

An Overview of Parallel Computing Models for Image Segmentation with Ncut

Nhu Y Tran^{1,2}, Trung Hieu Huynh¹, The Bao Pham^{3,*} 

¹Industrial University of Ho Chi Minh City, Vietnam

²Ho Chi Minh City University of Industry and Trade, Vietnam

³Sai Gon University, Ho Chi Minh City, Vietnam

*Corresponding author. Email: ptbao@sgu.edu.vn

ARTICLE INFO

Received: 24/03/2023
Revised: 26/04/2023
Accepted: 02/11/2023
Published: 28/04/2024

KEYWORDS

Parallel model;
Parallel computing;
GPU;
CPU;
Ncut.

ABSTRACT

Image segmentation is a prerequisite in most image processing applications. There are lots of methods for image segmentation, and there are also numerous methods to evaluate the results of such segmentations. Among them, the Ncut algorithm by J. Shi using graph theory has significantly improved the efficiency of digital image processing. Most of the results indicate that the partitions of the image are all according to human vision. However, with a large image set, the algorithm will compute at a low speed; it takes time and high memory usage for the computation. The parallel model is a model that scientists are interested in and use to improve performance in image segmentation with large-size images. The article summarizes the overview of parallel models in image segmentation and the comments and evaluations for a few parallel models on the Ncut algorithm. Experimental results reveal that the time required to compute eigenvalues in the Ncut algorithm when executed in parallel on a GPU is significantly lower compared to execution on a CPU. Furthermore, as image sizes increase, the execution time on the GPU only marginally increases in comparison to CPU execution, while still yielding similar image segmentation results.

Tổng Quan Mô Hình Tính Toán Song Song Với Ncut cho Bài Toán Phân Đoạn Ảnh

Trần Như Ý^{1,2}, Huỳnh Trung Hiếu¹, Phạm Thế Bảo^{3,*} 

¹Trường Đại học Công nghiệp Thành phố Hồ Chí Minh, Việt Nam

²Trường Đại học Công Thương Thành phố Hồ Chí Minh, Việt Nam

³Trường Đại học Sài Gòn, Thành phố Hồ Chí Minh, Việt Nam

*Tác giả liên hệ. Email: ptbao@sgu.edu.vn

THÔNG TIN BÀI BÁO

Ngày nhận bài: 24/03/2023
Ngày hoàn thiện: 26/04/2023
Ngày chấp nhận đăng: 02/11/2023
Ngày đăng: 28/04/2024

TỪ KHÓA

Mô hình song song;
Tính toán song song;
GPU;
CPU;
Ncut.

TÓM TẮT

Phân đoạn ảnh là điều kiện tiên quyết trong hầu hết các ứng dụng xử lý ảnh. Có nhiều phương pháp để phân đoạn ảnh và một lượng lớn các phương pháp để đánh giá kết quả phân đoạn được đề xuất. Trong số đó, thuật toán Ncut của tác giả J. Shi sử dụng lý thuyết đồ thị đã mang lại hiệu quả đáng kể trong xử lý ảnh số, hầu hết các kết quả đều thể hiện được các phân vùng theo thị giác con người. Tuy nhiên khi tập ảnh lớn thuật toán thực thi với tốc độ chậm, mất nhiều thời gian và chiếm nhiều bộ nhớ trong tính toán. Mô hình song song là mô hình được các nhà khoa học quan tâm và sử dụng nhằm nâng cao hiệu suất trong phân đoạn hình ảnh với kích thước lớn. Bài báo tóm tắt tổng quan mô hình song song trong phân đoạn ảnh và những nhận xét, đánh giá về một số mô hình song song trên thuật toán Ncut. Kết quả thực nghiệm thời gian tìm trị riêng trong thuật toán Ncut khi song song trên GPU ít hơn đáng kể so với thời gian thực hiện trên CPU. Ngoài ra, khi kích cỡ ảnh tăng dần thì thời gian chạy trên GPU tăng không nhiều so với thực hiện trên CPU với kết quả phân đoạn ảnh gần giống nhau.

