Thuật Toán Phân Tích Thể Tích Tim Nguyên Khối 3D Bằng Hàm Đa Biến Và Cấu Trúc Dữ Liệu Cây Chỉ Số Nhị Phân

Nguyen Le Quoc Bao & Le Tuan Hy

Tóm Tắt

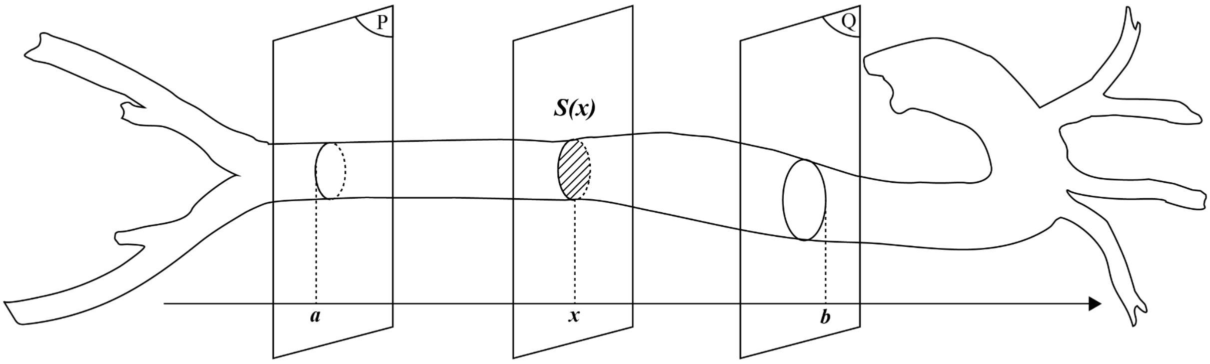
Các mô hình học sâu đang rất phát triển trong lĩnh vực y học, đặc biệt là nhiệm vụ phân vùng và tái tạo cấu trúc 3D của tế bào, nội quan. Để hỗ trợ vào dự án nghiên cứu “Phần mềm tích hợp học sâu để phân vùng và tái tạo cấu trúc tim nguyên khối trong không gian 3D mô phỏng cho ứng dụng thực hành y khoa”, chúng tôi đã nghiên cứu và xây dựng nên một thuật toán mới áp dụng giải tích hàm đa biến, tích phân bội, cấu trúc dữ liệu cây chỉ số nhị phân (hay cây Fenwick) có thể tính toán thể tích một bộ phận trong không gian với tốc độ tối thiểu, giúp cho việc tích hợp vào phần mềm trở nên dễ dàng, nhanh chóng, hiệu quả. Thông qua mô hình toán học và thiết kế thuật toán nghiêm ngặt, chúng tôi chứng minh tính hiệu quả của phương pháp tiếp cận của mình, mở đường cho việc cải thiện độ chính xác và hiệu quả trong phân tích thể tích cấu trúc tim.

I. Mở Đầu

Trong bối cảnh hình ảnh y tế hiện đại, việc chuyển đổi dữ liệu chụp cắt lớp thô thành mô hình ba chiều (3D) chính xác được coi là một xu hướng đang phát triển có tầm quan trọng hàng đầu. Sự phát triển này đòi hỏi các phương pháp xử lý hậu kỳ mạnh mẽ để đảm bảo phép đo và phân tích tỉ mỉ các cấu trúc 3D phức tạp này với độ chính xác và hiệu quả tối đa. Điều quan trọng đặc biệt là việc định lượng các thông số khác nhau trong hệ thống tim mạch, trong đó bao gồm đo chỉ số đường kính, diện tích và thể tích của các cấu trúc quan trọng, ví dụ như ống động mạch chủ. Các số liệu này đóng vai trò là chỉ số quan trọng cho các bệnh lý như phì đại hoặc hẹp, gây ra rủi ro đáng kể cho sức khỏe của bệnh nhân. Do đó, việc đánh giá chính xác và lập kế hoạch trước phẫu thuật được hỗ trợ bởi các phân tích như vậy sẽ nâng cao đáng kể hiệu quả lâm sàng và độ an toàn của các can thiệp phẫu thuật. Tuy nhiên, mặc cho các phần mềm tái tạo ảnh chụp cắt lớp đã có những bước tiến, việc đo thể tích chính xác của mô hình tim 3D vẫn là một thách thức, nhấn mạnh sự cần thiết của một giải pháp tối ưu trong bối cảnh công nghệ hình ảnh y học hiện nay. Mặc dù phần mềm hiện có chuyển đổi thành thạo dữ liệu chụp cắt lớp y tế thành các mô hình 3D toàn diện, khả năng thực hiện phân tích thể tích trong thời gian thực vẫn chưa có hoặc chưa tối ưu cho nhiều lần truy vấn liên tục do người dùng thực hiện cắt mổ, thay đổi kết quả phân vùng tái tại 3D. Vì vậy một thuật toán tối ưu với độ phức tạp thuật toán tối ưu cho việc phân tích thể tích là cực kỳ cần thiết.

