

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---



**NGUYỄN THỊ THU HƯƠNG**

**KỸ THUẬT RENDER VOLUME TRONG HIỂN THỊ DỮ  
LIỆU 3D TỪ HÌNH CHIẾU**

**Chuyên ngành: KHOA HỌC MÁY TÍNH**

**Mã số: : 60.48.01**

**TÓM TẮT LUẬN VĂN THẠC SỸ KỸ THUẬT**

**HÀ NỘI – NĂM 2012**

Luận văn được hoàn thành tại:  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**Người hướng dẫn khoa học: PGS.TS. ĐỖ NĂNG TOÀN**

Phản biện 1:.....

Phản biện 2: .....

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận  
văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn  
thông

Vào lúc: .....giờ.....ngày.....tháng.....năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính  
Viễn thông

# LỜI MỞ ĐẦU

## Lý do chọn đề tài

Cùng với phát triển của Đồ họa máy tính và Xử lý ảnh số trong máy tính làm thay đổi sự tương tác giữa người và máy, khi các kỹ thuật ứng dụng đồ họa ngày càng cao hơn nên nhiều người quan tâm nghiên cứu đến lĩnh vực này. Do đó mà các ứng dụng đồ họa trên máy tính được ra đời như: phim hoạt hình, Game với các hệ thống thực tại ảo... đã đóng góp cho sự phát triển chung của ngành Công nghệ thông tin. Vì vậy đồ họa máy tính trở thành một lĩnh vực hấp dẫn và có nhiều ứng dụng trong thực tế.

Hiện thị dữ liệu 3 chiều (3D) từ hình chiếu được coi là các bước khởi đầu cho hệ thống mô phỏng thực tại ảo, góp phần tạo nên hệ thống mô phỏng hoàn chỉnh. Một trong những cách tiếp cận hiện thị dữ liệu 3 chiều phổ biến hiện nay là dựa trên kỹ thuật Render volume.

Trong quá trình hiện thị dữ liệu 3 chiều ngoài các vấn đề hiện thị dữ liệu đảm bảo chất lượng còn phải đáp ứng yêu cầu về thời gian hiện thị cho các bước mô phỏng. Hiện thị hình ảnh ba chiều của các vật thể thực từ hình chiếu là một trong những lĩnh vực thu hút được sự quan tâm nhiều nhất của giới nghiên cứu trong lĩnh vực công nghệ thông tin trong mấy chục năm qua. Hình ảnh hiện thị từ máy tính đã được sử dụng hiệu quả trong nhiều lĩnh vực khác nhau như giáo dục, giải trí, kiến trúc, đặc biệt là chuẩn đoán hình ảnh trong y tế, ... Kỹ thuật Render volume trong hiện thị hình ảnh 3D từ hình chiếu là một đề tài mới mẻ và có ứng dụng lớn trong trong lĩnh vực tái tạo và phục dựng đối tượng. Xuất phát từ thực tế đó luận văn lựa chọn đề tài: “Kỹ thuật render volume trong hiện thị dữ liệu 3D từ hình chiếu”

## Mục đích của đề tài

- Nghiên cứu Kỹ thuật Render Volume trong hiện thị hình ảnh 3D từ hình chiếu.
- Trên cơ sở kiến thức tìm hiểu được, cài đặt thử nghiệm chương trình hiện thị hình ảnh 3D từ các ảnh chụp cắt lớp trong y tế.

## Tổ chức của luận văn

Nội dung của luận văn gồm có:

**Phần mở đầu:** Trình bày lý do chọn đề tài và mục đích cũng như tổ chức của luận văn.

**Chương 1:** Khái quát về đồ họa 3 chiều (3D) và hiển thị dữ liệu từ hình chiếu. Trong chương này trình bày khái quát về đồ họa 3D, các ứng dụng cơ bản của đồ họa ba chiều và bài toán cũng như các cách tiếp cận trong việc hiển thị dữ liệu từ hình chiếu.

**Chương 2:** Một số Kỹ thuật hiển thị ảnh 3 chiều từ hình chiếu. Trong chương này giới thiệu một số kỹ thuật về hiển thị hình ảnh 3D từ các hình ảnh hình chiếu 2D

**Chương 3:** Chương trình thử nghiệm. Trình bày cài đặt thử nghiệm kỹ thuật hiển thị hình ảnh Render Volume từ các ảnh 2D của ảnh y tế DICOM

Và cuối cùng kết luận của đồ án về những việc đã làm được và những hạn chế của luận văn.

## **CHƯƠNG 1: TỔNG QUAN VỀ ĐỒ HOẠ 3 CHIỀU VÀ HIỂN THỊ DỮ LIỆU TỪ HÌNH CHIẾU**

Trong chương này trình bày khái quát về đồ họa 3D, các ứng dụng cơ bản của đồ họa ba chiều, tiêu chuẩn ảnh Dicom sử dụng trong ngành y tế và bài toán cũng như các cách tiếp cận trong việc hiển thị dữ liệu từ hình chiếu.

### **1.1. Khái quát về đồ họa 3 chiều**

#### **1.1.1. Đồ họa 3D**

##### **+ Phương pháp biểu diễn 3D**

Có hai phương pháp biểu diễn đối tượng ba chiều là phương pháp biểu diễn bề mặt (B-reps) và biểu diễn theo phân hoạch không gian (space-partitioning representation).

##### **+ Các phép biến đổi hình học :**

Các phép biến đổi thường sử dụng là phép tịnh tiến, phép quay, phép biến dạng,... Các phép biến đổi được mô tả bằng các ma trận. Ma trận của mỗi phép biến đổi có dạng khác nhau.

##### **+ Vấn đề chiếu sáng (illumination) :**

Vật thể được chiếu sáng nhờ vào các ánh sáng đến từ nguồn sáng sau khi phản xạ nhiều lần qua các vật thể xung quanh vật thể ta đang quan sát. Do vậy ánh sáng đến được vật là ánh sáng tổ hợp từ khắp mọi hướng, ta gọi đó là *ánh sáng xung quanh (ambient light)* hay *ánh sáng nền (background light)*.

##### **+ Trực quan hóa (Visualization)**

Trực quan hóa trong đồ họa máy tính là sử dụng máy tính để tính toán dữ liệu sau đó sử dụng đồ họa máy tính, đặc biệt là đồ họa 3D để minh họa, biểu diễn dữ liệu thành

những hình ảnh mà con người có thể hiểu được dễ dàng và giúp cho con người có thể tương tác với dữ liệu[4]. Dữ liệu đó có thể là các dữ liệu phát sinh do mô phỏng hoặc do đo đạc trong thực tế. Kết quả biểu diễn phải biểu diễn chính xác tính chất của tập dữ liệu.

### **1.1.1.1 Các kỹ thuật đồ họa**

#### **1.1.1.1.1 Kỹ thuật đồ họa điểm (Sample based-Graphics)**

- 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 hình, hình ảnh đối tượng mà chúng ta muốn hiển thị.

#### **1.1.1.1.2. Kỹ thuật 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

#### **1.1.1.2. Các chuẩn giao diện của hệ đồ họa**

Mục tiêu căn bản của phần mềm đồ họa được chuẩn là tính tương thích. Khi các công cụ được thiết kế với hàm đồ họa chuẩn, phần mềm có thể được di chuyển một cách dễ dàng từ hệ phần cứng này sang hệ phần cứng khác và được dùng trong nhiều cài đặt và ứng dụng khác nhau.

### **1.1.2. Các ứng dụng cơ bản của đồ họa 3D**

Những lĩnh vực đang được nghiên cứu ứng dụng đồ họa 3D một cách mạnh mẽ hiện nay là: Y học, Giáo dục, Tin học, Thương mại, Giao thông, Hàng không, Xây dựng Thiết kế nội thất và trang trí nhà cửa, Giải trí, Quân sự, Điện ảnh...

#### **Ý nghĩa của việc ứng dụng đồ họa 3D**

Như vậy chúng ta thấy được ý nghĩa to lớn của việc ứng dụng đồ họa 3D, bởi những vấn đề khó khăn mà nếu không có đồ họa 3D thì có thể nói là khó lòng mà giải quyết, hay nếu có thể giải quyết được thì hiệu quả không cao và chi phí sẽ rất tốn kém. Còn khi ứng dụng đồ họa 3D vào, thì những vấn đề đó trở lên hết sức đơn giản, và hiệu quả của nó mang lại thì thực sự là to lớn, kể cả vật chất lẫn tinh thần.

## 1.2 Hiện thị dữ liệu từ hình chiếu

### 1.2.1. Giới thiệu

Đồ họa 3 chiều (3D computer graphics) bao gồm việc bổ xung kích thước về chiều sâu của đối tượng, cho phép ta biểu diễn chúng trong thế giới thực một cách chính xác và sinh động hơn.