Doi: <https://doi.org/10.54644/jte.2024.1370>

Copyright © JTE. This is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

1. Mô hình tính toán song song

Tính toán song song trên dữ liệu ảnh đóng một vai trò không thể thiếu trong lĩnh vực xử lý hình ảnh và khoa học máy tính. Đây là việc sử dụng đồng thời nhiều tài nguyên tính toán để gia tăng hiệu suất trong việc xử lý dữ liệu ảnh. Thông thường, điều này được thực hiện bằng cách tận dụng nhiều lõi xử lý hoặc đơn vị tính toán đồng thời GPU. Một trong những ứng dụng quan trọng của tính toán song song trong xử lý hình ảnh là phân đoạn ảnh thành các phần nhỏ, nhưng vẫn giữ được thông tin cấu trúc và ý nghĩa của hình ảnh. Tính toán song song không chỉ giúp cải thiện hiệu suất xử lý, mà còn mở ra nhiều cơ hội trong việc phát triển ứng dụng và dịch vụ dựa trên hình ảnh.

1.1. Các hướng tiếp cận song song trong phân đoạn ảnh

Trong quá trình phân đoạn ảnh với dữ liệu ảnh có độ phân giải cao, việc tính toán song song trên dữ liệu ảnh được chia thành ba hướng tiếp cận. Hướng tiếp cận xử lý hệ thống cụm nhiều máy làm việc cùng nhau trong một mạng cục bộ tốc độ cao dựa trên nền tảng công nghệ Hadoop, MapReduce và Spark [1]-[6] và được dựng để lập trình song song trong việc phân đoạn ảnh như hình 2. Theo Zhenhua Lv và cộng sự [1] đã sử dụng thuật toán Kmeans tuần tự và song song trên Hadoop và MapReduce để phân tích ảnh viễn thám. Trong quá trình tiền xử lý ảnh, bài báo thực hiện chuyển đổi từng pixel từ giá trị hệ màu RGB sang hệ màu LAB phù hợp hơn để phân biệt màu sắc. Phân cụm được đánh giá dựa trên tổng phương sai tối thiểu cho mỗi cụm và các báo cáo về hiệu suất dựa trên các thử nghiệm được đưa ra. Tác giả [2] đề xuất phân đoạn ảnh nông nghiệp bằng thuật toán phân đoạn FCM song song dựa trên nền tảng tính toán bộ nhớ phân tán Apache Spark cho dữ liệu lớn hình ảnh nông nghiệp. Đầu tiên, hình ảnh đầu vào được chuyển đổi từ không gian màu RGB sang không gian màu LAB. Sau đó, dữ liệu ảnh được phân vùng và lưu trữ trong các nút khác nhau, quá trình tính toán các điểm pixel đến các trung tâm cụm khác nhau được tính cho đến khi điều kiện dừng được thỏa mãn. Cuối cùng, dữ liệu ảnh được khôi phục sau khi phân nhóm để tái tạo lại hình ảnh đã phân đoạn. Trên nền tảng Spark, hiệu suất của thuật toán FCM song song được đánh giá và đạt tốc độ trung bình là 12,54 trên mười nút máy tính. Kết quả thử nghiệm cho thấy thuật toán FCMs song song dựa trên Spark có thể tăng tốc độ đáng kể và bộ thử nghiệm hình ảnh nông nghiệp mang lại hiệu suất cải thiện tốt hơn 128% so với phương pháp dựa trên Hadoop.