II. Phương Pháp

1. Tích Phân Của Hàm Số Diện Tích



Hình 1: Mô tả ý tưởng tính thể tích ống động mạch chủ bằng tích phân của hàm số diện tích

Ta có thể xác định ống động mạch trên có mắc bệnh phì đại hay không bằng cách so sánh thể tích của ống so với thông số tiêu chuẩn. Ta cắt động mạch chủ với 2 mặt phẳng $(P),(Q)$ bằng phương pháp. Giới hạn a,b trên trục Ox của một vật thể là $(x=a, x=b, a<b)$. Một mặt phẳng tùy ý vuông góc với trục Ox tại điểm x và $(a \leq x \leq b)$ cắt vật thể theo thiết diện có diện tích là $S(x)$ và phương trình $S(x)$ liên tục trên đoạn $[a;b]$. Vậy thể tích của bộ phận tim trên được tính bởi công thức tích phân sau:

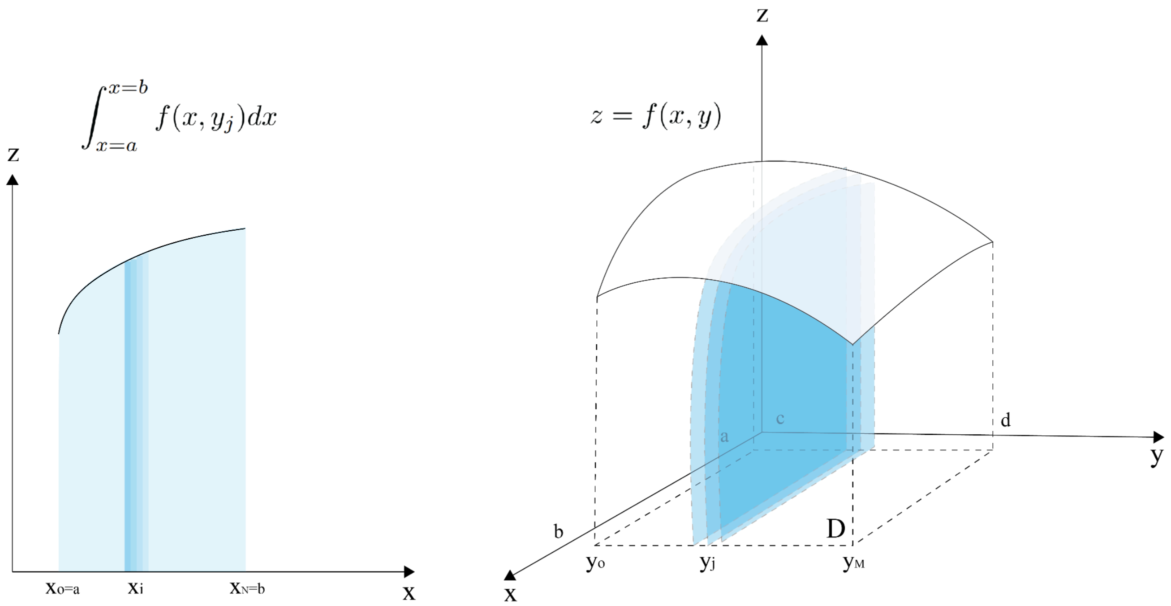
$$V = \int_a^b S(x)dx$$

Các hình hộp có hình dạng cố định thì $S(x)$ cố định khi x thay đổi. Tuy nhiên $S(x)$ trong trường hợp trên lại không cố định. Với bộ ảnh cắt lớp nếu ta có n lắt cắt thì với mỗi lắt thứ i thì ta có S_i là diện tích của bộ phận đó. Điều này tương đương với việc ta có một phương trình $S(i)$ với i là biến số. Việc xấp xỉ một hàm số $S(x)$ không phải là điều dễ dàng. Ta có thể dùng phương pháp nội suy Lagrange. Tuy nhiên điều này thực sự mang rất nhiều hạn chế. Thứ nhất việc làm trên chỉ đúng nếu ta chia bộ phận thành n lát cắt và $n \to +\infty$ thì mới đảm bảo tính chính xác. Điều này thực sự bất khả thi trong thực tế vì số lát cắt n thường không đủ nhiều $(300 \to 512)$ và có giới hạn. Thứ hai, việc xấp xỉ một hàm số cũng là một nhiệm vụ phức tạp khiến cho thuật toán này chạy lâu. Như vậy cách tiếp cận bài toán với phương pháp trên là không tốt. Tuy nhiên, ứng dụng của giải tích và tích phân có rất nhiều tiềm năng trong việc đo thể tích vật thể, cụ thể với tích phân kép và tích phân bội.

2. Giải Tích Cho Hàm Đa Biến

Giải tích cho hàm đa biến tập trung giải quyết những bài toán của phương trình $z = f(x, y)$ trong không gian 3D. Đặc biệt bộ môn toán này có nhiều áp dụng thực tế. Điều này mở ra tiềm năng mới cho ứng dụng vào đề tài liên quan trực tiếp đên không gian 3 chiều của chúng tôi. Tích phân kép và tích phân bội được ứng dụng chủ yếu trong việc đo thể tích một vật.

2.1 Tích Phân Kép



Hình 2: Ý tưởng bài toán tích phân kép để tính thể tích phía dưới một mặt cong

Bài toán tích phân đơn (hình trái) : diện tích bên dưới một đồ thị $f(x)$ cong $S = \int_a^b f(x)dx$ với dx tượng trưng khoảng cách giữa x_{i+1} và x_i là rất nhỏ hay $dx0$. Với bài toán tích phân kép, nếu ta nhìn kỹ thì đây cũng chỉ là tổng hợp nhiều bài toán tích phân đơn lại với nhau. Đầu tiên ta chia mặt cong $z = f(x, y)$ thành các mặt $S_j, S_{j+1}, \cdots, S_M$ tương ứng với từng $y_j, y_{j+1}, \cdots, y_M$ thì mỗi mặt S_j với y_j cố định là một bài toán tích phân đơn. Sau cùng ta sẽ cộng gộp các mặt S_j lại để tính thể tích V, đây chính là tích phân của S. Vì vậy ta có công thức:

$$D = \begin{cases} a \leq x \leq b \\ c \leq y \leq d \end{cases}$$

$$S = \int_a^b f(x, y)dx$$

$$V = \int_c^d \left(\int_a^b f(x, y)dx \right) dy = \int_c^d dy \int_a^b f(x, y)dx$$

Điều kiện để làm tích phân kép ở trên là biết được phương trình $z = f(x, y)$. Tuy nhiên trong thực tế ta chỉ biết $(x_i, y_i, z_i)|i \to n$. Nên ta sử dụng công thức xấp xỉ bằng brute force như sau:

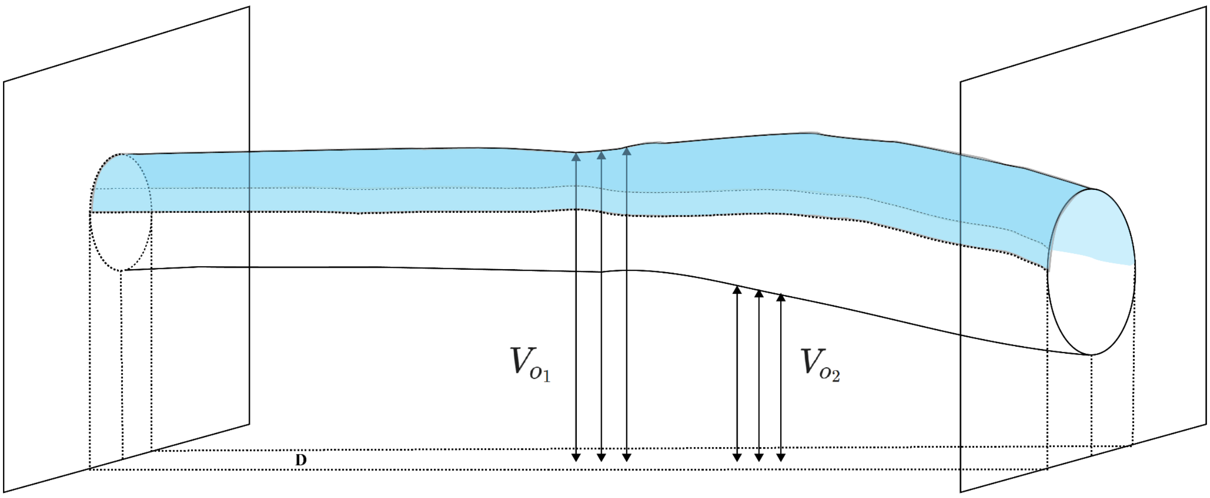
$$\begin{cases} \Delta x_i = x_{i+1} - x_i \\ \Delta y_i = y_{i+1} - y_i \end{cases}$$

$$S_j = \lim_{\Delta x_i \to 0} \Sigma_{i=a}^{i=b} z_{ji} \Delta x_i$$

$$V = \lim_{\Delta y_j \to 0} \Sigma_{j=c}^{j=d} S_j \Delta y_j$$

$$\to V = \lim_{\Delta x_i \to 0} \Sigma_{j=c}^{j=d} (\Sigma_{i=a}^{i=b} z_{ji} \Delta x_i) \Delta y_j$$

2.2 Áp Dụng Thực Tế



Hình 3: Tính toán thể tích một phần cung động mạch chủ với tích phân kép

Khi muốn tính thể tích một bộ phận (ví dụ động mạch chủ - aorta) ta chia cái ống này ra hai phần là nửa trên (o_1) và nửa dưới (o_2) nên miền D là như nhau. Thể tích phần bên trong sẽ là $V_{o_1} - V_{o_2}$:

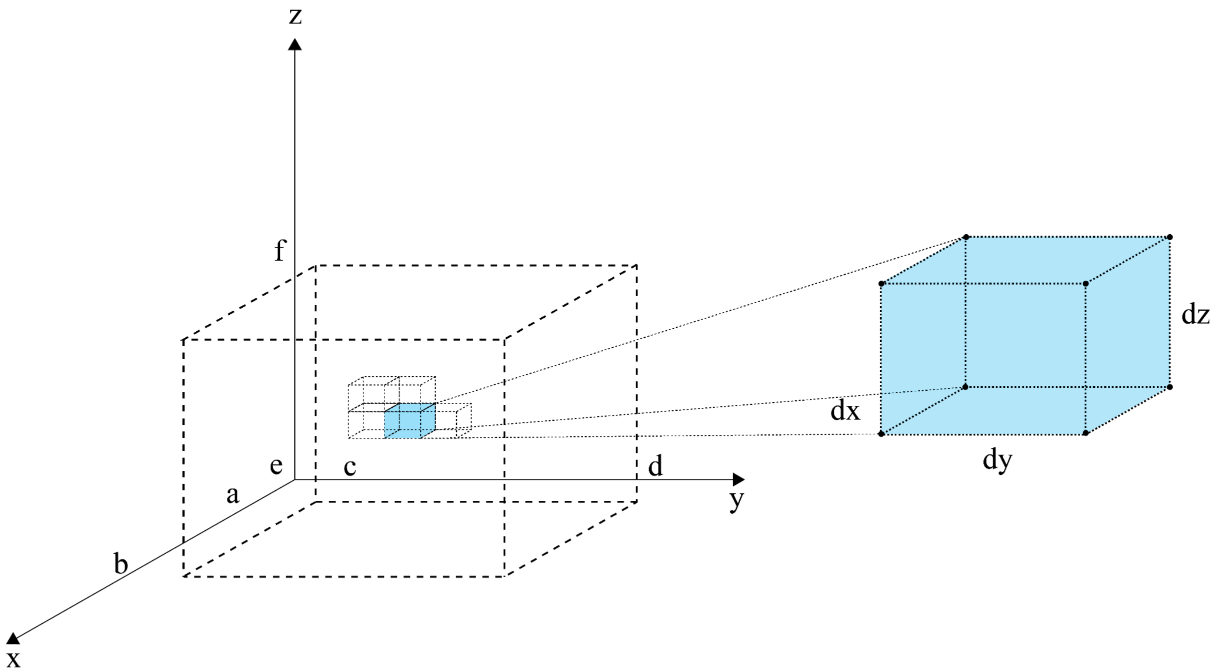
$$V_o = V_{o_1} - V_{o_2}$$

$$\to V_o = \lim_{\substack{\Delta x_i \to 0 \\ \Delta y_j \to 0}} (\Sigma_{j=c}^{j=d} (\Sigma_{i=a}^{i=b} z_{ij}^{o_1} \Delta x_i) \Delta y_j) - \Sigma_{j=c}^{j=d} (\Sigma_{i=a}^{i=b} z_{ij}^{o_2} \Delta x_i) \Delta y_j)$$

$$\to V_o = \lim_{\substack{\Delta x_i \to 0 \\ \Delta y_j \to 0}} \Sigma_{j=c}^{j=d} (\Sigma_{i=a}^{i=b} (z_{ij}^{o1} - z_{ij}^{o_2}) \Delta x_i) \Delta y_j$$

Với cách làm này, ta cần tìm được cắt đôi ở giữa để chia vật thể ra làm hai phần và chỉ sử dụng những voxel nằm ở vị trí ngoài biên. Tuy nhiên với cách tính với tích phân kép như trên thì khi có chuyên gia/bác sĩ thực hiện thao tác cắt mổ hay thay đổi kết quả phân vùng dẫn đến thay đối kết quả tái tạo 3D thì ta phải chạy thuật toán tính thể tích lại lại từ đầu. Nếu gọi K là tổng số lần thực hiện thao tác cắt bỏ hay cập nhật thì độ phức tạp của thuật toán sẽ là $O(K \times M \times N \times S)$ với S là độ phức tạp thực hiện các thao tác phụ như tìm đường cắt giữa và xác định voxel ngoài biên. Vì lý do này nên chúng tôi chuyển sang nghiên cứu tích phân bội ba và ứng dụng cấu trúc dữ liệu khác để giải quyết vấn đề trên.

2.3 Tích Phân Bội Ba



Hình 4: Tính toán thể tích vật thể với tích phân bội ba

Ý tưởng của tích phân bội ba xuất phát từ việc chia khối vật thể O thành n khối hình hộp chữ nhật hay hình lập phương rất nhỏ O_i với chiều dài, chiều rộng, chiều cao lần lượt là dx, dy, dz. Việc tính xấp xỉ thể tích của vật thể ban đầu đơn giản bằng cách lấy tổng thể tích các O_i với $1 \leq i \leq n$ và $n \to +\infty$. Dựa vào hình minh họa và ý tưởng ta có công thức:

$$D = \begin{cases} a \leq x \leq b \\ c \leq y \leq d \\ e \leq z \leq f \end{cases}$$

$$V = \iiint_D dz\, dy\, dx = \int_a^b \int_c^d \int_e^f dz\, dy\, dx$$

Hay trong lập trình ta sử dụng công thức brute force sau:

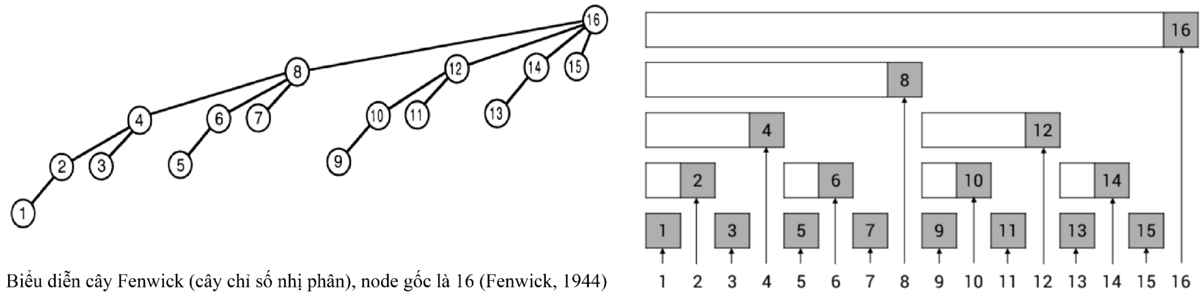
$$V = \lim_{n \to +\infty} \Sigma_{i=1}^{n} \Delta z_i \Delta y_i \Delta x_i$$