#### 1.2.1.1. Đặc điểm của kỹ thuật đồ họa 3D

Có các đối tượng phức tạp hơn các đối tượng trong không gian 2D.

- Bao bởi các mặt phẳng hay các bề mặt.
- Có các thành phần trong và ngoài.

Các phép biến đổi hình học phức tạp.

Các phép biến đổi hệ tọa độ phức tạp hơn.

Thường xuyên phải bổ sung thêm phép chiếu từ không gian 3D vào không gian 2D luôn phải xác định các bề mặt hiển thị.

#### 1.2.1.2 Các phương pháp hiển thị 3D

Với các thiết bị hiển thị 2D thì chúng ta có các phương pháp sau để biểu diễn đối tượng 3D:

- Kỹ thuật chiếu(projection):Trực giao (orthographic)/phối cảnh(perspective)
- Kỹ thuật đánh dấu độ sâu (depth cueing)
- Nét khuất (visible line/surface identification)
- Tô chất bề mặt (surface rendering)
- Cắt lát (exploded/cutaway scenes, cross-sections)

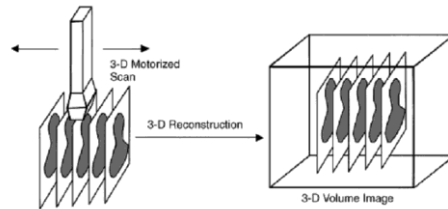
#### Các thiết bị hiển thị 3D:

- Kính stereo - Stereoscopic displays\*
- Màn hình 3D – Holograms

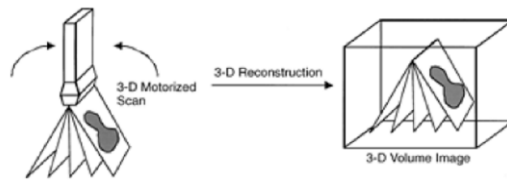
#### 1.2.1.4. Tái tạo cấu trúc ba chiều từ các hình chiếu

Quá trình lấy mẫu thông thường là dùng các thiết bị để thu thông tin bên trong vật thể dưới dạng các lát cắt 2D. Các tập ảnh 2D gồm một số dạng: các ảnh cắt lớp song song

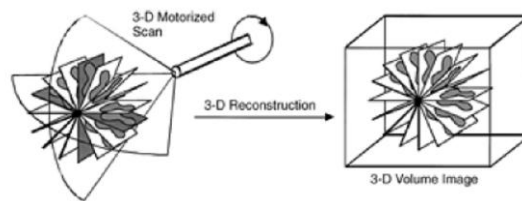
(parallel, serial, translation), các ảnh cắt lớp xuyên tâm (oscillation, rotation), các ảnh cắt lớp tự do (freehand). Ảnh cắt lớp song song thường do các hệ thống máy CT, MRI, siêu âm...tạo ra, đây cũng là dạng thường gặp nhất. Ảnh cắt lớp xuyên tâm thường do máy siêu âm tạo ra. Ảnh cắt lớp theo kiểu tự do thường gặp ở các hệ thống siêu âm. Các ảnh 2D trong tái tạo ảnh nổi là một dạng khác, đây là các hình chiếu thu được từ các cảm biến hoặc các camera đặt xung quanh vật thể.



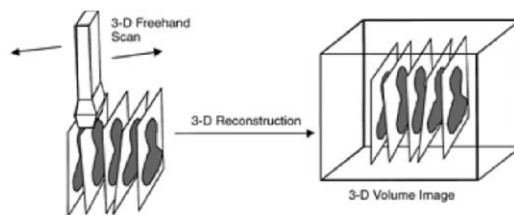
a)



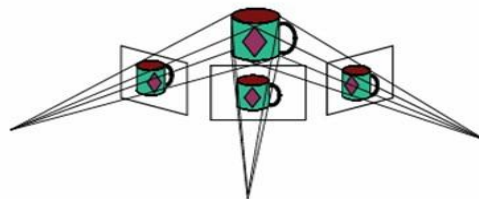
b)



c)



d)



a) Ảnh quét song song (translation), b) Ảnh quét oscillation c) Ảnh quét rotation, d) Ảnh quét tự do (freehand) e) Ảnh nổi (stereo)

### Hình 1.8 Các dạng ảnh 2D dùng để tái tạo ảnh 3D thường gặp

Nguyên tắc của quá trình tái tạo ảnh ba chiều từ các tập ảnh cắt lớp là tìm cách sắp xếp lại các dữ liệu từ các lát cắt sao cho phù hợp với vị trí không gian thực tế của chúng, sau đó dùng đồ họa máy tính để biểu diễn thành các hình ảnh. Ví dụ với các lát cắt song song ta sẽ sắp xếp các lát cắt này song song với nhau như xếp các đĩa CD trên giá. Với các lát cắt tự do thì việc sắp xếp khá phức tạp, chúng ta cần các cảm biến vị trí không gian tại các đầu dò để xác định chính xác vị trí của lát cắt.

#### 1.2.2. Chuẩn Dicom trong Y tế.

##### 1.2.2.1. Giới thiệu

Tiêu chuẩn DICOM cho phép việc tích hợp dễ dàng các máy thu nhận hình ảnh, server, trạm làm việc (workstation), máy in và các thiết bị phần cứng khác có nối mạng từ các nhà sản xuất khác nhau vào trong hệ thống PACS (Hệ thống PACS được ứng dụng trong quá trình thu thập, truyền tải, lưu trữ, quản lý, chẩn đoán, xử lý thông tin của các thiết bị trị liệu kỹ thuật số như CT, MR, US, X quang, DSA, CR.). Các thiết bị khác nhau được đi kèm một bảng đáp ứng các tiêu chuẩn DICOM để làm rõ các lớp dịch vụ mà thiết bị này hỗ trợ. DICOM đã dần dần được chấp nhận rộng rãi ở các bệnh viện và phòng khám.

##### 1.2.2.2. Phần Header

##### 1.2.2.3. Tập dữ liệu - Data Set

#### 1.3 Kết luận chương 1

Chương này đã trình bày khái quát về đồ họa 3D, các kỹ thuật đồ họa như đồ họa điểm, đồ họa vector đồng thời đi tìm hiểu các chuẩn giao diện đồ họa, các ứng dụng cơ bản của đồ họa ba chiều như ứng dụng trong y học, trong quân đội, trong vũ trụ, trong dịch vụ giải trí,... tìm hiểu cách tái tạo ảnh 3 chiều từ hình chiếu và giới thiệu tiêu chuẩn ảnh Dicom sử dụng trong ngành y tế và bài toán cũng như các cách tiếp cận trong việc hiển thị dữ liệu từ hình chiếu. Trong chương sau ta sẽ đi tìm hiểu một số kỹ thuật hiển thị ảnh 3 chiều từ hình chiếu

## CHƯƠNG 2: MỘT SỐ KỸ THUẬT HIỂN THỊ HÌNH ẢNH 3 CHIỀU TỪ HÌNH CHIẾU

*Trong chương này giới thiệu một số kỹ thuật về hiển thị hình ảnh 3D từ các hình ảnh hình chiếu 2D như kỹ thuật Render volume, kỹ thuật biểu diễn bề mặt.*

### 2.1. Kỹ thuật Render Volume





**Hình 2.1 Hình ảnh 3D biểu diễn theo kỹ thuật VR**

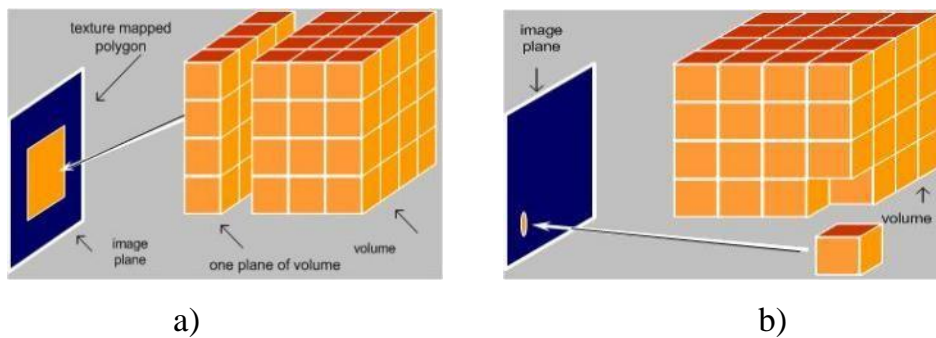
Volume Rendering là kỹ thuật chuyển các mẫu dữ liệu (sampled data) vào trong một bức ảnh. Đây là kiểu biểu diễn trực tiếp (direct display) tức là chuyển trực tiếp các dữ liệu thể tích từ khối dữ liệu đã được sắp xếp thành các pixel trên màn hình.