Hướng tiếp cận song song dựa trên mô hình lai CPU-GPU với hai mô hình lập trình song song phổ biến. Cụ thể, mô hình Open Multi-Processing (OpenMP) lập trình song song dựa trên đồng bộ luồng, cho phép chia sẻ bộ nhớ và thực hiện tính toán trên các luồng xử lý khác nhau; và mô hình Compute Unified Device Architecture (CUDA) lập trình song song được sử dụng trong tính toán đồ họa, cho phép các lõi xử lý trong GPU thực hiện tính toán và xử lý đa luồng [7]-[12]. Tác giả Baydoun và cộng sự [9] đã triển khai thuật toán song song bằng K-mean với dữ liệu hình ảnh RGB trên CPU và GPU. Trong CPU, họ sử dụng Cilk Plus (một thư viện dùng để song song hóa do Intel phát triển) và OpenMP. Các thuộc tính của dữ liệu đầu vào như số lượng cụm, số lượng mẫu và số lượng các đối tượng địa lý có ảnh hưởng đến hiệu suất của thuật toán song song K-mean. Cuối cùng, nhóm tác giả nhận thấy rằng OpenMP phù hợp hơn Cilk Plus cho tác vụ song song hóa. Sirotkovi và cộng sự [7] đã trình bày phân đoạn hình ảnh bằng thuật toán K-mean sử dụng CUDA trên GPU. Dữ liệu hình ảnh có độ phân giải 512×512 pixel được sử dụng làm đầu vào cho phân đoạn. Kết quả cho thấy sự cải thiện về thời gian thực thi của GPU so với phiên bản tuần tự của thuật toán được triển khai trên CPU. Baker và Balhaf [8] đã thực hiện phân đoạn hình ảnh bạch cầu bằng cách tích hợp CPU và GPU. Quá trình chuyển đổi không gian màu từ RGB sang HIS và triển khai K-means được thực hiện tính toán song song trên GPU và việc trích xuất màu được thực hiện trên CPU. Kết quả phương pháp kết hợp được đề xuất có hiệu suất hiệu quả hơn trên CPU và GPU riêng.

Hướng tiếp cận hiện thực song song dựa trên tính toán lượng tử. Phân đoạn ảnh bằng phương pháp phân cụm được xem là bài toán tối ưu hóa tổ hợp và tốn nhiều chi phí. Cụ thể, các thuật toán di truyền tổ hợp thường tiêu tốn thời gian, tốc độ hội tụ chậm và giá trị tìm được thường là giá trị tối ưu cục bộ. Tính toán lượng tử được biết đến với các nguyên tắc của cơ học lượng tử như sự không chắc chắn, chồng chất, vướng víu và song song hóa ngầm định [13]-[17]. Những đặc tính này làm cho lượng tử có tính đa dạng tốt hơn và cân bằng giữa việc khám phá và khai thác tốt hơn so với các thuật toán tối ưu hóa tổ hợp thông thường. Theo nhóm tác giả Yangyang Li [14] một cách tiếp cận phân cụm đa mục tiêu mới

được đề xuất tận dụng lợi thế của cơ chế tối ưu hóa đa mục tiêu và sự chồng chất của các trạng thái lượng tử. Do đó thuật toán QMEC có khả năng tìm kiếm, đa dạng phần tử dự tuyến và đạt được hiệu suất cao về tỷ lệ phân cụm chính xác và hiệu ứng hình ảnh. Theo nhóm tác giả Sunanda Das [16], thuật toán QIANA-based FRCM được trình bày dựa trên sự kết hợp của thuật toán di truyền và lượng tử để phân đoạn hình ảnh từ thang đo độ xám nhiều cấp. Các chỉ số đánh giá chất lượng cho thấy thời gian tính toán giảm và chất lượng đầu ra được phân đoạn tốt hơn. Cải tiến của thuật toán này nằm ở việc cải thiện khởi tạo quần thể và phân lai chéo quần thể.

