origin for the earth.(tao mot hinh cau cho trai
dat)
  var earth = o3djs.primitives.createSphere(g.pack,
                                            g.noTextureMaterial,
                                            25,
                                            50,
                                            50);
// Get a the element so we can set its material later.
  g.earthPrimitive = earth.elements[0];
  g.atmosphereState = g.pack.createObject('State');
  g.atmosphereState.getStateParam('AlphaBlendEnable').value = true;
  g.atmosphereState.getStateParam('SourceBlendFunction').value =
      g.o3d.State.BLENDFUNC_SOURCE_ALPHA;
  g.atmosphereState.getStateParam('DestinationBlendFunction').value =
      g.o3d.State.BLENDFUNC_INVERSE_SOURCE_ALPHA;
  g.atmosphereState.getStateParam('ZWriteEnable').value = false;
  g.atmosphereMaterial.state = g.atmosphereState;

  g.root = g.pack.createObject('Transform');
  g.root.parent = g.client.root;
  g.earth = g.pack.createObject('Transform');
  g.earth.addShape(earth);
  g.earth.parent = g.root;
// Create a sphere at the origin for the atmosphere.(tao hinh cau tai
nguon goc cua khi quyen)
  var atmosphere = o3djs.primitives.createSphere(g.pack,
                                                 g.atmosphereMaterial,
                                                 26,
                                                 50,
                                                 50);
  g.atmospherePrimitive = atmosphere.elements[0];
  g.atmospherePrimitive.priority = 1;
  g.atmosphere = g.pack.createObject('Transform');
  g.atmosphere.addShape(atmosphere);
  g.atmosphere.parent = g.root;

  g.energyShape = createEnergyShape(g.pack,
                                    g.energyMaterial,
                                    g.ENERGY_WIDTH,
                                    g.ENERGY_HEIGHT);

  addEnergyShard(0, 0, 1, 1, [1, 1, 1, 1]);
```

KẾT LUẬN

Lĩnh vực đồ họa 3D ngày càng phát triển và có nhiều đóng góp quan trọng trong cuộc sống của con người. Nhu cầu tạo ra các thế giới ảo 3 chiều ngày càng nhiều. Với đề tài "Tìm hiểu về một 3D Plug-in và ứng dụng", khóa luận này đã trình bày được tổng quan về các kỹ thuật đồ họa 3 chiều cơ bản, trong đó tập trung đi sâu vào nghiên cứu và xây dựng mô phỏng các tương tác vật lý, mô phỏng về địa lý, sử dụng về 3D Plung-in. Bước đi đầu tiên để tạo ra các ứng dụng có quy mô lớn hơn và hoàn thiện hơn.

Trong tương lai em sẽ tiếp tục nghiên cứu thêm để có thể hoàn thiện hơn nữa về giao diện và tính năng của chương trình nhằm trở thành một sản phẩm thương mại hoàn thiện, mang tính chuyên nghiệp cao.

Tuy nhiên do hạn chế về điều kiện và thời gian, khoá luận sẽ không thể tránh khỏi những thiếu xót. Kính mong được sự đóng góp ý kiến của thầy cô và các bạn để em có thể hoàn thiện hơn đề tài nghiên cứu của mình trong đợt làm khoá luận tốt nghiệp này.

Em xin trân trọng cảm ơn!

TÀI LIỆU THAM KHẢO

[1.] James D.Foley, Andrie van Dam, Steven K.Feiner, Jonhn F. Hughes, Computer Graphics Principles and Practice, Addison Wesley, 1994.

[2.] Lê Tấn Hùng, Huỳnh Quyết Thắng. Kỹ thuật đồ hoạ máy tính, NXB khoa học và kỹ thuật, 2002.

[3.] Steven Harrington, Computer Graphics A Programming Approach, McGraw Hill International Edition, 1987.

[4.] http://code.google.com/apis/o3d/