### 2.1.1. Quy trình (Rendering Process)

Quy trình chuyển khối dữ liệu thành hình ảnh gọi là biểu diễn thể tích. Thông thường biểu diễn thể tích có 3 bước sau :

- Tạo một RGBA volume từ khối dữ liệu
- Xây dựng một hàm liên tục từ các giá trị rời rạc.
- Chiếu lên một mặt phẳng ảnh (image plane) từ một điểm nhìn nào đó: Có nhiều kỹ thuật chiếu nhưng đa số đều thuộc một trong hai loại: object - order và image – order

Đối với kỹ thuật SR thuộc loại object – order, ta quét xuyên qua (tranverse) khối từ sau phía sau (back to front hoặc from 3D scene to 2D image), dữ liệu được chiếu lên trên một mặt phẳng ảnh. Kết quả mà mỗi voxel để lại trên mặt phẳng ảnh gọi là các footprint. Một dạng của phương pháp này trải dữ liệu lên một mặt phẳng gọi là Splatting (Lee Westover,1990).

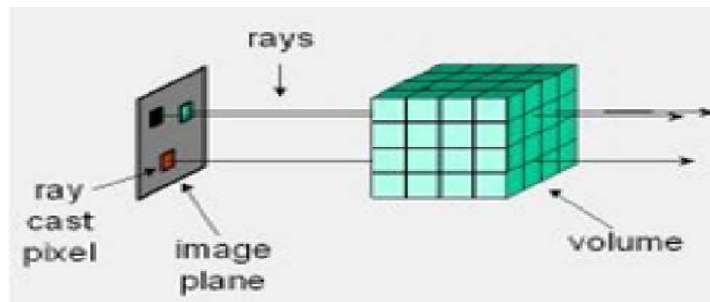


a) Texture mapping plane – by – plane; b)Splatting cell – by – cell

**Hình 2.2 Minh họa kỹ thuật object -order**

Đối với kỹ thuật VR thuộc loại image – order, ảnh được quét lần lượt từng pixel, các

tia chiếu ra (cast) từ mỗi pixel đi xuyên qua thể tích (from 2D image to 3D scene) để xác định giá trị màu sắc cuối cùng cho mỗi pixel. Biểu diễn thể tích kiểu image – order còn gọi là phương pháp ray – casting



**Hình 2.3 Minh họa kỹ thuật image -order**

Tốc độ phương pháp image – order phụ thuộc vào kích thước ảnh trong khi tốc độ của phương pháp object – order phụ thuộc vào kích thước khối.

### 2.1.2. Các phương pháp tạo bố cục ảnh (image composition)

Các phương pháp tạo bố cục cho ảnh thường được sử dụng là X- ray, MIP (maximum intensity projection), MinIP (minimum intensity projection), alpha compositing và NPVR (non-photorealistic volume rendering),...

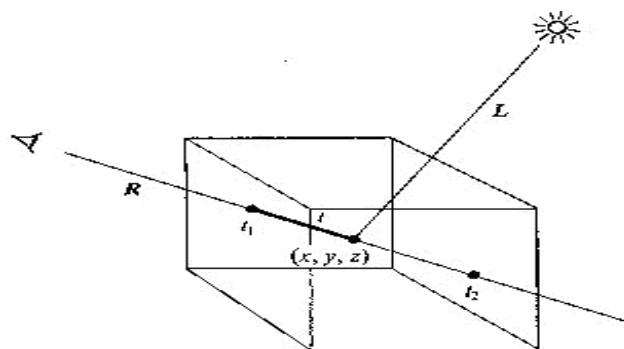
- X-ray : Phương pháp này tính tổng tất cả các giá trị ghi nhận được trên tia chiếu để tạo nên giá trị điểm ảnh.

#### **Nội dung kỹ thuật Volume rendering với Ray casting như sau:**

Mục tiêu cơ bản của kỹ thuật Ray casting là cho phép sử dụng một cách tốt nhất dữ liệu ba chiều không có cấu trúc hình học. Nó đặc biệt phù hợp với hình ảnh y tế.

#### **Cơ sở lý thuyết**

Hiện nay, hầu hết các kỹ thuật Volume rendering sử dụng Ray casting dựa trên mô hình Blinn / Kajiya. Trong mô hình này, chúng tôi có một hình bao có mật độ là  $D(x,y,z)$ , một đường Ray lọt vào.



**Hình 2.4 Mô hình Blinn / Kajiya**

Tại mỗi điểm dọc theo Ray có một chiếu sáng  $I(x,y,z)$  đạt điểm  $(x,y,z)$  từ nguồn sáng. Mắt nhìn sẽ phụ thuộc vào mật độ  $D(x,y,z)$  địa phương dọc theo Ray. Hàm mật độ biểu diễn bằng tham số dọc theo Ray như sau:

$$D(x(t), y(t), z(t)) = D(t)$$

Và sự chiếu sáng từ nguồn sáng:

$$I(x(t), y(t), z(t)) = I(t)$$

Và sự chiếu sáng rải rác dọc theo Ray từ một điểm có khoảng cách  $t$  dọc theo Ray là:

$$I(t)D(t)P(\cos \theta)$$

Trong đó  $\theta$  là góc giữa R và L.

Xác định  $I(t)$  không phải là đơn giản - nó liên quan đến việc tính toán như thế nào từ các nguồn bức xạ ánh sáng thông qua số lượng các điểm quan tâm. Tính toán này giống với tính toán ánh sáng rải rác tại điểm  $(x, y, z)$  ảnh hưởng đến kết quả của Ray tại điểm nhìn. Trong hầu hết các thuật toán, tuy nhiên, tính toán này được bỏ qua và  $I(x, y, z)$  được thiết lập để được thống nhất trong cả hình bao. Đối với hầu hết các ứng dụng thực tế chúng tôi đang quan tâm đến ảnh kết quả, và bao gồm cả dòng tách rời từ một điểm  $(x, y, z)$  với nguồn ánh sáng thực tế có thể không mong muốn. Trong hình ảnh y tế, ví dụ, nó sẽ không thể nhìn thấy vào các khu vực bao quanh bởi xương nếu xương bị coi là dày đặc. Mặt khác, trong các ứng dụng mà bóng nội bộ được mong muốn, tách rời này phải được tính toán.

Sự giảm đi của hàm mật độ có thể được tính như sau:

$$\exp \left[ -\tau \int_{t_1}^{t_2} D(s) ds \right]$$

Trong đó  $\tau$  là một hằng số thể hiện sự giảm đi của hàm mật độ.

Cường độ ánh sáng tới điểm nhìn theo hướng của Ray cho bởi:

$$B = \int_{t_1}^{t_2} \left[ \exp \left[ -\tau \int_{t_1}^t D(s) ds \right] \right] (I(t)D(t)P(\cos \theta)) dt$$

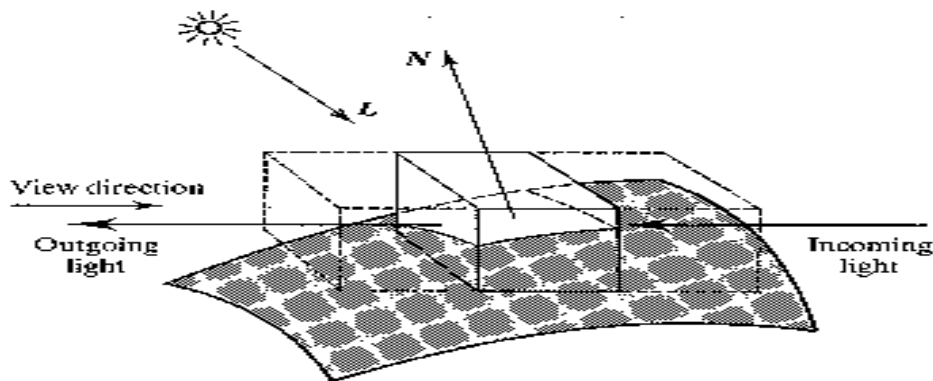
Thực hiện

Khi nói đến “volume visualization”, Ray casting thường được gọi là Ray Tracing. Nói như vậy cũng không thật sự chính xác, vì các phương pháp Ray tracing mà chúng ta đã biết thường là phức tạp hơn so với Ray casting, tuy nhiên ý tưởng cơ bản của hai phương pháp này là giống hệt nhau. Và tất nhiên là kết quả cũng giống hệt nhau.