1.2. Mô hình tính toán song song trong phân đoạn ảnh

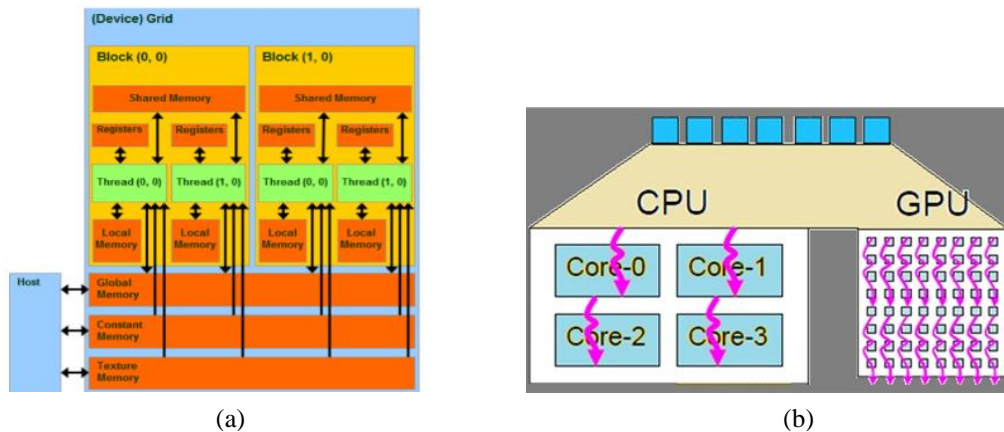
Trong quá trình phân đoạn ảnh, tính toán song song là một phương pháp quan trọng để gia tăng hiệu suất xử lý dữ liệu hình ảnh. Dưới đây là các mô hình tính toán song song trong phân đoạn ảnh.

1.2.1. Phân đoạn ảnh trên thiết bị đồ họa

Một trong những cách phổ biến nhất để tận dụng tính toán song song trong phân đoạn ảnh là sử dụng GPU. GPU có nhiều lõi tính toán và khả năng xử lý song song mạnh mẽ, đặc biệt là trong việc thực hiện các phép tính trên ma trận. Mô hình lập trình song song trên thiết bị đồ họa ra đời vào cuối những năm 1990 như một bộ xử lý chuyên dụng nhằm tăng tốc xử lý đồ họa, hỗ trợ lập trình song song ở mức cao, và xử lý dữ liệu lớn. GPU tích hợp hàng trăm lõi xử lý tính toán, do đó hiệu suất của nó cao hơn rất nhiều so với CPU trong các thuật toán phức tạp đòi hỏi hiệu năng cao. Kiến trúc bộ nhớ của GPU được chia thành nhiều loại vùng nhớ với các chức năng truy cập và lưu trữ khác nhau, được minh họa trong hình 1a. Kiến trúc này bao gồm các loại bộ nhớ sau:

- Bộ nhớ thanh ghi (Registers): Dành cho truy cập đọc và ghi của các tiểu trình và chỉ riêng nội bộ trên một tiểu trình.
- Bộ nhớ cục bộ (Local memory): Cho phép truy cập trên nội bộ của mỗi tiểu trình.
- Bộ nhớ chia sẻ (Shared memory): Cho phép truy cập của một nhóm tiểu trình (gọi là khối tiểu trình) trong cùng một khối.
- Bộ nhớ toàn cục (Global memory): Cho phép truy cập đọc và ghi từ tất cả các tiểu trình trên GPU.
- Bộ nhớ hằng (Constant memory): Cho phép truy cập dữ liệu chỉ đọc từ các tiểu trình trong cùng grid.
- Vùng nhớ Texture: Là vùng nhớ chỉ đọc với tốc độ truy cập nhanh hơn so với bộ nhớ chia sẻ.

Tùy theo cách truy xuất của các tiểu trình và khối tiểu trình (block, grid), các loại vùng nhớ này có quyền truy cập khác nhau. Lập trình song song trên GPU đòi hỏi kỹ năng chuyên môn cao và thường sử dụng các thư viện đặc biệt như CUDA (dành cho GPU của NVIDIA) hoặc OpenCL (hỗ trợ nhiều loại GPU) để viết mã tùy chỉnh và thực hiện phân đoạn ảnh trên GPU. Việc này có thể làm tăng độ phức tạp của mã nguồn và đôi khi khó trong việc kiểm tra và bảo trì.