Trong việc tái tạo 3D một bộ ảnh cắt lớp y học, một tệp ảnh có cách tổ chức một khối (volume) với ba chiều (N, M, P) với N là chiều rộng, M là chiều dài và P là chiều cao. Vậy dù cho mô hình 3D có bị cắt mổ hay kết quả phân vùng có bị thay đổi thì vẫn nằm trong vùng không gian $N \times M \times P$. Vậy nếu như ta quản lý được vùng không gian này và lưu trữ được voxel nào nằm bên trong thể tích của vật thể một cách hiệu quả thì nhiệm vụ thực hiện hai truy vấn là truy vấn trên vùng và cập nhật trên vùng trở nên dễ dàng. Bài toán này đưa chúng tôi đến cấu trúc dữ liệu Fenwick Tree hay Cây Chỉ Số Nhị Phân.

3. Cây Chỉ Số Nhị Phân

3.1 Nguyên Lý Hoạt Động

Fenwick Tree là một cấu trúc dữ liệu được sử dụng khá phổ biến trong lập trình thi đấu, được giới thiệu trong nghiên cứu "A new data structure for cumulative freqency tables" (Peter M.Fenwick, 1994). Fenwick Tree có các đặc diểm sau: Truy vấn kết quả bài toán con $[L, R]$ với độ phức tạp $O(logN)$, cập nhật giá trị cho 1 (hoặc 1 đoạn) với độ phức tạp $O(logN)$, bộ nhớ thấp $O(N)$ và tốc độ xử lý nhanh (nhờ sử dụng phép xử lý bit). Cụ thể, với mỗi thao tác cập nhật, bit cuối cùng luôn được dịch lên ít nhất 1 lần, dẫn đến *logN* lần dịch chuyển bit tối đa.



Biểu diễn cây Fenwick (cây chỉ số nhị phân), node gốc là 16 (Fenwick, 1944)

Trong Fenwick Tree, phần tử i tronng cây lưu kết quả $F[i]$ của bài bài toán con chứa 2^k phần tử bắt đầu/kết thúc tại vị trí i với k là bit 1 thấp nhất trong biểu diễn nhị phân của i. Thông thường ta sử dụng cây Fenwick tiền tố.