Thuật toán thực hiện kỹ thuật Ray casting được mô tả ở trên bao gồm cả quá trình

đơn giản hóa tính toán cường độ ánh sáng từ nguồn sáng tới điểm nhìn. Phương pháp mà có thực hiện việc đơn giản hóa này gọi là “additive reprojection”. Nó thực chất là một phép chiếu các Voxels dọc theo một hướng nhìn cố định. Cường độ của Voxels song song và dọc theo tia nhìn là một phép chiếu cung cấp cường độ lên mặt phẳng khung nhìn. Voxels của một chiều sâu quy định có thể được chỉ định một độ mờ tối đa, do đó độ sâu mà hình bao là hình tượng để có thể kiểm soát.

Additive reprojection sử dụng một mô hình chiếu sáng mà là một sự kết hợp của phản ánh và truyền ánh sáng từ các voxel. Tất cả các phương pháp tiếp cận là một tập con của mô hình trong hình dưới đây.



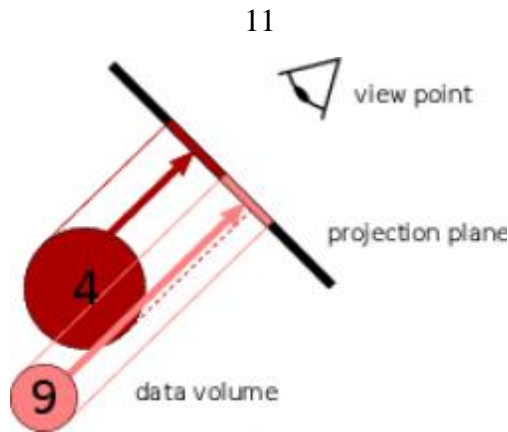
**Hình 2.5 Minh họa kỹ thuật đơn giản hóa tính toán cường độ ánh sáng**

Trong hình trên, ánh sáng đi được tạo thành:

- Ánh sáng phản chiếu theo hướng nhìn từ nguồn ánh sáng.
- Ánh sáng đến bằng cách lọc các voxel.
- Ánh sáng bất kỳ phát ra bởi các voxel.

Đối với mỗi điểm ảnh trong ảnh đầu ra, là kết quả của một tia bắn vào khối dữ liệu. Tại một số điểm thì giá trị màu sắc và độ trong suốt sẽ được tính toán bằng cách nội suy. Các giá trị này sau đó sẽ được kết hợp với nền để tính toán ra màu sắc cho từng pixel ảnh đầu ra.

- Phương pháp MIP : Sử dụng giá trị lớn nhất của các biến trong khối dọc theo một tia vuông góc với mặt phẳng nhìn (view plane) để tạo giá trị (optical property) của mỗi điểm ảnh. Phương pháp MIP ban đầu có nhiều bất tiện vì phải truy cập rất nhiều voxel. Tuy nhiên hiện nay đã có rất nhiều cải tiến cho phương pháp này.



**Hình 2.6 Sơ đồ tổng quan của rendering MIP.**

- Phương pháp MinIP : là một phương pháp trực quan dữ liệu cho phép phát hiện cấu trúc mật độ thấp trong một khối lượng nhất định. Thuật toán sử dụng tất cả các dữ liệu trong một khối lượng quan tâm để tạo ra một hình ảnh duy nhất hai chiều, nói cách khác bao gồm các dữ voxel với giá trị suy giảm thấp nhất trên mỗi điểm trong suốt khối lượng lên một hình ảnh 2D.

Phương pháp này trái ngược với phương pháp MIP khi chúng ta sử dụng giá trị nhỏ nhất dọc theo tia để tạo giá trị của điểm ảnh.

- Phương pháp alpha compositing : Còn có một số tên khác như (Translucency/opacity ). Đây là phương pháp thường được sử dụng phổ biến nhất. Trong phương pháp này các giá số (density value) được đưa vào dọc theo tia để tạo ra màu sắc và độ trong suốt cho ảnh. Giá trị của tia chiếu tại mỗi voxel có thể tính theo công thức sau :

Dạng “Back to Front”

$$V(i) = V(i-1)(1 - a(i)) + c(i).a(i) \text{ Dạng “Front to Back”}$$

$$V(i) = V(i-1) + c(i).a(i).(1-a(i))$$

Trong đó :

$V(i)$  : giá trị của tia chiếu khi ra khỏi voxel thứ  $i$

$V(i-1)$  : giá trị của tia chiếu sau khi ra khỏi voxel thứ  $i - 1$   $a$  : giá trị được chọn để

điều khiển độ chắn sáng.

$c$  : giá trị được chọn để điều khiển độ chói (luminance)

Volume Rendering là kĩ thuật khó vì những lí do: thứ nhất là ở bước shading (tính toán màu sắc cho mỗi điểm dữ liệu trong thể tích) và classification (tính toán độ chắn sáng cho mỗi điểm dữ liệu trong thể tích), ta phải xác định màu sắc và độ chắn sáng (hoặc độ trong suốt) cho toàn bộ khối; thứ hai là khâu chiếu sáng, ta phải phải xét sự tương tác của ánh sáng khuếch tán bên trong vật thể chứ không chỉ trên bề mặt, ta phải tạo ra vật thể có dạng bán trong suốt (semi – transparent) ; thứ ba là hiệu quả, dữ liệu thể tích rất lớn và có

tính tương tác cao nên đòi hỏi phải tính toán rất nhiều và dữ liệu phát sinh trong quá trình tính toán là rất lớn.

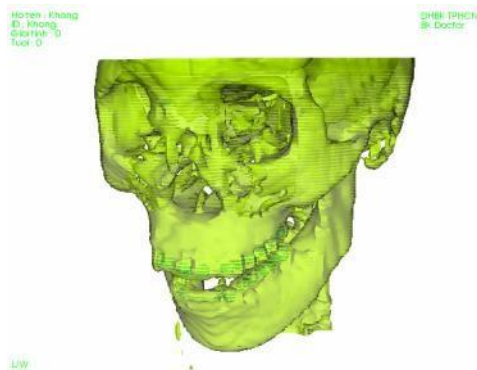
Để tăng tốc độ tính toán trong phương pháp VR người ta thường tìm cách sắp xếp lại dữ liệu để đạt hiệu quả tính toán cao. Ví dụ sắp xếp dữ liệu lại dưới dạng cây cho ta phương pháp Hierarchical Volume Rendering ,... Hiện nay người ta đã có thể thực hiện VR theo thời gian thực.

Phương pháp VR thường dùng để tạo ảnh 3D cho các ảnh có độ tương phản thấp. So với kỹ thuật SR thì kỹ thuật VR đòi hỏi phải tính toán nhiều hơn do đó cần các phần cứng mạnh hơn.

## 2.2. Kỹ thuật biểu diễn bề mặt (surface rendering – SR)

Trong kỹ thuật này chúng ta tạo những bề mặt đi qua các điểm có cùng giá trị vô hướng, những giá trị này trong tiếng Anh isosurface value . Điều này rất có ý nghĩa đối với ảnh y tế vì các ảnh y tế thường là các ảnh đa mức xám. Trên các ảnh cắt lớp các mô cùng loại được thể hiện với cùng một độ xám (gray level). Dùng kỹ thuật này chúng ta có thể tái tạo lại bề mặt của các mô. Ví dụ chúng ta có thể tái tạo hình ảnh của xương sọ hay hình ảnh các mạch máu não từ các ảnh cắt lớp đầu. Các isosurface thường được tô cùng một màu để dễ theo dõi.

Có nhiều thuật toán khác nhau để tạo bề mặt từ các điểm dữ liệu rời rạc. Trong đó chia ra làm hai loại là tạo bề mặt từ các đường viền và tạo bề mặt từ dữ liệu khối.



**Hình 2.7 Hình ảnh 3D được biểu diễn theo phương pháp SR**

### - Tạo bề mặt từ các đường viền (contour based data):

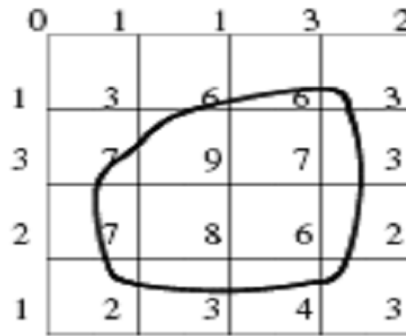
Để tạo bề mặt từ các đường viền cần hai bước: trích biên và tái tạo bề mặt.

+ **Trích biên:** Dùng các thuật toán trích biên để tạo các đường biên trên mỗi lát cắt.

Việc trích biên có thể thực hiện tự động hoặc thực hiện thủ công. Nếu thực hiện tự động người ta dùng một số thuật toán trích biên như LOG của Marr và Canny, thuật toán Snake

model của Terzopoulos , thuật toán balloon model của Cohen, thuật toán Level Set của Leventon,... Với các dữ liệu có cấu trúc như các ảnh cắt lớp ta có thể dùng thuật toán Marching Square.