Hình 1. (a) Kiến trúc bộ nhớ GPU [18] và (b) Hệ thống lai CPU+GPU [19]

1.2.2. Phân đoạn ảnh trên CPU đa lõi

Sử dụng nhiều lõi CPU đồng thời xử lý nhiều phần của ảnh là một phương pháp thường được áp dụng. Trong lĩnh vực khoa học và công nghệ, nhu cầu về tính toán đang tăng cao chưa từng thấy, nhưng việc sử dụng phương pháp xử lý tuần tự trên một bộ xử lý đơn đã trở thành một thách thức lớn. Để giải quyết vấn đề này, cần sử dụng tính toán song song trên các hệ thống máy tính đồng thời và siêu máy tính. Mặc dù hiệu suất xử lý của các bộ xử lý đã được cải thiện đáng kể, tuy nhiên do các hạn chế về mật độ vật lý, tốc độ xử lý vẫn bị giới hạn ở một mức độ nào đó. Các nhà sản xuất đang quan tâm đến việc phát triển bộ xử lý tích hợp nhiều đơn vị xử lý trên một chip để thúc đẩy sự tiến bộ cơ bản trong công nghệ phần cứng. Sử dụng các bộ xử lý đa lõi có thể biến máy tính thành một hệ thống song song nhỏ. Trong thực tế, các hệ thống với bộ xử lý đơn lõi không đáp ứng được yêu cầu về thời gian, do đó cần sử dụng các hệ thống đa bộ xử lý. Tính toán song song giúp các bộ xử lý hợp tác để cùng tính toán và giải quyết vấn đề. Sử dụng CPU đa lõi và nhiều nhân có thể tận dụng cho mô hình công việc song song cho nhiều nhiệm vụ, trong đó các nhiệm vụ này có các chỉ lệnh khác nhau trên tập dữ liệu khác nhau. Các thư viện và framework như OpenMP cho phép bạn dễ dàng tận dụng tính năng đa lõi của CPU để thực hiện phân đoạn ảnh trên các luồng đa lõi khác nhau.

1.2.3. Phân đoạn ảnh phân tán

Trong các hệ thống có nhiều máy tính kết nối với nhau qua mạng, có thể áp dụng tính toán phân tán để thực hiện phân đoạn ảnh. Các máy tính này chia sẻ công việc phân đoạn và kết hợp kết quả để hoàn thành nhiệm vụ. Mô hình lập trình song song trên hệ thống cụm nhiều máy tính hoạt động cùng nhau trong một mạng cục bộ tốc độ cao. Một lợi ích khi sử dụng hệ thống song song này là các hệ điều hành đều hỗ trợ đa nhiệm, cho phép áp dụng phương pháp lập trình song song trên chúng. Tuy nhiên, thách thức đặt ra là làm thế nào để các bộ xử lý độc lập có thể hợp tác và cùng tham gia giải quyết một vấn đề. Mặc dù vậy, hệ thống cụm nhiều máy vẫn có nhược điểm cơ bản từ góc độ hiệu quả khai thác và sử dụng, bao gồm giá thành, tỷ lệ hiệu năng sử dụng, và chi phí vận hành (bao gồm năng lượng, quản trị, và bảo trì hệ thống) rất cao. Hơn nữa, nếu không phân phối công việc một cách hợp lý, các nút xử lý trong cụm có thể gặp tình trạng tắc nghẽn và dẫn đến giảm hiệu suất của toàn bộ hệ thống. Cuối cùng, việc đồng bộ hóa dữ liệu giữa các nút xử lý trong một cụm có thể là một thách thức lớn, đặc biệt là khi các nút xử lý cần truy cập vào các tài nguyên chung như cơ sở dữ liệu hoặc bộ nhớ chia sẻ. Các framework như Apache Hadoop hoặc Apache Spark cung cấp khả năng tính toán phân tán để thực hiện phân đoạn ảnh trên nhiều máy chủ.

