

WORKSHOP REPORT

Nguyễn Huỳnh Hải Đăng

MSSV: 22127052

nhhdang22@clc.fitus.edu.vn

Lưu Nam Đạt

MSSV: 22127062

ln-dat22@clc.fitus.edu.vn

Nguyễn Bá Công

MSSV: 22127046

nbcong22@clc.fitus.edu.vn

Đặng Trần Anh Khoa

MSSV: 22127024

dtakhoa22@clc.fitus.edu.vn

LỜI GIỚI THIỆU

Workshop Xử lý ảnh y khoa đã được Trường Đại học Khoa học tự nhiên, ĐHQG-HCM và Viện Công nghệ Kyoto, Nhật Bản (Kyoto Institute of Technology) đồng tổ chức từ ngày 24-26/02/2025. Đây là báo cáo cho Workshop, do nhóm Orz thực hiện.

MỤC LỤC

1 CT	2
1.1 Khái niệm	2
1.2 Lịch sử	2
2 MRI	3
2.1 Khái niệm	3
2.2 Nguyên tắc	3
2.3 Các chế độ hình ảnh	3
3 Định lý Phép chiếu Lát cắt (Projection Slice Theorem)	4
4 Kiến trúc chương trình Slicer	5
5 DICOM	5
6 Mô hình dữ liệu Slicer	8
7 Chẩn đoán hỗ trợ bởi máy tính	9
7.1 Quy trình hoạt động của CAD	10
7.2 Vai trò của CAD trong y học	10
7.3 Một số trở ngại của CAD:	10
7.4 Các thách thức cho CAD	10
7.5 Quá trình bùng nổ AI và tác động đến CAD	11
8 Phòng thí nghiệm chăm sóc sức khỏe kỹ thuật số (Digital Healthcare Lab)	11
8.1 Thực tế ảo (VR) - Thực tế tăng cường (AR) trong chăm sóc sức khỏe	12

8.2 Ứng dụng Gamification trong trị liệu	12
8.3 AI Dự đoán trong chăm sóc sức khỏe	13
9 Fast Marching Algorithm	13
9.1 Phương trình Eikonal	13
9.2 Xấp xỉ phương trình Eikonal	14
9.3 Giải thuật	15
9.4 Ứng dụng trong Phân tách mạch máu	17
9.4.1 Thực nghiệm trên 3D Slicer	19
Tham khảo	20
A Phụ lục	20

1 CT

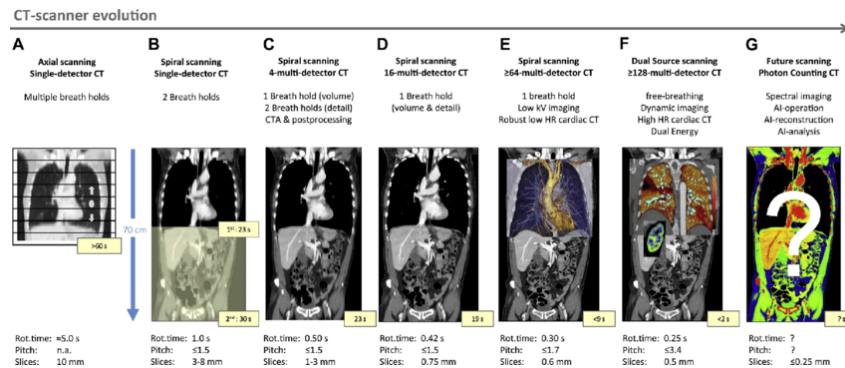
1.1 KHÁI NIỆM

Chụp cắt lớp vi tính, còn gọi là chụp CT (Computerized Tomography), là kỹ thuật kết hợp máy tính và tia X để tạo ra hình ảnh các cơ quan, xương và các mô của cơ thể theo lát cắt ngang, cho phép phóng 1 chùm tia X liên tục qua cơ thể. Đây là một kỹ thuật không xâm lấn.

Hình ảnh thu được sẽ được máy tính xử lý, tái tạo để cho ra cấu trúc bên trong. Độ phân giải (hay độ sắc nét) hình ảnh tùy thuộc vào thể hệ máy chụp là 4, 16, hay 64 lát cắt. Nếu máy chụp có số lát cắt càng cao thì hình ảnh càng rõ nét. Tùy theo độ cần tia X, các tổn thương ở những bộ phận bên trong cơ thể được diễn tả bằng cách tăng hoặc giảm độ đậm.

Trong quá trình chụp CT, người bệnh được chỉ định nằm trên một chiếc máy, khi ấy máy dò và ống tia X sẽ xoay xung quanh, mỗi vòng quay sẽ cho ra một hình ảnh lát mỏng của cơ thể. Sau đó, những hình ảnh này được gửi đến máy tính, nơi chúng được kết hợp để tạo ra hình ảnh các lát cắt hoặc mặt cắt ngang của cơ thể. Quá trình này không mất nhiều thời gian và cũng không gây đau đớn cho người bệnh.

1.2 LỊCH SỬ



Hình 1: Quá trình phát triển của máy chụp CT.

Năm 1917, Định lý Radon ra đời, đóng vai trò quan trọng trong việc phát triển thuật toán tái tạo ảnh của chụp CT sau này.

Giai đoạn sơ khai (1971 - 1985):

- 1971: Hệ thống chụp CT đầu tiên được phát minh, tập trung vào quét não (**Head CT**).
- 1974: Công nghệ mở rộng ra để chụp toàn bộ cơ thể (**Body CT**), đánh dấu bước tiến quan trọng trong chẩn đoán y khoa.

- 1975: Sự ra đời của máy CT thế hệ thứ ba (**3rd Generation CT**), giúp tăng tốc độ chụp ảnh.

Giai đoạn cải tiến công nghệ (1987 - 2000):

- 1987: Phát triển công nghệ **Solid State Detectors** (Bộ dò trạng thái rắn), giúp tăng cường độ nhạy và cải thiện chất lượng hình ảnh.
- 1990: Ứng dụng công nghệ **Slip Ring**, giúp chụp xoắn ốc (Helical CT), mở đường cho chụp ảnh 3D tốc độ cao.
- 1993: Công nghệ CT có khả năng chụp 4 hoặc 8 lát cắt (4/8-Slice CT), tăng đáng kể độ chi tiết của hình ảnh.
- 1995: Tốc độ quét giảm xuống còn 1.0 giây/lần chụp.

Giai đoạn phát triển mạnh mẽ (2000 - 2010):

- 2000: Cải tiến tốc độ quét còn 0.8 giây/lần, đồng thời áp dụng phương pháp giảm liều phóng xạ **AEC Dose Reduction**.
- 2002: Ra mắt CT 16 lát cắt (**16-Slice CT**), cải thiện đáng kể khả năng phân giải không gian.
- 2005: Tiếp tục giảm tốc độ quét xuống còn 0.5 giây/lần, giúp chụp nhanh hơn.
- 2007: CT 32/64 lát cắt (**32/64-Slice CT**) xuất hiện, hỗ trợ chẩn đoán chi tiết hơn, đặc biệt là trong tim mạch.
- 2008: Phát triển công nghệ **Spectral Imaging** và **ASiR** (Adaptive Statistical Iterative Reconstruction), giúp giảm nhiễu ảnh và tối ưu hóa chất lượng hình ảnh.

Giai đoạn hiện đại:

- 2010: Công nghệ CT có độ phân giải cao (Hi-Res CT), cung cấp hình ảnh sắc nét hơn.
- 2011: Giới thiệu **Veo Iterative Reconstruction**, một thuật toán tái tạo ảnh tiên tiến giúp giảm liều phóng xạ nhưng vẫn giữ được chất lượng hình ảnh cao.
- 2012: Xuất hiện công nghệ **Motion Correction**, giúp cải thiện hình ảnh của các bộ phận di chuyển như tim và phổi.

2 MRI

2.1 KHÁI NIỆM

Chụp cộng hưởng từ hay còn gọi là chụp MRI (Magnetic Resonance Imaging) là phương pháp sử dụng từ trường mạnh, sóng vô tuyến và máy tính để phác họa hình ảnh chi tiết bên trong cơ thể con người. Đây là một kỹ thuật không xâm lấn. Không giống như chụp X-quang và chụp cắt lớp vi tính (CT), MRI không sử dụng bức xạ ion hóa gây hại của tia X.

Máy MRI hoạt động bằng cách tạo ra một từ trường mạnh bên trong cơ thể. Một máy tính lấy các tín hiệu từ MRI để tạo ra hàng loạt ảnh, mỗi bức ảnh cho thấy một phần mỏng của cơ thể.

2.2 NGUYÊN TẮC

- Một từ trường mạnh được tạo ra bởi máy MRI, làm cho các hạt nhân hydrogen trong cơ thể bệnh nhân (các proton) thẳng hàng theo hướng của từ trường.
- Các hạt nhân hydrogen bắt đầu quay quanh hướng từ trường chính với một tần số gọi là tần số Larmor, liên quan đến cường độ từ trường và đặc tính của hạt nhân.
- Máy MRI phát một xung radio tại tần số Larmor, làm các hạt nhân hydrogen bị kích thích và lệch khỏi hướng ban đầu của từ trường.
- Khi xung radio ngừng, các hạt nhân hydrogen trở về trạng thái ban đầu. Trong quá trình này, chúng phát ra tín hiệu mà máy MRI thu thập.
- Các tín hiệu thu thập được từ cơ thể bệnh nhân được máy tính xử lý để tạo ra hình ảnh 2D hoặc 3D của các mô mềm.