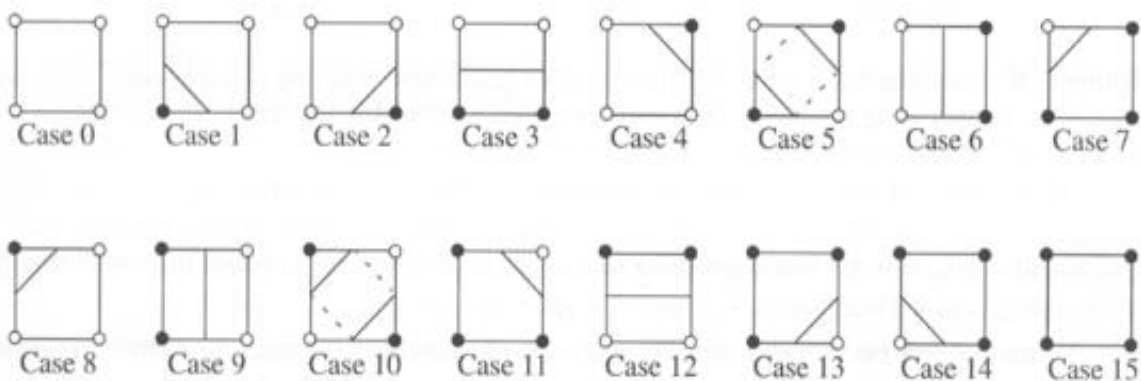
+ **Marching Square:** Sử dụng cho dữ liệu dạng lưới 2 chiều. Ý tưởng của phương pháp này là tạo ra một đường cong mô tả cho một giá trị vô hướng trong lưới dữ liệu, giá trị này gọi là isovalue.



**Hình 2.8 Minh họa thuật toán Marching square**

*Đường cong mô tả giá trị 5 trong lưới dữ liệu*

Đường nối giữa hai cạnh của một ô của lưới (cell) trong phương pháp này là đường thẳng. Giao điểm của đường nối này với các cạnh được tính bằng nội suy tuyến tính từ các giá trị ở các đỉnh nằm trên cạnh đó. Ta giả sử các đỉnh của ô sẽ nằm trong đường nối nếu giá trị tại đó lớn hơn giá trị isovalue và nằm ngoài nếu nhỏ hơn. Có  $2^4 = 16$  cách tạo ra các đường này.



**Hình 2.9 trường hợp Marching Square**

Các bước thực hiện Marching Square :

- + Chọn một ô.
- + Tính toán trạng thái trong,ngoài của mỗi đỉnh của ô.
- + Tìm “topological state” của ô để quyết định đường nối sẽ đi qua cạnh nào của

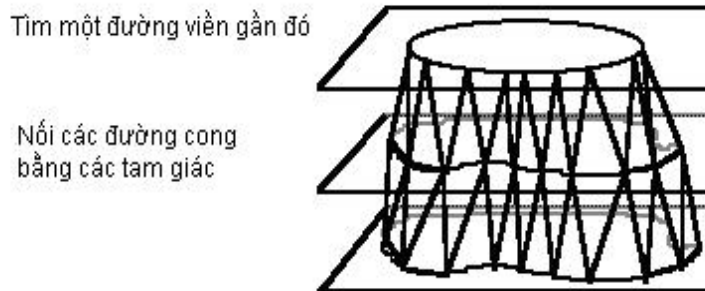
“cell”.

- + Tính toán giao đi ểm của các đường với các cạnh của ô .
- + Chuyển (march) tới ô khác.

Phương pháp Marching Square có ưu điểm là giúp tính toán nhanh nhưng nhược điểm là trong một số trường hợp ta có thể có nhiều cách tạo ra các đường đi qua “cell” và có thể tạo ra những lỗ.

+Tái tạo bề mặt: Sau khi đã xác định được các đường viền ta xây dựng một mặt từ các đường này. Các phương pháp được sử dụng hiện nay có phương pháp của Keppel (1975); phương pháp của Fush (1977): xây dựng một mặt giữa hai đường biên kề nhau; hay của Ekoule, Peyrin, Odet (1991).

Nối các lát cắt



**Hình 2.10 Minh họa tạo bề mặt từ các đường viền**

**- Tạo bề mặt từ dữ liệu khối (volume data, voxel based reconstruction):**

Trước hết cần sắp xếp lại dữ liệu thành dạng khối. Đối với các ảnh cắt lớp song song ta sẽ xếp các lát cắt liên tục nhau, xác định khoảng cách giữa các lát cắt, mỗi pixel trên các ảnh cắt lớp sẽ biến thành một voxel trong khối dữ liệu. Sau đó dùng các thuật toán để tạo bề mặt từ khối dữ liệu này. Các thuật toán được sử dụng phổ biến hiện nay là:

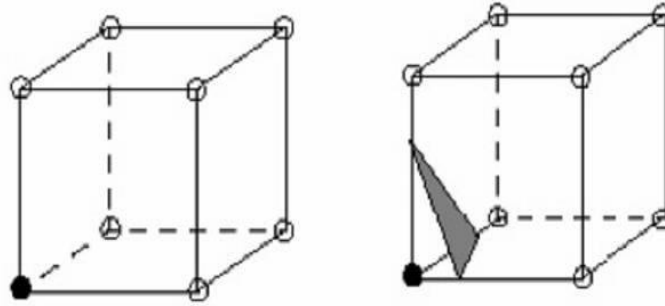
+ **Marching Cubes (MC)**: Thuật toán này được phát minh bởi William E. Lorensen và Harvey E. Cline và đã được cấp bản quyền sở hữu vào tại Mỹ vào ngày 5/6/1985. Theo quy định chung các tác giả được bảo hộ bản quyền trong 20 năm. Do đó, ở thời điểm này giấy phép đã hết hiệu lực và chúng ta có quyền tự do sử dụng thuật toán này cho các mục đích thương mại.

Thuật toán này tương tự như thuật toán Marching Square đã trình bày ở trên nhưng được thực hiện cho dữ liệu 3 chiều. Nguyên tắc của thuật toán này là chia khối dữ liệu thành các hình lập phương, mỗi hình lập phương được tạo từ 8 voxel nằm kề nhau. Sau đó xác



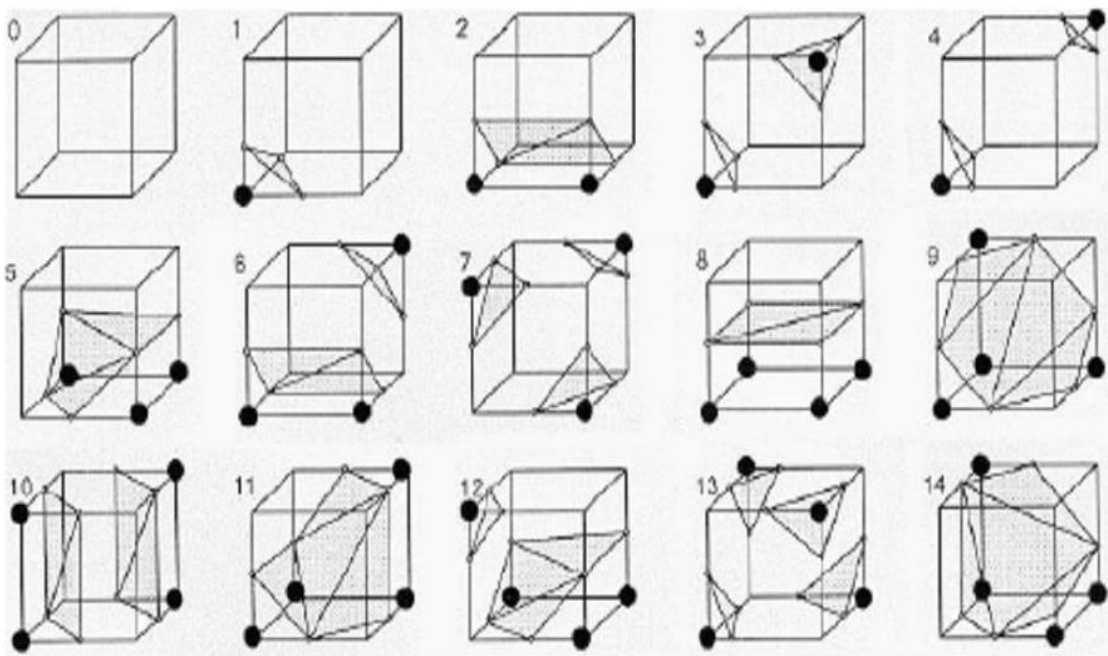
định một mặt đi xuyên qua mỗi hình lập phương, tính toán các véc tơ pháp tuyến, phát triển (march) đến hình lập phương tiếp theo. Từ đó ta có thể xấp xỉ một isosurface bởi một lưới tam giác (triangle mesh).