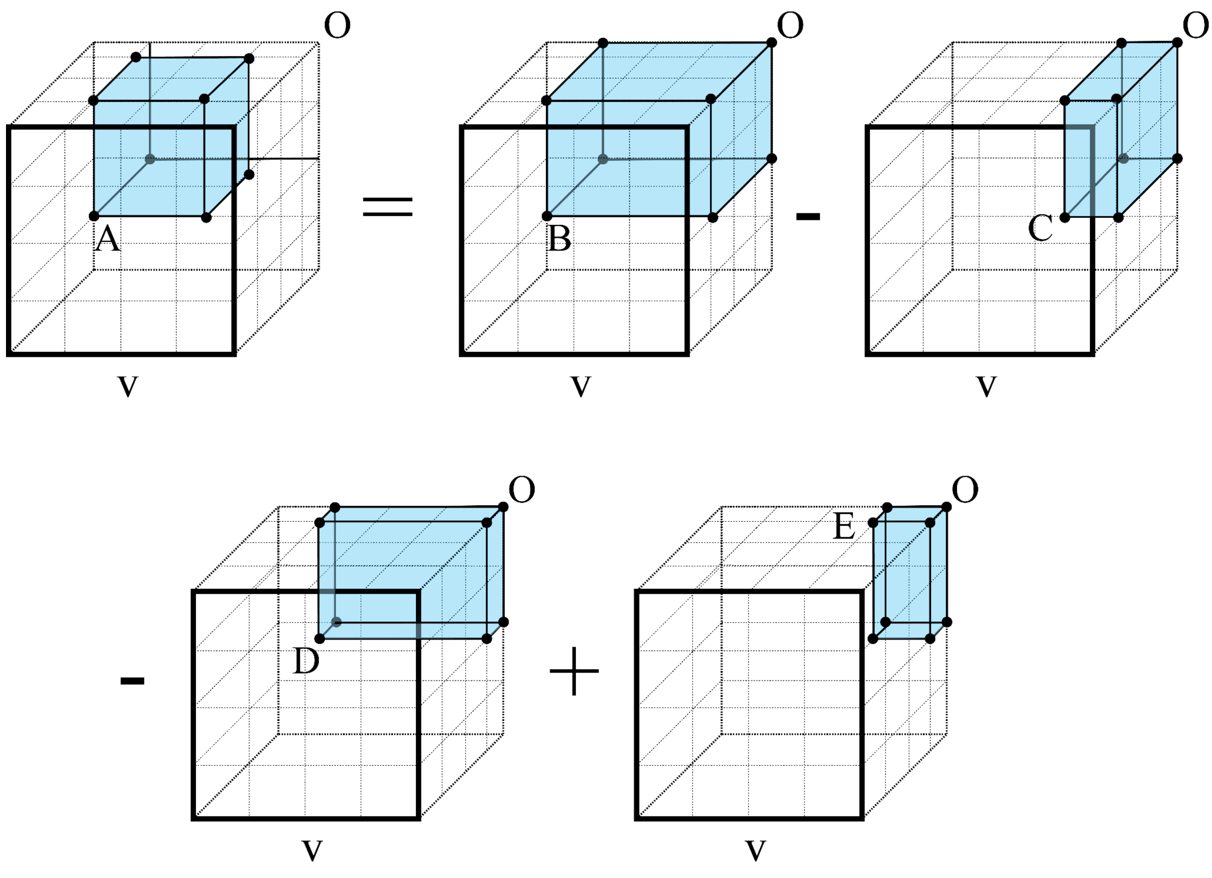
Nhìn vào biểu diễn cây bên trên, ta dễ dàng thấy được các node có số thự tự lẻ thì chỉ quản lý một mình node đó, còn các node có thứ tự id sẽ có cha với thứ tự là $id + 2^k$. Việc tìm ra số 2^k được thực hiện đơn giản bằng cách thực hiện phép xử lý bit sau $2^k = -id \quad id$.

Cây Fenwick hay cây BIT cho phép chúng ta thực hiện: cập nhật một phần tử, truy vấn đoạn, cập nhật một đoạn, truy vấn một phần tử. Thao tác truy vấn trên đoạn giống như việc ta loại bỏ hai bên (thao tác cắt mổ) nhưng không làm mất đi thông tin của phần bị cắt bỏ để hỗ trợ chức năng hoàn tác. Truy ván tổng đoạn con từ (u, v) như sau với độ phức tạp $O(logN)$:

$$\begin{aligned} \text{sumRange}(u, v) &= \text{sumRange}(1, v) - \text{sumRange}(1, u - 1) \\ &= \text{getSum}(v) - \text{getsum}(u - 1) \end{aligned}$$

3.2 Cây Chỉ Số Nhị Phân 3D

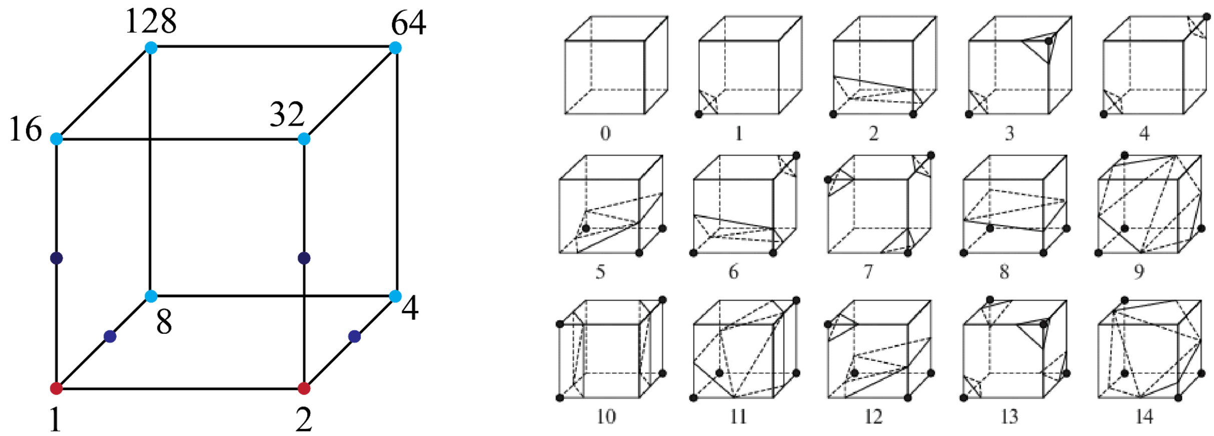
Đây là một áp dụng mới mẻ của cây tìm kiếm nhị phân áp dụng cho các mảng ba chiều. Cũng giống như cây nhị phân 1D và 2D, ta áp dụng quy luật bù trừ (inclusion & exclusion) để truy vấn tổng các phần tử trên một vùng không gian 3D (bỏ các phần xung quanh, tương tự như cắt bỏ nhưng không làm mất dữ liệu) và thực hiện cập nhật một vùng không gian (thay đổi kết quả phân vùng) với độ phức tạp tối thiểu $O(logN \times logM \times logP)$ với (N, M, P) là kích thước của khối ảnh cắt lớp.