1.2.4. Phân đoạn ảnh với mô hình lai trên CPU – GPU

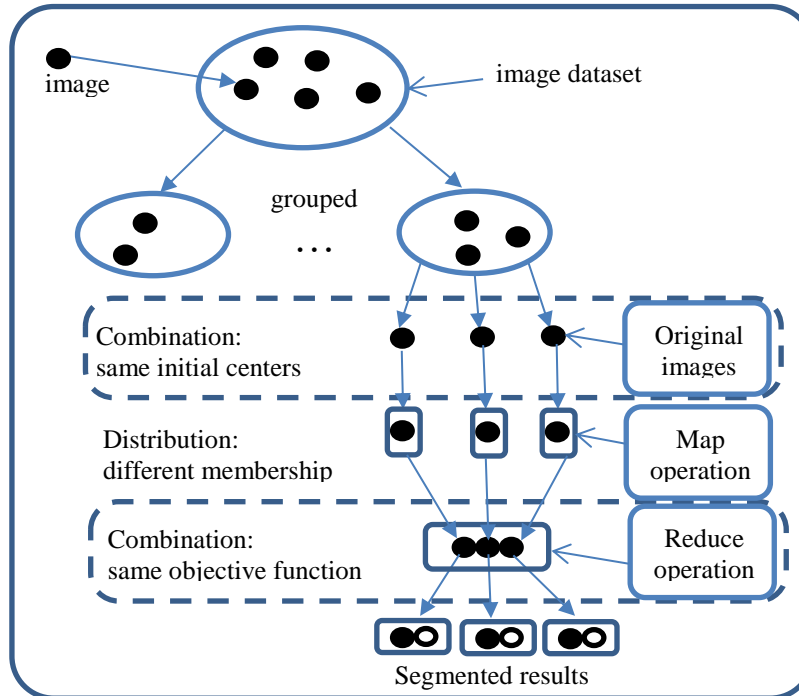
Mô hình lập trình song song với mô hình lai trên CPU – GPU [19] là một hệ thống kết hợp xử lý mạnh mẽ vì CPU và GPU có những thuộc tính kết hợp sử dụng cả hai loại bộ xử lý bổ sung nhau cho phép thực thi nhiều ứng dụng lớn đạt hiệu năng tối ưu. Chip CPU đa nhân đảm trách việc xử lý tuần tự, trong khi đó chip xử lý GPU với hàng ngàn nhân xử lý được thiết kế cho phần tính toán và xử lý song song. Hệ thống lai sẽ là kiến trúc tính toán đầy hứa hẹn trong tương lai giúp tăng tốc nhiều ứng dụng lớn trong tính toán hiệu năng cao. Theo hình 1b kiến trúc mô hình lai CPU+GPU thực thi một số luồng chạy đồng thời trong hệ thống ở cấp độ CPU. Ngoài ra, đối với mỗi GPU trong hệ thống, một luồng cụ thể được tạo, luồng này sẽ chịu trách nhiệm xử lý dữ liệu trên GPU. Khi một luồng hoàn thành công việc, chúng được chỉ định một phiên mới bởi kỹ thuật cân bằng tải động. Tuy nhiên, khó khăn trong việc đồng bộ hóa giữa CPU và GPU đòi hỏi phải quản lý tốt thời gian và tài nguyên để đảm bảo rằng các nhiệm vụ được thực hiện một cách đồng bộ và hiệu quả. Vì vậy, việc đồng bộ hóa này khó khăn do CPU và GPU có tốc độ và cách thức xử lý khác nhau.