2.3 CÁC CHẾ ĐỘ HÌNH ẢNH

1. T1-Weighted (T1W) Imaging

Đặc điểm:

- Chụp với thời gian lặp ngắn (TR) và thời gian vang ngắn (TE).
- Rất nhạy với mỡ.

Nhìn chung:

- Dịch não tủy (CSF): Rất đen.
- Mỡ: Trắng.
- Chất xám: Xám đậm.
- Chất trắng: Xám sáng.
- Ứng dụng: Tốt để quan sát các cấu trúc giải phẫu.

2. T2-Weighted (T2W) Imaging

Đặc điểm:

- Chụp với thời gian lặp dài (TR-Repetition) và thời gian vang dài (TE-Echo time).
- Rất nhạy với dịch não tủy (CSF).

Nhìn chung:

- CSF: Rất trắng.
 - Mỡ: Xám.
 - Chất xám: Xám sáng.
 - Chất trắng: Xám đậm.
 - Ứng dụng: Phát hiện tổn thương, phù nề, hoặc các vấn đề liên quan đến nước.
3. Các chế độ nâng cao khác
- FLAIR (Fluid-Attenuated Inversion Recovery): Loại bỏ tín hiệu từ CSF, giúp làm nổi bật các tổn thương.
 - DWI (Diffusion-Weighted Imaging): Tập trung vào sự khuếch tán của nước, hữu ích trong chẩn đoán đột quỵ.
 - DTI (Diffusion Tensor Imaging): Phân tích các đường dẫn của chất trắng, thường dùng trong nghiên cứu thần kinh.

3 ĐỊNH LÝ PHÉP CHIẾU LÁT CẮT (PROJECTION SLICE THEOREM)

Đây là một định lý quan trọng trong Xử lý ảnh y khoa và Tomography.

Cho $f(x, y)$ là ảnh 2 chiều.

Ta định nghĩa phép biến đổi Fourier 2 chiều của nó:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (1)$$

Giả sử ta lấy một phép chiếu theo hướng θ , tức là tích lũy cường độ theo trục vuông góc với hướng đó:

$$P_{\theta}(r) = \int_{-\infty}^{\infty} f(r \cos \theta - s \sin \theta, r \sin \theta + s \cos \theta) ds \quad (2)$$

Đây là phép chiếu Radon, cho ta một ảnh 1D (1 chiều) biểu diễn cách ảnh gốc lan truyền theo hướng θ .

Định lý Phép chiếu Lát cắt khẳng định rằng biến đổi Fourier 1 chiều của phép chiếu

$$P_{\theta}(r) \quad (3)$$

chính là một lát cắt của biến đổi Fourier 2 chiều $F(u, v)$ dọc theo đường thẳng có góc θ trong miền tần số:

$$P_{\theta}(k) = F(k \cos \theta, k \sin \theta) \quad (4)$$

4 KIẾN TRÚC CHƯƠNG TRÌNH SLICER

1. Lõi ứng dụng Slicer (Slicer Application Core):

Đây là phần trung tâm của ứng dụng, nơi quản lý hoạt động chính của hệ thống.

2. Màn hình quan sát chính MRML (MRML Scene):

- Màn hình quan sát chính này tương tác trực tiếp với các mô-đun (Modules) khác nhau.
- Là nơi chứa và tổ chức tất cả các nút (Nodes), cũng như quản lý dữ liệu và mối quan hệ giữa chúng.

3. Các Mô-đun (Modules):

- Bao gồm nhiều mô-đun từ cơ bản (Segment Editor, Transforms, Registration) đến nâng cao (DICOM, Radiomics, SlicerIGT)
- Các mô-đun này thực hiện các chức năng chuyên biệt, ví dụ: phân đoạn, đăng ký hình ảnh hoặc phân tích.

Chúng tương tác với màn hình quan sát chính MRML để đọc và ghi dữ liệu.

Một kiểu mô-đun là C++ loadable:

- Được viết bằng C++
- Có thể truy cập đầy đủ API Slicer
- Hữu ích trong thực thi các thành phần/cơ sở hạ tầng ứng dụng phức tạp, hiệu suất quan trọng

4. Cách tạo một mô-đun tùy chỉnh:

- Chọn mục Module Selection >> Developer Tools >> Extension Wizard.
- Chọn Create Extension. Trong một tiện ích mở rộng, có thể có nhiều mô-đun khác nhau. Đặt tên và phân loại của tiện ích.
- Thêm một mô-đun mới trong tiện ích.
- Cấu trúc của tiện ích có thể được quan sát trong Extension Editor. Tính chất của mỗi module có thể được chỉnh sửa bằng file .py trong tiện ích.
- Trên thanh Module Selection, chọn mô-đun đã tự cài đặt bằng cách chọn phân loại và tên của tiện ích.

5. Tiện ích mở rộng (Extensions):

- Bao gồm các mô-đun mở rộng như SlicerIGT, SlicerRT, SlicerIGSO.
- Các tiện ích này mở rộng thêm chức năng của chương trình, phù hợp với các nhu cầu cụ thể.

6. Thư viện tích hợp (Libraries):

Chương trình sử dụng các thư viện như Qt, VTK, ITK, CTK, Python, numpy, v.v., để xử lý giao diện, hiển thị hình ảnh, tính toán, và nhiều chức năng khác.

5 DICOM

Hình ảnh kỹ thuật số và truyền thông trong y học (Digital Imaging and Communications in Medicine - DICOM) là một tiêu chuẩn được sử dụng rộng rãi để trao đổi thông tin về X quang kỹ thuật số. Trong hầu hết các trường hợp, thiết bị hình ảnh (máy quét CT và MR) được sử dụng trong bệnh viện sẽ tạo ra hình ảnh được lưu dưới dạng đối tượng DICOM.

DICOM sắp xếp dữ liệu theo thứ bậc

- Bệnh nhân ... có thể có 1 hoặc nhiều

- Nghiên cứu (lần khám chụp ảnh đơn) ... có thể có 1 hoặc nhiều
 - Chuỗi (thu thập hình ảnh đơn, thường tương ứng với một khối lượng hình ảnh đơn) ... có thể có 1 hoặc nhiều
 - Trường hợp (thường là mỗi Chuỗi sẽ chứa nhiều Trường hợp, với mỗi Trường hợp tương ứng với một lát cắt duy nhất của hình ảnh)

Kết quả của quá trình chụp ảnh, thiết bị chụp ảnh sẽ tạo ra các tệp DICOM, trong đó mỗi tệp tương ứng với một Trường hợp và được gắn thẻ thông tin cho phép xác định thông tin về Loạt, Nghiên cứu và Bệnh nhân để đưa vào đúng vị trí trong hệ thống phân cấp.

Có nhiều đối tượng DICOM được định nghĩa theo tiêu chuẩn. Các loại đối tượng phổ biến nhất là các đối tượng lưu trữ khối lượng hình ảnh do máy quét CT và MR tạo ra. Những đối tượng đó thường có nhiều tệp (thể hiện) cho mỗi chuỗi. Các tác vụ xử lý hình ảnh thường liên quan đến việc phân tích toàn bộ khối lượng hình ảnh, thường tương ứng với một Chuỗi duy nhất.

Cơ sở dữ liệu DICOM của Slicer

Để sắp xếp dữ liệu và cho phép truy cập nhanh hơn, Slicer lưu trữ Cơ sở dữ liệu DICOM cục bộ chứa các bản sao (hoặc liên kết đến) tệp DICOM và thông tin cơ bản về nội dung của từng tệp. Có thể có nhiều cơ sở dữ liệu trên máy tính cùng một lúc và chuyển đổi giữa chúng nếu, ví dụ, chúng bao gồm dữ liệu từ các dự án nghiên cứu khác nhau. Mỗi cơ sở dữ liệu chỉ đơn giản là một thư mục trên đĩa cục bộ có một vài tệp SQLite và thư mục con để lưu trữ dữ liệu hình ảnh. Không sửa đổi thủ công nội dung của các thư mục này. Dữ liệu DICOM có thể nhập vào cơ sở dữ liệu thông qua nhập tệp hoặc thông qua truyền mạng DICOM. Các mô-đun Slicer cũng có thể điền kết quả tính toán vào cơ sở dữ liệu DICOM.

Lưu ý rằng tiêu chuẩn DICOM không chỉ định cách các tệp sẽ được sắp xếp trên đĩa, vì vậy nếu người dùng có dữ liệu DICOM từ CDROM hoặc được chuyển từ máy quét, nhìn chung không thể biết bất cứ điều gì về nội dung từ tên tệp hoặc thư mục. Tuy nhiên, sau khi dữ liệu được nhập vào cơ sở dữ liệu, nó sẽ được sắp xếp theo phân cấp Bệnh nhân/ Nghiên cứu/Loạt tiêu chuẩn DICOM.