Ta xây dựng mặt phẳng này với giả thiết nếu giá trị tại đỉnh lớn hơn giá trị isovalue thì đỉnh đó nằm bên trong mặt và ngược lại. Để xác định giao điểm của mặt phẳng này với các cạnh của hình lập phương ta cần nội suy tuyến tính từ giá trị tại hai đỉnh trên cạnh đó.



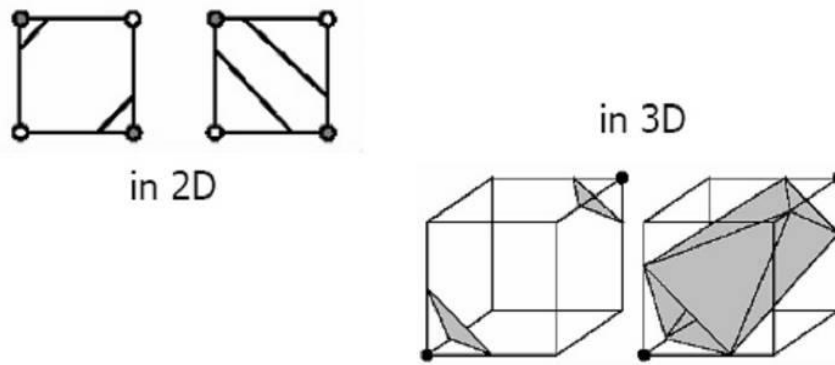
**Hình 2.11** Xây dựng bề mặt theo giá trị của các đỉnh

Hình lập phương có 8 đỉnh nên ta có tổng cộng  $2^8 = 256$  trường hợp mặt đi qua khối lập phương. Tuy nhiên do tính chất đối xứng của khối lập phương nên ta có thể giảm ước còn 15 trường hợp.



**Hình 2.12** Các trường hợp một mặt đi qua khối lập phương trong thuật toán Marching Cubes

Thuật toán MC tạo ảnh có độ phân giải cao. Tuy nhiên cũng như Marching Square, nhược điểm của thuật toán này là có thể tạo ra những lỗ (hole) trong bề mặt. Nguyên nhân là chúng ta có thể xây dựng những mặt khác nhau cho cùng một trường hợp (ambiguous face).

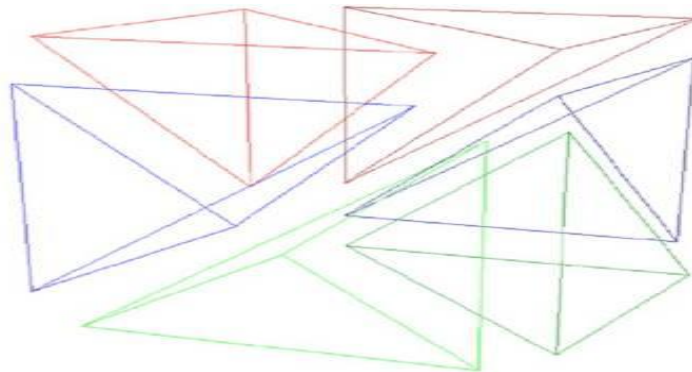


**Hình 2.13 Một trường hợp lỗi của Marching Cubes**

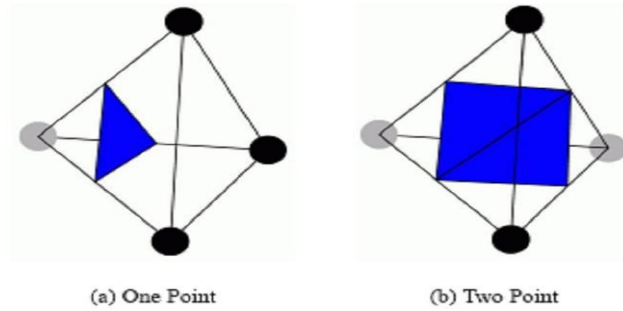
Tuy nhiên, có một số kỹ thuật được đưa ra để khắc phục điều này, ví dụ kỹ thuật Asymptotic Decider của G.M.Nielson và Bernd Hamann (Computer Science Arizona State University) đưa ra năm 1991 .

+ **Marching Tetrahedra (Marching Tetrahedrons, MT)** : đây cũng là một thuật toán được cải tiến từ thuật toán Marching Cubes. Thuật toán Marching Cubes xây dựng một mặt trong khối lập phương gồm 8 voxel kế cận, đơn vị thể tích này còn khá lớn. Để giảm các phép tính Doi, Guezie, Treece và một số tác giả khác đã chia khối lập phương thành các đơn vị thể tích nhỏ hơn là các khối tứ diện. Ví dụ Gauzie đã chia khối lập phương thành 5 khối tứ diện. Phương pháp này giúp bề mặt tạo ra khép kín hơn (tránh những trường hợp ambiguity) và định hướng hơn (to be closed and oriented). Ngoài lí do về kĩ thuật, các tác giả này phát triển thuật toán Marching Tetrahedra còn vì lí do thuật toán Marching Cubes được bảo hộ bản quyền.

Mặt khác phương pháp này sẽ tạo ra nhiều tam giác hơn phương pháp Marching Cubes và ta phải xác định cách chia khối lập phương thành các tứ diện.



**Hình 2.14 Chia khối lập phương thành các khối tứ diện**



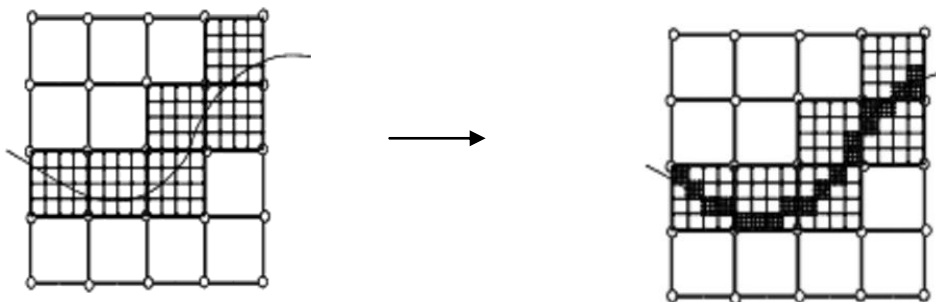
**Hình 2.15 Hai trường hợp mặt phẳng đi qua khối tứ diện trong thuật toán Marching Tetrahedra**

Cả hai thuật toán Marching Cubes và Marching Tetrahedra đều có nhược điểm là tạo ra nhiều đa giác (trong đó có nhiều đa giác không cần thiết) và poor aspect ratio. Trung bình trong thuật toán Marching Cubes mỗi khối lập phương sẽ tạo ra 3 tam giác. Vì vậy một tập dữ liệu  $32 \times 32 \times 16$  có thể tạo ra 3000 tam giác, một tập dữ liệu  $256 \times 256 \times 128$  có thể sinh ra tới 820000 tam giác. Có nhiều thuật toán đã được đưa ra nhằm làm giảm số đa giác này. Các thuật toán này được gọi là các thuật toán đơn giản hóa lưới (mesh simplification algorithm).

Các thuật toán MC, MT cũng không hiệu quả lắm đối với những tập dữ liệu lớn. Vì khi thực hiện với máy tính thì phép tìm kiếm trong các thuật toán này là phép tìm tuyến tính. Với những tập dữ liệu lớn ta phải có những kỹ thuật khác.

+ Dividing Cubes: Thuật toán này được đưa ra để giải quyết tình trạng thuật toán MC và MT thường tạo ra số đa giác nhiều hơn số pixel và có thể gây ra tình trạng “high rendering overhead”. Dividing Cubes được phát minh bởi Cline vào năm 1988 và được cấp bản quyền.

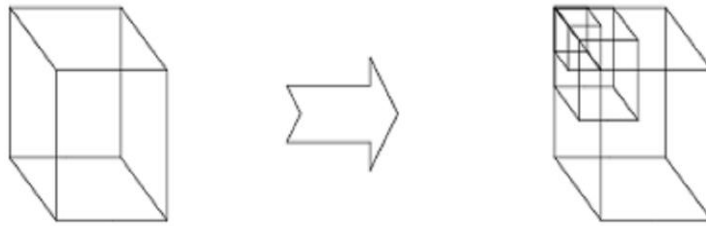
Ý tưởng của Dividing Cubes là không vẽ các đa giác mà chỉ vẽ các điểm. Để vẽ một đường ta tìm các pixel có giao với đường đó, sau đó chia nhỏ pixel này.



**Hình 2.16 Minh họa thuật toán Dividing Cubes để vẽ đường trong mặt phẳng**

Để vẽ mặt trong không gian 3 chiều trước hết ta cần tìm các voxel có giao với mặt,

sau đó chia nhỏ voxel. Tiếp tục tìm kiếm và chia nhỏ cho đến khi đạt đến độ phân giải cần thiết. Tìm những điểm giữa (mid - points) của voxel đó sau đó dùng phép chiếu để chuyển thành các pixel.