Hình 5: Cách truy vấn một vùng trong mảng 3D với cây chỉ số nhị phân

Gọi V là mảng 3D quản lý toàn bộ vùng không gian khi tái tạo 3D. Với bộ dữ liệu hiện tại của chúng tôi thì V có kích thước mộ $(N, M, K) = (512, 600, 600)$. Vậy một khối lập phương nằm hoàn toàn trong vật thể có thể tích là $1(\text{pixel}^3)$, Vậy có các giá trị nào khác của một voxel khi vật thể chỉ đi qua voxel đó nhưng không hoàn toàn đi qua. Điều này liên quan đến thuật toán tái tạo 3D marching cubes mà chúng tôi đang sử dụng để tái tạo vật thể 3D.

4. Thuật Toán Tái Tạo 3D Marching Cubes



Hình 6: Mô tả 15 trường hợp của thuật toán Marching Cubes

Lorensen & Cline (1987) đã nghiên cứu và tạo nên công trình, phương pháp, thuật toán Marching Cubes mang tính cách mạng. Thuật toán trên tập trung vào việc tái tạo lưới (mesh) ở lớp ngoài của vật thể. Vì vật thể sẽ đi qua một số đỉnh trong một khối lập phương. Để dễ hình dung, chúng tôi lấy ví dụ thuật toán Marching Squares: gọi A là điểm vật đi qua (vật bên trên) và B là điểm vật thể không đi qua (vật bên dưới), vậy điểm C nằm giữa hai điểm A và B có khả năng là điểm giao với vật. Tương tự với thuật toán Marching Cubes trong không gian 3D. Nếu 1 đỉnh có hai trường hợp là được qua hoặc không được di qua thì có đến $2^8 = 256$ trường hợp có thể xảy ra. Tuy nhiên nếu sử dụng phép xoay trong không gian thì ta chỉ còn lại 15 trường hợp. Vậy ta sẽ có 15 trường hợp thể tích của một khối lập phương cộng thêm trường hợp thể tích bằng 1 khi khôi lập phương nằm hoàn toàn trong vật thể. Vậy khi cấu trúc mesh này bị thay đổi ta chỉ cần sử dụng hàm cập nhật theo vùng với Cây Chỉ Số Nhị Phân và truy vấn tổng thể tích mới với độ phức tạp thấp trong thời gian ngắn. Vì trong thuật toán Marching Cubes đã có sẵn 3 vòng lặp duyệt hết một không gian lập phương 3D để tái tạo cấu trúc 3D nên ta sẽ tận dụng 3 vòng lặp này để đồng thời khởi tạo cây Fenwick (cây chỉ số nhị phân).

Tài Liệu Tham Khảo

Fenwick, P. M. (1994). A new data structure for cumulative frequency tables. Software: Practice and experience, 24(3), 327-336.

Lorensen, W. E., & Cline, H. E. (1998). Marching cubes: A high resolution 3D surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field* (pp. 347-353).

Liu, Y. S., Yi, J., Zhang, H., Zheng, G. Q., & Paul, J. C. (2010). Surface area estimation of digitized 3D objects using quasi-Monte Carlo methods. *Pattern Recognition*, *43*(11), 3900-3909.