Các plugin DICOM

Một chức năng chính của mô-đun DICOM là ánh xạ từ tổ chức dữ liệu thu thập thành biểu diễn khối lượng. Nghĩa là, các tệp DICOM thường mô tả các thuộc tính của quá trình chụp ảnh, như trình tự các vị trí của bảng trong quá trình thu thập CT, trong khi Slicer hoạt động trên các khối lượng ảnh có các pixel cách đều nhau. Ví dụ, nếu tốc độ chuyển động của bàn không nhất quán trong quá trình thu thập (có thể xảy ra đối với một số lần quét 'bolus chase' tương phản), mô-đun DICOM của Slicer sẽ cảnh báo người dùng rằng hình học thu thập không nhất quán và người dùng nên thận trọng khi diễn giải kết quả phân tích như phép đo.

Điều này có nghĩa là Slicer thường có thể gợi ý nhiều cách diễn giải dữ liệu (chẳng hạn như đọc tệp DICOM dưới dạng tập dữ liệu khuếch tán hoặc dưới dạng thể tích vô hướng. Khi có thể tính toán bằng cách kiểm tra các tệp, mô-đun DICOM sẽ chọn tùy chọn diễn giải có khả năng xảy ra nhất theo mặc định. Kể từ bản phát hành này, các plugin tiêu chuẩn bao gồm thể tích vô hướng và thể tích khuếch tán, trong khi các tiện ích mở rộng có sẵn cho các đối tượng phân đoạn, dữ liệu RT và dữ liệu PET/CT. Dự kiến sẽ có nhiều plugin hơn cho các phiên bản trong tương lai. Mục tiêu dài hạn là có thể biểu diễn hầu hết, nếu không muốn nói là toàn bộ, dữ liệu của Slicer trong các đối tượng dữ liệu DICOM tương ứng khi tiêu chuẩn phát triển để hỗ trợ chúng.

Cách tạo cơ sở dữ liệu DICOM Tạo cơ sở dữ liệu DICOM là điều kiện tiên quyết cho tất cả các hoạt động DICOM. Khi mô-đun DICOM được mở lần đầu tiên, Slicer sẽ tự động tạo cơ sở dữ liệu mới. Người dùng có thể chọn tạo cơ sở dữ liệu mới hoặc mở cơ sở dữ liệu đã tạo trước đó.

Người dùng có thể mở cơ sở dữ liệu ở vị trí khác bất kỳ lúc nào trong bảng điều khiển mô-đun DICOM / Cài đặt cơ sở dữ liệu DICOM / Vị trí cơ sở dữ liệu.

Đọc các tệp DICOM vào màn hình quan sát chính Vì các tệp DICOM thường nằm trong nhiều thư mục, chúng có thể tham chiếu chéo lẫn nhau và thường có thể được diễn giải theo nhiều cách khác nhau, nên việc đọc các tệp DICOM vào màn hình quan sát chính được thực hiện theo hai bước riêng biệt: nhập (lập chỉ mục các tệp để có thể hiển thị chúng trong trình duyệt cơ sở dữ liệu DICOM) và tải (hiển thị các mục DICOM đã chọn trong màn hình quan sát chính Slicer).

Nhập DICOM

- Đảm bảo rằng tất cả các tiện ích mở rộng Slicer bắt buộc đều được cài đặt. Lỗi Slicer chứa plugin nhập DICOM để nhập hình ảnh, nhưng có thể cần thêm các tiện ích mở rộng khác cho các đối tượng thông tin khác. Ví dụ, tiện ích mở rộng SlicerRT là cần thiết để nhập/xuất các đối tượng thông tin về xạ trị (bộ cấu trúc RT, liều lượng, hình ảnh, kế hoạch). Tiện ích mở rộng báo cáo định lượng là cần thiết để nhập/xuất các đối tượng phân đoạn DICOM, báo cáo có cấu trúc và bản đồ tham số.
- Chọn mô-đun của DICOM
- Chọn các thư mục chứa các tệp của DICOM
- Kéo thư mục chứa các tệp DICOM vào cửa sổ chương trình Slicer hoặc chọn nút “Import” ở góc trên bên trái ở trình duyệt DICOM rồi chọn thư mục chứa các tệp DICOM.

Nút nhập có một menu thả xuống, có thể hiển thị bằng cách nhấp vào nút mũi tên xuống nhỏ ở bên phải. Hiện tại, mục duy nhất trong menu là tùy chọn “Sao chép tệp đã nhập vào cơ sở dữ liệu DICOM” (“Copy imported files to DICOM database”). Bật tùy chọn này để sao chép tất cả các tệp DICOM đã nhập từ vị trí gốc của chúng vào cơ sở dữ liệu DICOM của Slicer. Điều này hữu ích khi tải dữ liệu từ phương tiện lưu trữ có thể tháo rời (ổ đĩa CD/DVD/USB hoặc ổ đĩa từ xa), để có thể tải bộ dữ liệu ngay cả sau khi phương tiện đã được đẩy ra. Nếu tùy chọn sao chép bị tắt thì chỉ đường dẫn của các tệp đã nhập sẽ được lưu trữ trong cơ sở dữ liệu DICOM (cùng với các giá trị của các thẻ DICOM thường dùng nhất).

DICOM loading

Đi đến mô-đun DICOM. Nhấp vào “Show DICOM database” nếu cửa sổ cơ sở dữ liệu DICOM chưa hiển thị (nó hiển thị danh sách bệnh nhân, nghiên cứu và loạt).

Nhấp đúp vào bệnh nhân, nghiên cứu hoặc loạt để tải.

Nhấp vào nút “Show DICOM database” để chuyển đổi giữa trình duyệt cơ sở dữ liệu (để tải thêm dữ liệu) và trình xem (để xem những gì đã được tải vào màn hình quan sát chính)

Xóa dữ liệu khỏi cơ sở dữ liệu DICOM

Bằng cách nhấp chuột phải vào Bệnh nhân, Nghiên cứu hoặc Series, bạn có thể xóa mục nhập khỏi cơ sở dữ liệu DICOM. Lưu ý rằng để tránh mất dữ liệu vô tình, Slicer không xóa các tệp dữ liệu hình ảnh tương ứng nếu chỉ có liên kết của chúng được thêm vào cơ sở dữ liệu. Các tệp DICOM được sao chép vào cơ sở dữ liệu DICOM sẽ bị xóa khỏi cơ sở dữ liệu.

Xuất dữ liệu từ màn hình quan sát chính đến cơ sở dữ liệu DICOM Dữ liệu trong màn hình quan sát chính có thể được xuất sang định dạng DICOM, để lưu trữ trong cơ sở dữ liệu DICOM hoặc xuất sang các tệp DICOM:

1. Đảm bảo rằng tất cả các tiện ích mở rộng Slicer bắt buộc đều được cài đặt. Lỗi Slicer chứa plugin xuất DICOM để xuất hình ảnh, nhưng có thể cần thêm các tiện ích mở rộng khác cho các đối tượng thông tin khác.
 - Cần có tiện ích mở rộng “SlicerRT” để nhập/xuất các đối tượng thông tin xạ trị: bộ cấu trúc RT, liều RT, hình ảnh RT, kế hoạch RT.
 - Cần có tiện ích mở rộng “Quantitative reporting” để nhập/xuất các đối tượng phân đoạn DICOM, báo cáo có cấu trúc và bản đồ tham số.
2. Đi tới Mô-đun dữ liệu hoặc Mô-đun DICOM.