**Hình 2.17 Minh họa thuật toán Dividing Cubes trong không gian ba chiều**

Bề mặt tạo bởi kỹ thuật Dividing Cubes trông mịn hơn. Tuy nhiên, nếu các mẫu chia còn lớn hơn độ phân giải màn hình chúng ta sẽ nhìn thấy cấu trúc của bề mặt.

#### **Một số kĩ thuật tăng tốc cho SR:**

+ Surface Decimation : Tối ưu hóa số đa giác dùng để biểu diễn bề mặt như bỏ bớt đỉnh, cạnh. Đây cũng là một trong những thuật toán đơn giản hóa lưới.

+ Visibility Culling : Không biểu diễn những đa giác mà mắt không nhìn thấy.

+Parallel Rendering : Thực hiện công việc song song trên nhiều máy trạm.

Kĩ thuật SR thường được dùng với các ảnh có độ tương phản cao như CT hay MRI vì có thể xác định biên dễ dàng, các ảnh này cũng ít nhiễu. Với những ảnh có độ tương phản thấp và có độ nhiễu cao như siêu âm, PET, SPECT thì cần có các thuật toán lọc nhiễu và trích biên tốt.

### **2.3. Kỹ thuật Render Volume tương tác.**

#### **2.3.1 Thăm dò và tương tác miền kép**

#### **2.2.2 Dữ liệu thăm dò vật dụng**

#### **2.2.3 Phân loại vật dụng ( Classification Widgets)**

#### **2.2.4 Bảng chọn màu sắc.**

### **2.4 Kết luận chương 2**

Chương này đã giới thiệu một số kỹ thuật về hiển thị hình ảnh 3D từ các hình ảnh hình chiếu 2D như kỹ thuật Render Volume, kỹ thuật biểu diễn bề mặt. Ta đã đi chi tiết các kỹ thuật như trong kỹ thuật Render Volume ta đã tìm hiểu quy trình và phương pháp bố cục ảnh, trích biên... Trong chương tiếp theo ta sẽ đi thử nghiệm thực tế từ kỹ thuật Render Volume bằng chương trình kết hợp giữa lập trình và một số phần thư viện mở.

## CHƯƠNG 3 - CHƯƠNG TRÌNH THỬ NGHIỆM

*Trong chương này ta sẽ đi cài đặt thử nghiệm kỹ thuật hiển thị hình ảnh Render Volume từ các ảnh 2D của ảnh y tế DICOM*

### 3.1 Bài toán

Bài toán đầu vào là dãy các ảnh 2D là các lát cắt của cùng một đối tượng thu nhận bởi bằng máy chụp cắt lớp và được lưu dưới dạng chuẩn Dicom. Đầu ra là hình ảnh 3D của đối tượng ở các góc độ khác nhau. Phương pháp hiển thị sử dụng là kỹ thuật Render Volume.

### 3.2. Phân tích thiết kế

#### 3.2.1 Các công cụ

Trên cơ sở tìm hiểu nhiều bộ công cụ hỗ trợ lập trình đồ họa người viết đã chọn hai bộ công cụ VTK , ITK để xây dựng phần mềm của mình vì những lí do:

+ VTK,ITK là hai trong những bộ công cụ đ ược sử dụng nhiều nhất trên thế giới. Hầu hết các bài giảng về xử lí ảnh, đồ họa 3D và trực quan hóa của các trường đại học lớn tr ên thế giới đều lấy VTK v à ITK làm ví dụ minh họa, trên thế giới cũng có nhiều phần mềm hay được phát triển từ VTK và ITK.

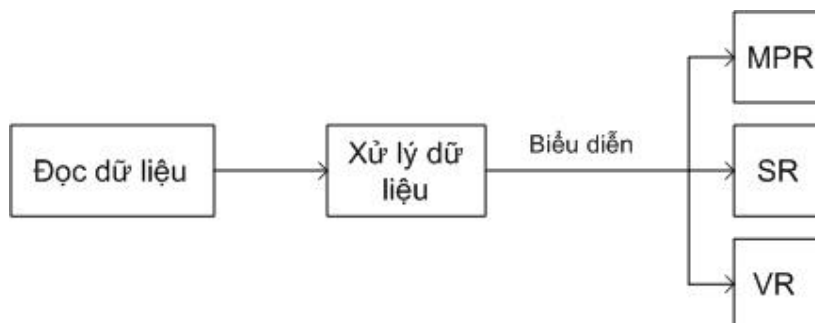
+ VTK, ITK đư ợc thiết kế rất chuyên nghiệp: hai bộ toolkit này được viết bằng C++ theo kiểu hướng đối tượng, có thể sử dụng trong nhiều ngôn ngữ khác như Java, Tcl, Python ,.NET ,...; hai bộ toolkit này có mã nguồn mở và liên tục được nâng cấp.

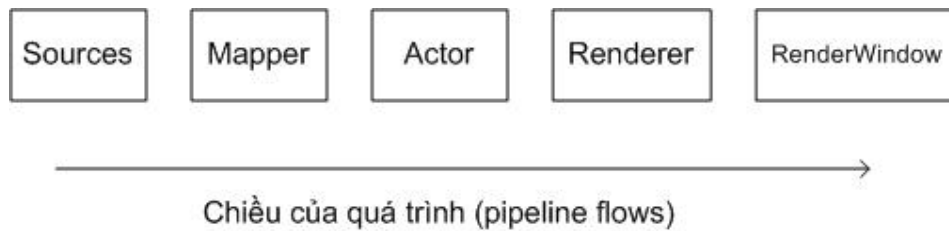
+VTK, ITK có nhi ều hỗ trợ cho xử lí ảnh y tế.

#### 3.2.2 Chương trình cài đặt

##### ❖Pipeline

Pipeline là chuỗi các bước xử lý dữ liệu và hiển thị thành hình ảnh. Pipeline của chương trình cài đặt như sau



**Hình 3.7 Pipeline của chương trình cài đặt****Hình 3.8 Chu trình biểu diễn dữ liệu thành hình ảnh**

### 3.3. Một số kết quả chương trình

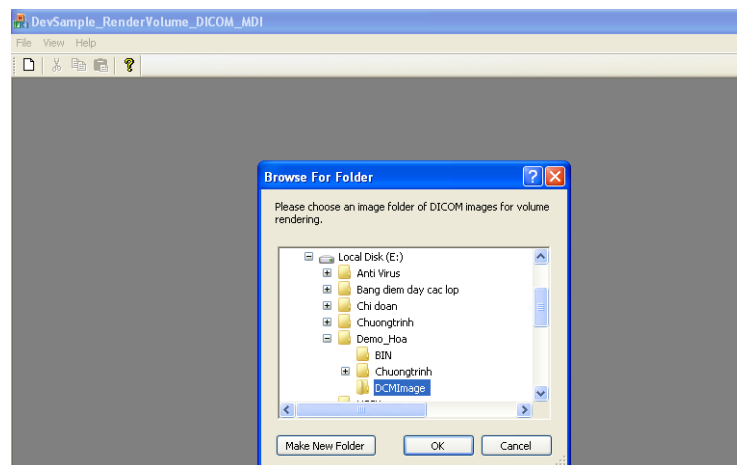
Chương trình được viết theo kiểu dialog -based, được thiết kế gồm có một cửa sổ chính và một số trang (tab). Dữ liệu đầu vào của chương trình là tập hợp các lát cắt được lưu theo định dạng DICOM.

Chương trình cho phép người sử dụng xem kết quả ảnh ba chiều dưới các góc độ khác nhau bằng cách dùng chuột. Người sử dụng có thể copy màn hình vào clipboard để đưa vào các chương trình khác, lưu màn hình dưới dạng ảnh JPEG hoặc xuất thành một trang HTML.