3. Nhấp chuột phải vào một nút dữ liệu trong cây dữ liệu sẽ được chuyển đổi sang định dạng DICOM.
4. Chọn loại xuất ở góc dưới bên trái của hộp thoại xuất. Điều này là cần thiết vì có thể có một số đối tượng thông tin DICOM có thể lưu trữ cùng một loại dữ liệu. Ví dụ: phân đoạn có thể được lưu trữ dưới dạng đối tượng phân đoạn DICOM (DICOM hiện đại) hoặc bộ cấu trúc RT (biểu diễn cũ, chủ yếu được sử dụng trong lập kế hoạch điều trị bức xạ).
 - Loại xuất “Gói dữ liệu Slicer” ghi toàn bộ màn hình quan sát chính vào định dạng DICOM bằng cách đóng gói gói MRB cảnh bên trong tệp DICOM. Kết quả là đối tượng chụp thứ cấp DICOM, có thể được lưu trữ trong bất kỳ hệ thống lưu trữ DICOM nào. Thông tin chụp thứ cấp này lưu trữ tất cả các chi tiết của màn hình quan sát chính nhưng chỉ có 3D Slicer mới có thể diễn giải dữ liệu.
 - Loại xuất: Sau khi người dùng chọn một nút, các plugin DICOM sẽ tạo các tệp có thể xuất cho chuỗi mà chúng có thể xuất. Danh sách kết quả sẽ xuất hiện trong phần này, được nhóm theo plugin. Số tin cậy sẽ là số tin cậy trung bình của từng chuỗi riêng biệt cho plugin đó.
5. Tùy chọn: Chỉnh sửa thẻ DICOM sẽ được sử dụng trong các tập dữ liệu đã xuất. Siêu dữ liệu từ nghiên cứu đã chọn sẽ được tự động điền vào hộp thoại Xuất và bạn có thể chọn một khối lượng Slicer để xuất.
 - Trình chỉnh sửa thẻ DICOM bao gồm một danh sách các bảng. Bảng cho các thẻ chung cho bệnh nhân và nghiên cứu ở trên cùng và các thẻ cho từng loạt bên dưới chúng.
 - “Thẻ” (Tags) trong bảng hiển thị không phải lúc nào cũng được ghi trực tiếp vào thẻ DICOM, chúng chỉ được các plugin DICOM sử dụng để điền thẻ DICOM vào các tệp đã xuất. Điều này cho phép linh hoạt hơn nhiều và các plugin DICOM có thể tự động điền một số thông tin và các plugin có thể hiển thị các tùy chọn xuất khác trong danh sách này (ví dụ: nén, quy ước đặt tên).
 - Lưu các thẻ đã sửa đổi: đánh dấu vào hộp kiểm này để lưu các giá trị thẻ mới trong màn hình quan sát chính một cách liên tục.
 - Cách thiết lập thẻ định danh duy nhất (Unique ID - UID):
 - ▶ Thẻ StudyInstanceUID chỉ định bệnh nhân và nghiên cứu nào sẽ được thêm vào chuỗi mới. Nếu giá trị được đặt thành trống thì một nghiên cứu mới sẽ được tạo. Nên giữ nguyên tất cả các giá trị bệnh nhân và nghiên cứu (PatientName, PatientID, StudyID, v.v.) trong các chuỗi trong cùng một nghiên cứu.
 - ▶ Thẻ SeriesInstanceUID xác định một chuỗi hình ảnh. Theo mặc định, giá trị của thẻ được đặt thành trống, điều này sẽ dẫn đến việc tạo một UID mới và do đó là một chuỗi mới. Trong những trường hợp rất hiếm, người dùng có thể muốn chỉ định một UID, nhưng UID không thể là bất kỳ UID hiện có nào vì điều đó sẽ dẫn đến việc các lát ảnh được xuất ra bị trộn lẫn vào một chuỗi khác. Do đó, UID chỉ được chấp nhận nếu nó không được sử dụng cho bất kỳ hình ảnh nào đã có trong cơ sở dữ liệu.
 - Thẻ FrameOfReferenceUID chỉ định một tham chiếu không gian. Nếu hai hình ảnh có cùng giá trị UID khung tham chiếu thì điều đó có nghĩa là chúng được căn chỉnh không gian. Theo mặc định, giá trị là trống, nghĩa là một UID khung tham chiếu mới được tạo ra và do đó hình ảnh được xuất không được liên kết với bất kỳ hình ảnh nào khác. Nếu một hình ảnh được đăng ký không gian với một hình ảnh khác thì nên sao chép giá trị UID khung tham chiếu từ hình ảnh khác, vì điều này có thể cần thiết để hiển thị hợp nhất các hình ảnh trong một số phần mềm xem lại hình ảnh.
6. Nhấp vào Xuất
 - Trong trường hợp có bất kỳ lỗi nào, một thông báo ngắn sẽ được hiển thị. Thông tin chi tiết hơn về lỗi được cung cấp trong nhật ký ứng dụng.

6 MÔ HÌNH DỮ LIỆU SLICER

Slicer sử dụng MRML để quản lý và tổ chức dữ liệu y tế bằng API được cung cấp bởi thư viện. Mỗi kiểu dữ liệu được đại diện bởi một nút (node) MRML đặc biệt.

Mô hình dữ liệu Slicer MRML được triển khai độc lập với các thành phần trực quan hóa và thuật toán của hệ thống. Các thành phần Slicer khác (Logic và GUI) quan sát các thay đổi trong màn hình quan sát chính MRML và các nút riêng lẻ và xử lý các sự kiện MRML thay đổi.

1. MRML Scene

MRML Scene quản lý các nút MRML: thêm, xóa, tìm, tìm theo loại, v.v. MRML Scene cung cấp tính bền bỉ của các nút MRML (đọc/ghi vào/từ tệp XML).

MRML Scene cung cấp cơ chế Hoàn tác/Làm lại để khôi phục trạng thái trước đó của màn hình quan sát chính và các nút riêng lẻ.

2. Các nút MRML

Vai trò của nút MRML:

- Lưu trữ dữ liệu: Mỗi nút đại diện cho một loại dữ liệu cụ thể trong 3D Slicer, chẳng hạn như hình ảnh (volumes), mô hình (models), phân đoạn (segmentations), hoặc điểm chú thích (fiducials).
- Chuyển đổi dữ liệu sang dạng XML: Nút MRML hỗ trợ việc tuần tự hóa (serialization) dữ liệu để lưu trữ vào tệp XML. Điều này giúp dữ liệu có thể được xuất và nhập một cách dễ dàng.
- Không tích hợp hiển thị hoặc xử lý: Nút MRML không chứa các phương pháp hiển thị hay xử lý dữ liệu; các chức năng này được xử lý bởi các lớp khác như Display Node hoặc Logic Layer.
- Các loại nút MRML cơ bản
 - ▶ Data Node (Nút dữ liệu): Chứa nội dung dữ liệu chính, chẳng hạn như hình ảnh, mô hình hoặc kết quả phân đoạn.
 - ▶ Display Node (Nút hiển thị): Xác định các tùy chọn hiển thị cho nội dung của nút dữ liệu; cho phép sử dụng nhiều nút hiển thị cho cùng một nút dữ liệu, giúp hiển thị dưới nhiều hình thức khác nhau.
 - ▶ Storage Node (Nút lưu trữ): Xác định định dạng tệp và tên tệp được sử dụng để lưu trữ dữ liệu của nút. Đảm bảo tính bền vững cho dữ liệu.

Các nút MRML được tổ chức thành các phân cấp lớp C++, tất cả đều bắt nguồn từ lớp `vtkMRMLNode`.

Tất cả các nút MRML phải triển khai một số API chuẩn nhất định: `ReadAttributes`, `WriteAttributes`, `Copy`, v.v.

Sau đây là một tập hợp các nút MRML cốt lõi lưu trữ trạng thái của các mô-đun cốt lõi: [Danh sách các nút](#)

Các mô-đun giao tiếp thông qua đọc/viết lên các nút mà không cần phải nhận biết nhau.

3. Tham chiếu

Một số nút MRML có tham chiếu đến các nút khác. Nút có thể chuyển đổi có tham chiếu đến một nút chuyển đổi. Nút chuyển đổi có tham chiếu đến nút chuyển đổi cha của nó. Tham chiếu được lưu trữ theo ID nút.

Sử dụng `vtkSetReferenceStringMacro` để đặt ID tham chiếu (nó đăng ký tham chiếu với màn hình quan sát chính).

Các phương thức truy cập sẽ kiểm tra xem nút được tham chiếu có vẫn nằm trong màn hình quan sát chính MRML bằng ID của nó hay không.

7 CHẨN ĐOÁN HỖ TRỢ BỞI MÁY TÍNH

Chẩn đoán hỗ trợ bởi máy tính (CAD) là một lĩnh vực kết hợp giữa công nghệ thông tin và y học, nhằm cải thiện độ chính xác và hiệu quả trong việc chẩn đoán bệnh. Hệ thống CAD sử dụng các thuật toán xử lý và phân tích hình ảnh để hỗ trợ các bác sĩ trong việc đưa ra quyết định.

7.1 QUY TRÌNH HOẠT ĐỘNG CỦA CAD

- Thu thập dữ liệu: Dữ liệu thường là hình ảnh y khoa (như X-quang, CT, MRI) được thu thập từ bệnh nhân.
- Tiền xử lý: Hình ảnh được xử lý để loại bỏ nhiễu và tăng độ rõ nét, giúp cải thiện chất lượng phân tích.
- Phân đoạn: Hệ thống CAD sử dụng thuật toán để phân chia hình ảnh thành các vùng cụ thể, như vùng tổn thương hoặc vùng không liên quan.
- Phát hiện và phân loại: Thuật toán nhận diện các bất thường trong hình ảnh (như khối u hoặc tổn thương) và phân loại chúng theo mức độ nguy cơ hoặc loại bệnh.
- Đánh giá: CAD cung cấp một bản phân tích chi tiết để hỗ trợ bác sĩ trong việc đánh giá tình trạng bệnh và quyết định điều trị.

7.2 VAI TRÒ CỦA CAD TRONG Y HỌC

- Ý kiến thứ hai (Second opinion): CAD không thay thế bác sĩ, nhưng đóng vai trò như một ý kiến hỗ trợ, giúp giảm nguy cơ chẩn đoán sai.
- Tăng độ chính xác: CAD có thể phát hiện các dấu hiệu sớm của bệnh mà đôi khi mắt người không nhận ra.
- Tiết kiệm thời gian: Hệ thống tự động phân tích hàng loạt hình ảnh, giảm thời gian cần thiết cho bác sĩ.