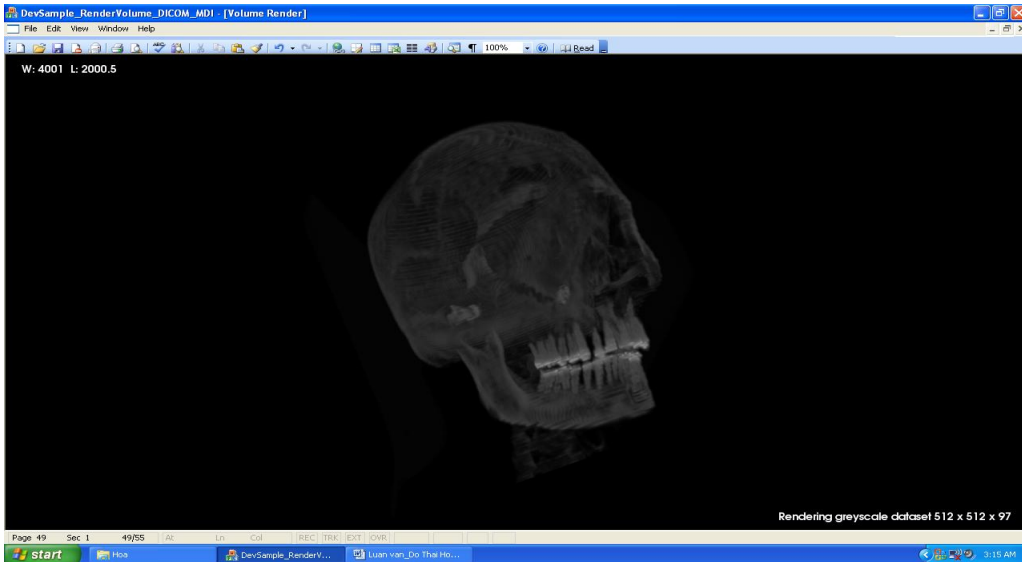
- Một số ảnh chạy chương trình

Các hình ảnh sau là kết quả thử nghiệm với dữ liệu CT demo .

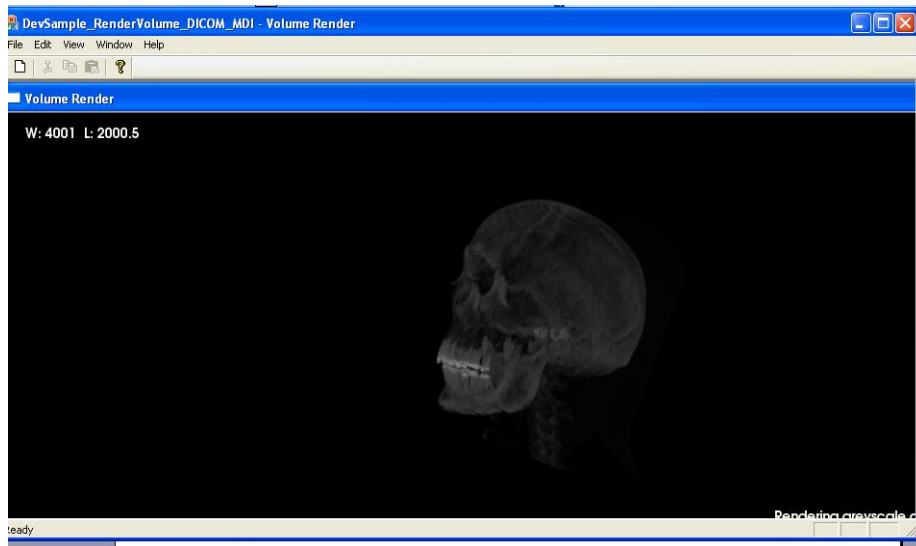
- Cửa sổ chính và trang duyệt tập ảnh Dicom

**Hình 3.9 Giao diện màn hình chính**

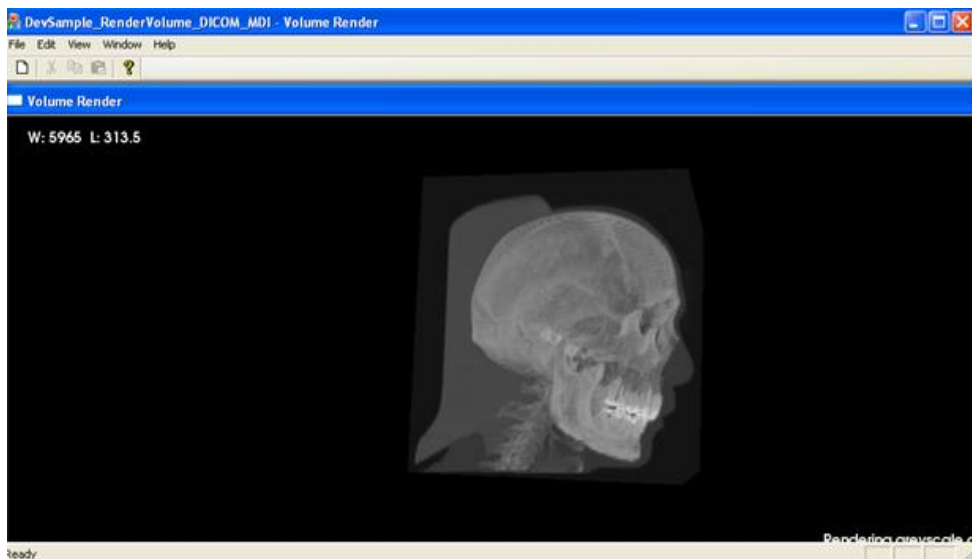
- Trang Render volume : Kết quả biểu diễn theo bề mặt. Trong trang này người sử dụng có thể lựa chọn biểu diễn các góc cạnh của hình ảnh 3D được hiển thị.



**Hình 3.10 Hình ảnh 3D được hiển thị**



**Hình 3.11 Các góc độ khác nhau của hình ảnh 3D được hiển thị**



**Hình 3.12 Hiển thị thêm mô mềm**

- Nhận xét:

Chương trình được xây dựng bằng Visual C++ và được đóng gói, do đó có thể cài đặt và sử dụng chương trình trên các máy tính có hệ điều hành Windows 98/2000/XP mà không cần nền (platform) nào khác. Chương trình phải làm việc với lượng dữ liệu lớn nên cấu hình máy phải tương đối mạnh. Tốc độ CPU nên từ 1,8GHz trở lên, RAM từ 256MB trở lên.

Giao diện chương trình được thiết kế thân thiện với người sử dụng, các bộ phận được bố trí tương tự các phần mềm đang được sử dụng trong thực tế. Dữ liệu vào của chương trình là kiểu dữ liệu thường gặp trong thực tế: tập ảnh theo định dạng DICOM. Chương trình đã chạy thử nghiệm với các tập ảnh dữ liệu DICOM của bệnh viện.

Chương trình cài đặt đã minh họa được các kỹ thuật tái tạo ảnh ba chiều từ lát cắt song song. Chất lượng hình ảnh tốt đối với dữ liệu CT.

### **3.4 Kết luận chương 3**

Trong chương này ta đã đi cài đặt thử nghiệm kỹ thuật hiển thị hình ảnh Render Volumme từ các ảnh 2D của ảnh y tế DICOM và đưa ra được những điểm đạt được và chưa đạt được của chương trình cài đặt.

## **KẾT LUẬN**

### **Những việc đã thực hiện được :**

- Tìm hiểu những vấn đề chung liên quan đến việc tái tạo hình ảnh ba chiều của các vật thể từ các hình chiếu. Tìm hiểu tình hình sử dụng các phần mềm xử lý hình ảnh trong y tế tại Việt Nam hiện nay.
- Tìm hiểu nguyên lý tái tạo ảnh ba chiều từ các ảnh cắt lớp nói chung và các ảnh cắt lớp song song nói riêng.
- Tìm hiểu các vấn đề liên quan đến kỹ thuật tái tạo ảnh nổi. Nắm được kỹ thuật tương ứng theo cường độ.
- Tìm hiểu một số lĩnh vực có thể ứng dụng kỹ thuật tái tạo nổi và một số đề tài đã và đang được nghiên cứu trên thế giới.
- Viết các phần mềm mang tính minh họa, có thể được sử dụng phục vụ đào tạo, nghiên cứu và phát triển các phần mềm ứng dụng trong tương lai.

### **Những hạn chế :**

Những nghiên cứu mới tập trung vào lý thuyết và phần mềm, chưa được tích hợp với các phần cứng.

Do thời gian nghiên cứu có hạn nên người viết chưa nắm một cách đầy đủ và hệ thống cơ sở lý thuyết của kỹ thuật stereo matching, đặc biệt là về vấn đề hiệu chỉnh camera



(camera calibration) để có thể phát triển tối ưu cho phần mềm ứng dụng, cũng như vấn đề tích hợp hỗ trợ phần cứng.

### **Hướng phát triển**

- Đối với vấn đề tái tạo ảnh ba chiều từ các lát cắt song song :

Hướng phát triển là bổ sung thêm các chức năng, tiếp tục tìm hiểu các bộ toolkit VTK và ITK. Tiếp tục tìm hiểu thực tế để phát triển thành một phần mềm có khả năng sử dụng tại Việt Nam.

Tiếp tục nghiên cứu các phương pháp tái tạo ảnh ba chiều với các dạng ảnh cắt lớp khác như cắt xuyên tâm, cắt tự do. Nghiên cứu đặc tính của các loại ảnh cắt lớp của siêu âm, MRI, hình ảnh y học hạt nhân từ đó có những phương án tái tạo tốt nhất cho mỗi loại.