7.3 MỘT SỐ TRỞ NGẠI CỦA CAD:

1. Sự đa dạng (Diversities)

CAD phải xử lý nhiều yếu tố đa dạng, bao gồm:

- Cơ quan/tổ chức mục tiêu: Các cơ quan hoặc mô cần phân tích rất khác nhau, từ tim, phổi đến não.
- Loại bệnh được chẩn đoán: Sự phức tạp trong nhiều loại bệnh khác nhau có thể yêu cầu các cách tiếp cận riêng biệt.
- Đặc điểm bệnh nhân: Tuổi, giới tính, chủng tộc, lịch sử y tế đều có thể ảnh hưởng đến kết quả chẩn đoán.

2. Thống kê trong tập dữ liệu (Statistics in Dataset)

- Số lượng bệnh nhân ít đối với bệnh nghiêm trọng: Những bệnh lý nghiêm trọng thường có số lượng bệnh nhân ít, làm hạn chế dữ liệu để huấn luyện mô hình.
- Bệnh phức tạp: Các căn bệnh khó chẩn đoán thường là mục tiêu của CAD, đòi hỏi sự hỗ trợ cao hơn từ hệ thống.

3. Tiền xử lý (Preprocessing)

- CAD cần thực hiện các bước như phân đoạn (segmentation) và đăng ký hình ảnh (registration), nhưng:
 - ▶ Đây là những bước quan trọng nhưng khó tự động hóa hoàn toàn.
 - ▶ Nhiều nhà nghiên cứu (bao gồm các chuyên gia công nghệ thông tin) có thể né tránh làm việc với giai đoạn này.
 - ▶ Dữ liệu tái tạo (rebuild datasets) thường chỉ có sẵn cho các bệnh phổ biến, gây khó khăn cho các bệnh hiếm.

7.4 CÁC THÁCH THỨC CHO CAD

1. Thu thập dữ liệu và đánh giá chất lượng (Data Collection and Quality Assessment):

- Dữ liệu y tế cần được chuẩn hóa và đảm bảo chất lượng cao để các hệ thống CAD hoạt động chính xác.
- Việc quản lý dữ liệu từ nhiều nguồn khác nhau là một khó khăn lớn.

2. Phát triển các thuật toán phân đoạn tiên tiến (Advanced Segmentation Approaches):
 - Phân đoạn hình ảnh là bước quan trọng trong CAD nhưng thường đòi hỏi các thuật toán mạnh mẽ hơn để xử lý các cấu trúc phức tạp.
3. Trích xuất và chọn lọc đặc điểm (Advanced Feature Extraction / Selection):
 - Việc lựa chọn đặc điểm ảnh hưởng trực tiếp đến khả năng phân loại và dự đoán của CAD.
4. Phân loại và khai thác dữ liệu (Classification and Data Mining):
 - Thuật toán phân loại cần được cải thiện để tăng độ chính xác trong việc nhận diện bệnh lý.
5. Xử lý dữ liệu lớn (Big Data):
 - CAD sẽ cần tích hợp các công nghệ xử lý dữ liệu lớn để phân tích hiệu quả và nhanh chóng hàng triệu trường hợp bệnh nhân.
6. Ứng dụng trong thực tiễn lâm sàng (Clinical Practice Adaptation):
 - Một trong những thách thức lớn nhất là làm cho CAD phù hợp và dễ sử dụng đối với các chuyên gia y tế trong môi trường thực tế.
7. Đánh giá hiệu suất tiêu chuẩn hóa (Standardized Performance Assessment):
 - Hiện chưa có tiêu chuẩn cụ thể để đánh giá hiệu suất của các hệ thống CAD. Điều này cần được phát triển để đảm bảo tính minh bạch và độ tin cậy.

7.5 QUÁ TRÌNH BÙNG NỔ AI VÀ TÁC ĐỘNG ĐẾN CAD

1. Kỷ nguyên đầu tiên của AI (1960):
 - AI thời kỳ này dựa trên phương pháp “Suy luận bằng tìm kiếm” (Inference by search).
 - Nghiên cứu CAD tự động chẩn đoán bắt đầu xuất hiện, đánh dấu bước khởi đầu cho công nghệ xử lý hình ảnh y tế.
2. Kỷ nguyên thứ hai của AI (1980):
 - AI tập trung vào “Suy luận bằng kiến thức” (Inference by knowledge).
 - Đây là thời điểm nghiên cứu CAD tại Đại học Chicago được khởi động, mở đường cho các ứng dụng CAD thực tế hơn trong y tế.
3. Kỷ nguyên thứ ba của AI (2010 và sau đó):
 - Sự phát triển mạnh mẽ của “Suy luận thông qua học tập” (Reasoning by learning), với AI chuyển sang sử dụng học sâu (Deep Learning).
 - CAD đã chuyển mình thành AI-CAD, tích hợp các thuật toán học sâu để cải thiện khả năng nhận diện và phân tích hình ảnh y tế.

Tương lai hứa hẹn của CAD trong thời bùng nổ AI

- Hiệu suất vượt trội:

AI giúp CAD đạt hiệu suất cao hơn 10 triệu lần so với thời kỳ đầu, nhờ sự cải tiến về khả năng tính toán của máy tính.

- Cá nhân hóa chẩn đoán:

AI cho phép phân tích dữ liệu đa chiều, từ hình ảnh y tế đến hồ sơ bệnh lý và dữ liệu di truyền, tạo ra các giải pháp chẩn đoán chính xác và phù hợp với từng bệnh nhân.

- Tự động hóa toàn diện:

Với AI, CAD có thể tự động hóa hầu hết quy trình chẩn đoán, giảm áp lực cho các chuyên gia y tế.

8 PHÒNG THÍ NGHIỆM CHĂM SÓC SỨC KHỎE KỸ THUẬT SỐ (DIGITAL HEALTHCARE LAB)

Mục tiêu: Phát triển các hệ thống kỹ thuật số để cải thiện sức khỏe và đời sống, nghiên cứu một số ứng dụng của từng chủ đề.

8.1 THỰC TẾ ảo (VR) - THỰC TẾ TĂNG CƯỜNG (AR) TRONG CHĂM SÓC SỨC KHỎE

Mọi người thường nghĩ rằng VR-AR chỉ được sử dụng trong trò chơi như:

- Nhập vai vào thế giới giả tưởng.
- Cạnh tranh với các game thủ khác.
- Chơi Beat Saber và nhiều trò chơi khác.

VR có mục đích gì? Không chỉ dành cho giải trí và xã hội. VR còn được sử dụng trong đào tạo phẫu thuật, trị liệu bằng VR... VR cũng hữu ích cho người mắc chứng mất trí nhớ. Phần lớn người dùng là những người bị khuyết tật thể chất và chứng mất trí nhớ nặng. Có 8 giai đoạn của chứng mất trí nhớ.

Mặc dù VR có khả năng tạo ra những thực thể thay thế... nhưng người cao tuổi có xu hướng quan tâm hơn đến các hoạt động mô phỏng cuộc sống thực:

- Thăm lại những nơi họ từng sinh sống.
- Trải nghiệm những hoạt động mà họ từng yêu thích.
- Du lịch đến những địa điểm mới.

Có nhiều ứng dụng của VR-AR dành cho người mắc chứng mất trí nhớ:

- Thiết kế VR dành riêng cho người mất trí nhớ.
- “Hiệu ứng xứ sở thần tiên” để thu hút sự chú ý.
- Giao diện hữu hình để khuyến khích sự tham gia tích cực.
- Video 360 độ thông qua kính VR HMD dành cho những người bị chứng mất trí nhớ trung bình đến nặng và đang sống trong các trung tâm chăm sóc.

Thách thức:

- Tích hợp và ứng dụng: Dù AR và VR mang lại tiềm năng lớn, việc tích hợp chúng vào hệ thống y tế hiện tại khá phức tạp. Các cơ sở y tế cần tìm cách triển khai công nghệ này mà không làm gián đoạn việc chăm sóc bệnh nhân.
- Chi phí cao: Việc triển khai giải pháp AR và VR rất tốn kém, từ việc mua phần cứng đến phát triển ứng dụng riêng. Điều này có thể khiến các cơ sở y tế nhỏ khó tiếp cận công nghệ.
- Tuân thủ quy định: Cân bằng giữa đổi mới công nghệ và tuân thủ quy định y tế là một bài toán khó. Cần đảm bảo các ứng dụng AR và VR đáp ứng các tiêu chuẩn nghiêm ngặt về an toàn bệnh nhân và bảo mật dữ liệu.

Cơ hội:

- Nâng cao chăm sóc bệnh nhân: Công nghệ AR và VR giúp các chuyên gia y tế cung cấp dịch vụ chăm sóc cá nhân hóa, hiệu quả và tập trung vào bệnh nhân hơn, từ đó cải thiện kết quả điều trị.
- Phát triển y tế từ xa: Việc kết hợp AR và VR vào y tế từ xa giúp thu hẹp khoảng cách tiếp cận dịch vụ y tế, đặc biệt ở những khu vực xa xôi hoặc thiếu thốn điều kiện y tế.
- Nghiên cứu sáng tạo: AR và VR mở ra những hướng nghiên cứu y học mới, giúp tăng tốc các thí nghiệm, khám phá thuốc và đột phá y khoa.

8.2 ỨNG DỤNG GAMIFICATION TRONG TRỊ LIỆU

Gamification là việc áp dụng các yếu tố trò chơi và kỹ thuật thiết kế game vào các lĩnh vực phi trò chơi nhằm thúc đẩy và tạo động lực cho người tham gia đạt được mục tiêu cụ thể.

Một số trò chơi đã được phát triển nhằm hỗ trợ điều trị chứng nghiện rượu, ví dụ như Addiction Beater Game - Bài tập huấn luyện điều chỉnh nhận thức cho người nghiện rượu:

- Người chơi được hiển thị hình ảnh có hoặc không có rượu.
- Nếu là hình ảnh có rượu: không làm gì; nếu không có rượu: nhấn nút.
- Người chơi phải phản ứng chính xác trong thời gian giới hạn.
- Lặp lại nhiều lần để tạo thói quen ức chế phản ứng đối với rượu.

Tuy nhiên, vấn đề lớn là tỷ lệ bỏ cuộc cao, đặc biệt là ở thanh thiếu niên do trò chơi gây nhàm chán.

Một số hướng đi khác:

- Dự án gamification dựa trên âm nhạc.
- Trị liệu thực tế ảo dựa trên IBT (Interpersonal Behavioral Therapy).

8.3 AI DỰ ĐOÁN TRONG CHĂM SÓC SỨC KHỎE

- AI hội thoại trong y tế: Hiện tại chủ yếu chỉ dừng ở việc trả lời các câu hỏi y tế cơ bản.
- Ứng dụng thực tế ảo và avatar thông minh vào chăm sóc sức khỏe
- Nguy cơ nội dung có hại

9 FAST MARCHING ALGORITHM

Bài toán lan truyền mặt biên (*Interface Propagation*) là một bài toán gắn liền với thực tế. Trong đó, sóng nước hay những đám cháy đều có thể diễn giải được bằng bài toán lan truyền mặt biên. Trong đó, Narrow Band Level set và Fast Marching là những phương pháp thường được dùng để giải những bài toán này.

Fast Marching algorithm (tạm dịch là Thuật toán Hành tiến nhanh) là một thuật toán tìm đường nhanh, được James A. Sethian sáng tạo vào những năm 1990.

9.1 PHƯƠNG TRÌNH EIKONAL

Phương trình Eikonal là một phương trình đạo hàm riêng phi tuyến tính, có dạng:

$$|\nabla u| = f(x, y) \quad (5)$$

trong đó:

- $u(x, y)$ là hàm ẩn cần tìm, thường biểu diễn khoảng cách hoặc thời gian di chuyển từ một điểm xuất phát
- ∇u là gradient của u , tức là đạo hàm bậc nhất theo các biến không gian. Đây là vector biểu diễn hướng và tốc độ thay đổi của $u(x, y)$.

$$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) \quad (6)$$

- $|\nabla u|$ là chuẩn (norm) của gradient, thể hiện độ lớn của tốc độ thay đổi của u theo không gian.

$$|\nabla u| = \sqrt{\frac{\partial u^2}{\partial x} + \frac{\partial u^2}{\partial y}} \quad (7)$$

- $f(x, y)$ là hàm số tốc độ của mỗi điểm trong không gian. Nếu $f(x, y) = 1$, phương trình trở thành phương trình Eikonal chuẩn, dùng để tính khoảng cách từ 1 điểm nguồn.

Phương trình Eikonal là một dạng tính của phương trình Hamilton-Jacobi. Phương trình Hamilton-Jacobi là một phương trình vi phân riêng rất quan trọng trong cơ học lý thuyết và toán lý.

Phương trình này mô phỏng việc một dòng chảy hoạt động trong một không gian 2 chiều hoặc 3 chiều hữu hạn.

Xét trường hợp đơn giản là khi $f(x, y) = 1$ thì phương trình này trở thành phương trình khoảng cách Euclid, giúp xác định đường đi ngắn nhất từ một tập hợp điểm cho trước.

Ta đã biết hàm số $u(x, y) = (x^2 + y^2)^{\frac{1}{2}} - 1$ ứng với phương trình khoảng cách của đường tròn có thể giải được phương trình Eikonal. Đường nhiên, việc đặt $F(x) = 1$ và cho 0 là điều kiện biên khiến cho bài toán trở thành bài toán tính khoảng cách tới đường cong ban đầu Γ .

Ngược lại, bây giờ hãy cho rằng dữ liệu biên được lấy từ 1 đường cong không lồi, và ta cần tìm khoảng cách từ bất kỳ điểm nào trong miền giá trị cho đến đường cong này.

Cách 1: tại mỗi điểm trên đường cong, ta đặt một “phần tử”, và di chuyển chúng ra khỏi đường cong theo một hướng chuẩn, với tốc độ đơn vị. Tại bất kỳ thời điểm C nào, vị trí của tất cả phần tử này trả về tập hợp tất cả các điểm, có khoảng cách với đường cong ban đầu bằng C .

Cách thứ 2 là vẽ lại tất cả các điểm, nhưng ở một khoảng cách nhất định so với đường gốc. Cách này áp dụng nguyên lý Huygens. Nguyên lý Huygens là một phương pháp hình học để tìm mặt sóng của một sóng lan truyền tại một thời điểm sau, nếu biết mặt sóng tại một thời điểm trước đó. Kết quả của phần tìm mặt sóng này là một “wavefront”, tức là một đường cong bất kỳ thể hiện tập hợp vị trí tất cả các điểm biên tại thời điểm t . [1]

Mặt ngoài (envelope) của đường sóng tạo thành một tập hợp cố định là những “điểm đến đầu tiên”, tức là một đường sóng lan truyền tới đâu, thì điểm đó là cố định. Mặt ngoài của đường sóng khi được kết nối lại sẽ tạo thành một đường cong; và đường cong này không khả vi, cũng như ta không thể tính được gradient ∇u . Vì cần tìm khoảng cách ngắn nhất đến các điểm “envelope”, ta chọn cách làm thứ 2.

Hãy xem đường sóng này như rìa của vũng nước: bất kỳ điểm nào nằm bên trong đường sóng này sẽ luôn ngập nước. Tương tự với một đám lửa, bất kỳ điểm nào, một khi đã cháy thì sẽ không bao giờ tắt. Điều này giúp cách làm trên thoả mãn điều kiện entropy: đảm bảo mọi phương pháp phải phù hợp về mặt vật lý, và thoả mãn tính đơn ánh.

Để làm được điều đó, ta lại cần cộng thêm một hạng tử “làm tròn”, nhằm gạt bỏ những điểm nhọn, đảm bảo kết quả là một vùng tròn, như vậy mới có thể giữ cho phương trình Eikonal luôn khả vi. Hơn nữa, trong thực tế, không có một đường sóng nào có “góc nhọn”, vậy nên cách làm này càng đảm bảo tính thực tế.

$$|\nabla u(x)| = F(x) + \varepsilon \nabla^2 u \quad (8)$$

Mục tiêu là ta cần xây dựng phương pháp số học để tự động trích xuất được giá trị hạng tử này.

9.2 XẤP XỈ PHƯƠNG TRÌNH EIKONAL

Ta có “upwind scheme” của Godunov, hay dạng rời rạc của phương trình Eikonal:

$$\begin{pmatrix} \max(D_{ijk}^{-x}u, -D_{ijk}^{+x}u, 0)^2 + \\ \max(D_{ijk}^{-y}u, -D_{ijk}^{+y}u, 0)^2 + \\ \max(D_{ijk}^{-z}u, -D_{ijk}^{+z}u, 0)^2 \end{pmatrix} = F_{ijk} \quad (9)$$

với:

- D^- và D^+ là các toán tử sai phân tiến và lùi
- F_{ijk} là độ chậm tại điểm lưới (i, j, j) , tức là nghịch đảo tốc độ lan truyền tại điểm đó.

Công thức này mô tả cách ước lượng giá trị của hàm u tại một điểm lưới dựa trên các giá trị lân cận.

Một cách giải thông dụng là sử dụng phương pháp lặp, như đề xuất của Rouy và Tourin. Ý tưởng cập nhật giá trị u từng bước dựa trên lược đồ stencil gồm 6 điểm lân cận, liên tục cho đến khi phương trình được thoả mãn.

QUADRATICSOLVE():

```
1  for iter ← 1 to n:
2      for i, j, k ← 1 to Mesh:
3          Solve Quadratic for  $u_{ijk}^{\text{iter}+1}$ , given
4           $u_{i-1,j,k}^{\text{iter}}, u_{i+1,j,k}^{\text{iter}}, u_{i,j-1,k}^{\text{iter}}, u_{i,j+1,k}^{\text{iter}}, u_{i,j,k-1}^{\text{iter}}, u_{i,j,k+1}^{\text{iter}}$ 
5      end for
6  end for
```

9.3 GIẢI THUẬT

Ý tưởng chính của Fast Marching là xây dựng nghiệm theo cách có trật tự, đi từ những điểm có giá trị nhỏ nhất rồi mở rộng dần ra.

Nó sử dụng một chiến lược gọi là upwind difference, nghĩa là chỉ lấy thông tin từ những điểm đã biết để tính toán điểm tiếp theo. Điều này đảm bảo rằng giá trị tại mỗi điểm luôn dựa trên những giá trị đã được xác định trước đó, phù hợp với điều kiện entropy.

Fast Marching không sử dụng phương pháp lặp đã nêu ở phần trước, mà sử dụng lược đồ cập nhật theo thứ tự chính xác (causal ordering) để tìm nghiệm.

Quy trình có thể được mô tả như sau:

1. Bắt đầu từ điểm gốc: Giả sử ta đang tính khoảng cách ngắn nhất từ 1 điểm xuất phát
2. Mở rộng tập nghiệm: Giống như một làn sóng lan ra xung quanh, thuật toán cập nhật các điểm gần đó theo thứ tự từ nhỏ đến lớn. Các điểm này có dạng một dải hẹp (narrow band), nằm liền kề với các điểm biên của làn sóng.
3. Cập nhật điểm kế tiếp: Mỗi lần mở rộng, thuật toán chọn điểm có giá trị nhỏ nhất trong dải hẹp, và đưa điểm đó vào bên trong làn sóng. Nói cách khác, trong bước kế tiếp, làn sóng sẽ đi qua điểm này.

Quá trình này sẽ được lặp lại.

FASTMARCHING():

```

1  mark all initial condition points as Alive
2  mark all points one grid point away as Close
3  mark all other points as Far
4
5  while Close is not empty:
6      Trial  $\leftarrow$  point in close with the smallest value of u
7
8  for each neighbor N of Trial:
9      if N is not Alive:
10         if N is in Far:
11             Remove N from Far
12             Add N to Close
13         end for
14  Recompute u for all Close neighbor of Trial using piecewise quadratic equation
15  Move Trial from Close to Alive

```

Thuật toán này hoạt động tốt, vì các điểm xung quanh sẽ không tồn tại giá trị nào thấp hơn nhưng điểm đã cho. Ta có thể hiểu quá trình này như một dòng nước đang chảy ở một vùng gồ ghề, trong đó, ở vùng kế cận với dòng nước, điểm nào có cao độ thấp hơn, nước sẽ chảy đến đó trước.

Độ phức tạp của Fast Marching là $O(N \log N)$, ứng dụng heap. Có nhiều cách để lưu trữ các điểm, nhưng trong trường hợp này, ta lưu trữ giá trị *Trial* theo 1 cấu trúc nhất định. Khi 1 điểm được chọn, các điểm liên kề được cập nhật, và giá trị *u* của chúng có thể thay đổi. Vì vậy, một phần nhỏ trong số các điểm phải được sắp xếp lại để có được thứ tự các điểm như ý muốn. Điều này liên tưởng đến cấu trúc dữ liệu heap. Cụ thể, ta xây dựng min-heap, là một cây nhị phân đầy đủ, với tính chất là giá trị tại tất cả các nút phải bé hơn hoặc bằng giá trị của tất cả các nút con. Trong thực tế, ta xây dựng heap dưới dạng 1 mảng, với 1 nút tại index *k*, và các nút con ở index $2k$ và $2k + 1$. Từ đây, ta rút ra rằng nút cha của bất kỳ nút nào nằm tại index $k / 2$, và phần tử nhỏ nhất nằm tại $k = 1$ của mảng. Truy xuất nút con, nút cha chỉ là thao tác trên mảng với độ phức tạp $O(1)$.

Ta gọi:

- **UpHeap** (Vun đống từ dưới lên): Khi thêm một phần tử mới vào Heap, phần tử này sẽ được đẩy lên vị trí thích hợp để duy trì tính chất của Heap.
- **DownHeap** (Vun đống từ trên xuống): Khi loại bỏ nút gốc, phần tử cuối cùng của Heap được đưa lên gốc và tiến hành điều chỉnh từ gốc xuống để duy trì tính chất của Heap. [2]

Các giá trị của *u* được lưu giữ cùng với các chỉ mục thể hiện vị trí trên không gian lưới. Thuật toán sẽ hoạt động như sau: Tìm phần tử nhỏ nhất trong *NarrowBand*. Gọi thao tác này là **FindSmallest**, nó xoá bỏ gốc của cây, rồi thực hiện **DownHeap** để thoả mãn cấu trúc heap được yêu cầu. Tìm các điểm tiếp theo có trạng thái “FarAway”, có bao gồm loại trừ các điểm đã đi qua (*Alive*), đánh dấu các điểm đó, rồi thực hiện thao tác **Insert**, tức thêm điểm mới vào heap. Kích cỡ của heap được tăng 1, phần tử mới được đặt ở cuối heap, rồi thực hiện **UpHeap** để sắp xếp lại.

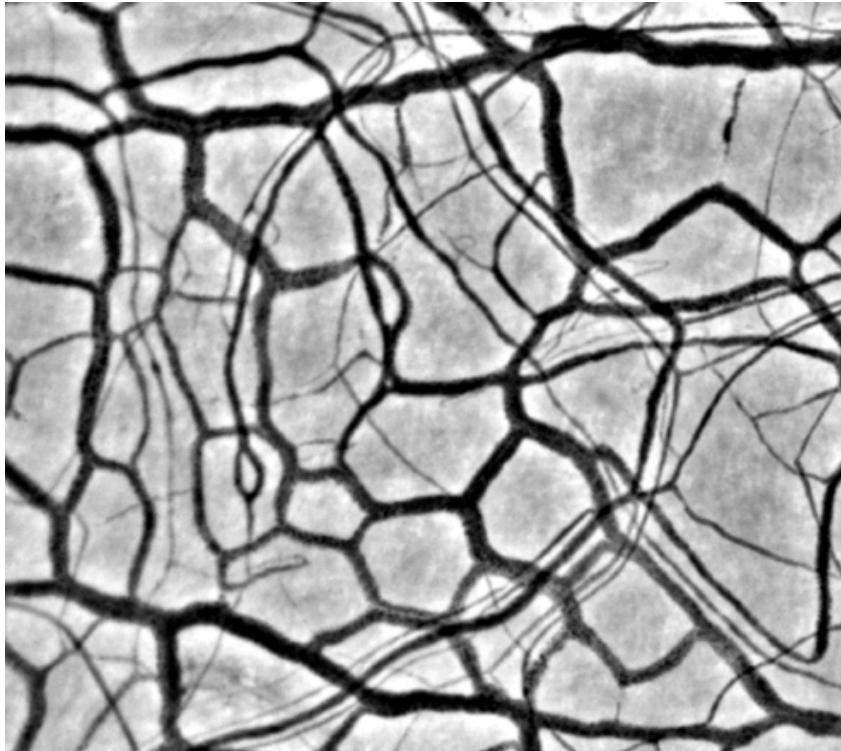
Nếu có điểm nào trong heap phát hiện thay đổi giá trị *u* thì thực hiện lại **UpHeap** để sắp xếp lại cho phù hợp.

Heap là cây nhị phân đầy đủ, luôn đảm bảo cân bằng, nên UpHeap và DownHeap luôn có độ phức tạp là $O(\log N)$. Ứng với N phần tử, độ phức tạp của cả quá trình tăng lên $O(N \log N)$. Đối với NarrowBand, nếu có 1 con trỏ đến phần tử tương ứng từ cấu trúc lưới đến heap thì độ phức tạp có thể giảm xuống $O(1)$. Nếu không, trường hợp tệ nhất là độ phức tạp tăng lên $O(N)$.

Kết hợp lại, trên một lưới gồm N điểm, độ phức tạp của Fast Marching là $O(N \log N)$. Trên không gian ba chiều, mỗi chiều là M điểm, độ phức tạp của Fast Marching được giảm từ $O(N^4)$ xuống $O(N^3 \log N)$. [3]

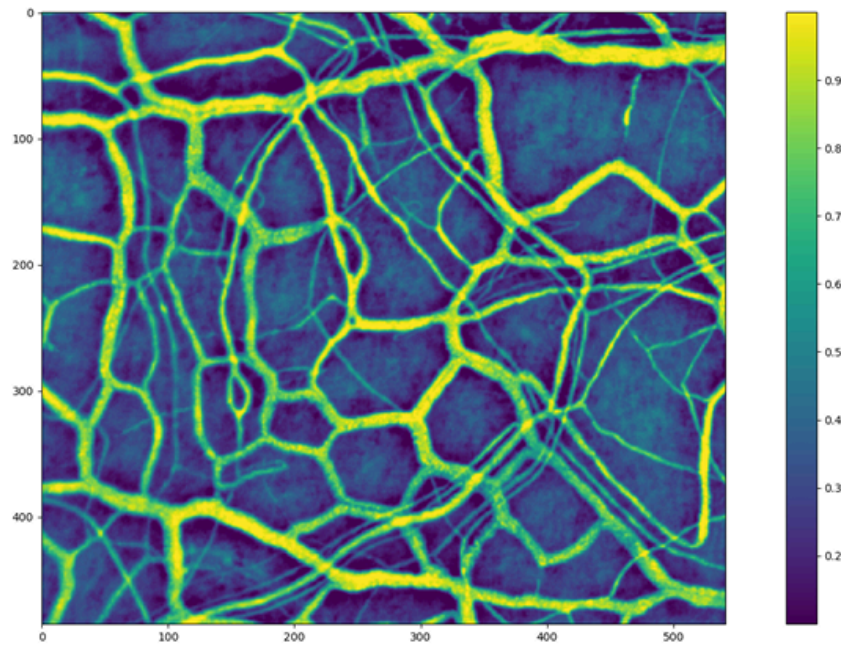
9.4 ỨNG DỤNG TRONG PHÂN TÁCH MẠCH MÁU

Để thực hành Fast Marching để phân tách mạch máu, một bức ảnh mạch máu được sử dụng:



Hình 2: Mạch máu ban đầu

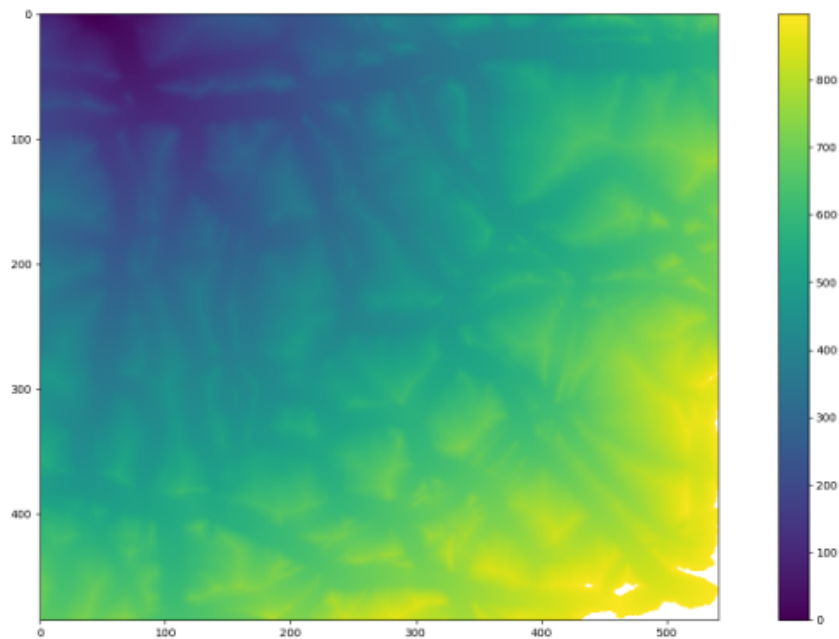
Hàm tốc độ lan truyền được dựa vào màu sắc của điểm ảnh. Điểm ảnh càng tối thì tốc độ càng cao, có nghĩa là tốc độ lan truyền trên các mạch máu sẽ cao hơn tốc độ bên ngoài mạch máu. Hàm tốc độ là $F(i, j)$ có giá trị tối đa là 1, giá trị tối thiểu là 0.1, tạo thành minh họa như hình sau.



Hình 3: Đồ thị hàm tốc độ

Ta có thể thấy rõ rằng những điểm thuộc về mạch máu có màu vàng hoặc xanh lục, ứng với tốc độ từ 0.6 trở lên, nghĩa là nó lan truyền đến rất nhanh. Ngược lại, các vùng còn lại mang màu xanh lục, nghĩa là tốc độ chậm.

Khi đã có được hàm tốc độ, thuật toán Hành tiến nhanh được thực thi.

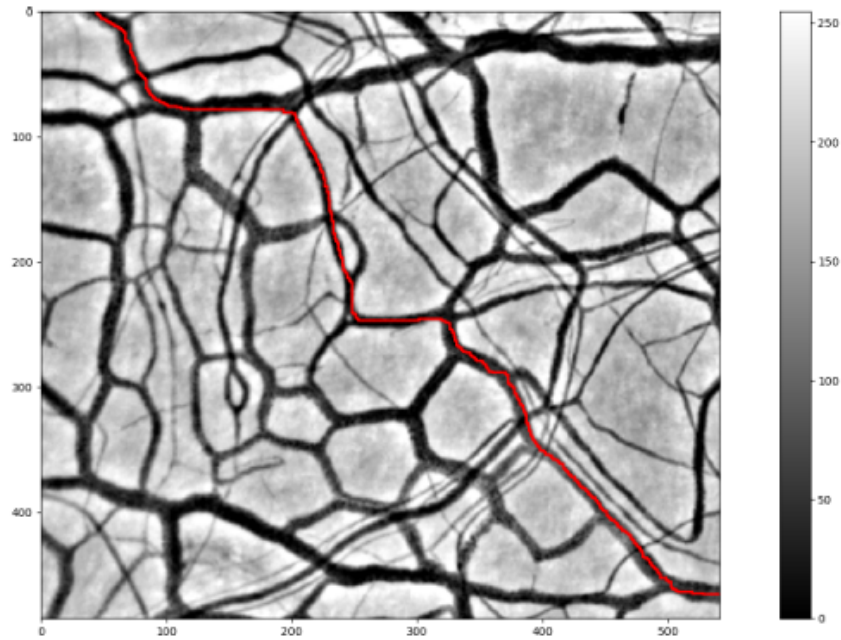


Hình 4: Đường đi của thuật toán từ xuất phát điểm bên trái - trên cùng

Từ bản đồ đường đi, có thể thấy thuật toán cơ bản đi dần đều từ phía trên cùng bên trái, đến phía dưới cùng bên phải. Tuy nhiên, các mạch máu vẫn mang màu xanh lục đậm,

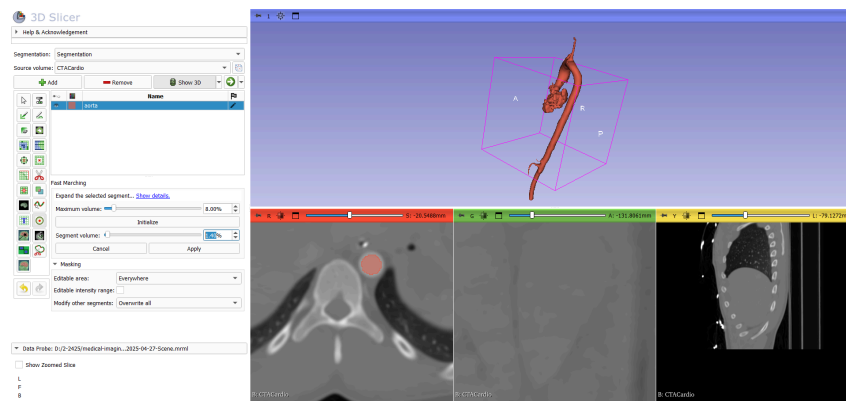
nghĩa là các mạch máu đã được thuật toán bao phủ nhanh chóng, trước khi bao phủ những vùng khác.

Bằng bản đồ T đã có, ta có thể xác định con đường ngắn nhất từ điểm khởi đầu cho đến điểm kết thúc. Con đường này được tìm thấy bằng Gradient Descent trong không gian liên tục. Đường nhiên, do tốc độ bên trong mạch máu nhanh hơn các điểm còn lại, con đường ngắn nhất có đi trên mạch máu.



Hình 5: Đường đi ngắn nhất thuật toán tìm được

9.4.1 THỰC NGHIỆM TRÊN 3D SLICER



Hình 6: Phân tích động mạch chủ trong ứng dụng 3D Slicer

Các bước thực hiện trong ứng dụng như sau:

1. Chọn dataset mẫu, ở đây là CTCardio.
2. Chọn công cụ Segment Editor từ thanh tác vụ phía trên màn hình.
3. Bật công cụ Paint, phía bên trái màn hình. Có thể chọn màu sắc nếu cần.
4. Lựa chọn độ sâu tương ứng của lát cắt cho đến khi nhìn thấy bộ phận tương ứng
5. Đánh dấu một số vị trí thuộc bộ phận động mạch chủ. Không có quy tắc nhất định trong việc đánh dấu. Càng nhiều dấu, động mạch chủ càng xuất hiện chính xác.

Có thể lặp lại bước 4 và 5 cho đến khi thấy hợp lý.

1. Bật công cụ **Fast Marching**
2. Không thay đổi thông số nào, bấm **Initialize**.
3. Khi đã hoàn thành, chỉnh **Segment Volume** về khoảng 1-2%.
4. Chọn **Show 3D** để mô hình 3D xuất hiện.

Fast Marching sẽ lấy gần hết tất cả chi tiết có trong mô hình. Tuy nhiên, ta chỉ cần trích xuất động mạch chủ, nghĩa là một thể tích rất nhỏ trong tổng thể tích của toàn bộ mô hình. Lựa chọn **Segment Volume** chỉ 1-2%, tức là chỉ lấy dưới 2% thể tích tính từ điểm khởi đầu của thuật toán Fast Marching (ở đây là bên trong động mạch chủ đã chọn sẵn bằng Paint).

THAM KHẢO

- [1] J. A. Sethian, “Fast Marching Methods,” Nov. 1998.
- [2] rdsic.edu.vn, “Upheap Downheap: Hướng Dẫn Chi Tiết Thuật Toán Vun Đống Đầy Đủ Nhất — rdsic.edu.vn.”
- [3] D. Alblas, “Implementing and Analysing the Fast Marching Method.” Jun. 30, 2018.

A Phụ Lục