Ví dụ về phân đoạn ảnh bằng việc lập trình song song giữa CPU và GPU. Đầu tiên, chọn một mô hình mạng nơ-ron tích chập phù hợp cho nhiệm vụ phân đoạn ảnh. Sau đó, sử dụng TensorFlow hoặc một thư viện tương tự để tạo và đào tạo mạng CNN trên GPU. Tiếp theo, phân chia công việc giữa CPU và GPU có thể tiến xử lý ảnh và chia nó thành các phần nhỏ trên CPU, sau đó gửi chúng cho GPU để phân đoạn bằng mạng CNN. Trong khi GPU đang xử lý một phần của ảnh, chúng có thể tiếp tục xử lý các phần còn lại trên CPU. Cuối cùng, sau khi phân đoạn ảnh bằng mạng CNN trên GPU, chúng ta kết hợp kết quả này với phần được xử lý trên CPU để tạo ra kết quả phân đoạn hoàn chỉnh.

1.2.5. Phân đoạn ảnh dựa trên lượng tử

Lý thuyết lượng tử là một nhánh của vật lý và toán học tập trung vào nghiên cứu các hiện tượng xảy ra ở mức độ nguyên tử và phân tử. Trong việc áp dụng lý thuyết lượng tử vào phân đoạn ảnh, chúng ta sử dụng đại lượng qubit lượng tử để biểu diễn thông tin trong hình ảnh. Mỗi qubit có khả năng mang thông tin về một pixel hoặc một vùng trong hình ảnh. Sử dụng phép tính lượng tử cho phép chúng ta thực hiện các phép tính đồng thời trên nhiều trạng thái của hình ảnh, từ đó giúp tăng cường tốc độ của quá trình phân đoạn. Các phép tính lượng tử cơ bản bao gồm biến đổi Fourier lượng tử, thuật toán Grover và thuật toán lượng tử liên quan đến ma trận.

Ứng dụng lượng tử trong phân đoạn ảnh có khả năng cải thiện đáng kể chính xác và tốc độ của quá trình phân đoạn hình ảnh. Điều này thể hiện qua việc cải thiện trong các nhiệm vụ như nhận dạng đối tượng, phát hiện biên, phân loại hình ảnh và nhiều ứng dụng khác. Tuy nhiên, phân đoạn ảnh dựa trên lượng tử cũng đối diện với một số thách thức, chẳng hạn như độ phức tạp của các phép tính lượng tử và khả năng triển khai chúng một cách hiệu quả trên máy tính lượng tử. Tuy vậy, với sự phát triển của máy tính lượng tử, chúng ta có cơ hội giải quyết những thách thức này và mở ra nhiều tiềm năng hứa hẹn cho việc phân đoạn ảnh dựa trên lượng tử.



Hình 2. Mô hình xử lý phân đoạn ảnh trên MapReduce[1]

2. Thuật toán Ncut

Thuật toán Ncut [20] sử dụng lý thuyết đồ thị để tìm nhóm các điểm ảnh trong không gian đặc trưng của ảnh số. Tập hợp những điểm ảnh trong không gian đặc trưng bất kỳ được biểu diễn bằng đồ thị trọng số vô hướng $G = (V, E)$. Trong đó, tập đỉnh V của đồ thị là những điểm ảnh hoặc vùng trong không gian Euclide. Mỗi cạnh $(v_i, v_j) \in E$ có trọng số w_{ij} thể hiện mối quan hệ hoặc tính chất giữa hai điểm được nối, cũng được xây dựng bằng hàm tính độ tương đồng giữa hai điểm ảnh i và j . Thuật toán Ncut dùng để phân đoạn một ảnh thành những vùng khác nhau có thuộc tính tương tự. Các phân vùng được tìm dựa vào độ tương đồng cao giữa các đỉnh trong cùng một tập đồ thị và độ tương đồng thấp giữa các đỉnh nằm trong các tập đồ thị khác nhau.

Theo tác giả Shi và Malik đã đề xuất tiêu chuẩn Ncut, thay vì chỉ quan tâm đến giá trị tổng trọng lượng các cạnh giữa hai vùng, tiêu chuẩn Ncut tính giá trị 'cut' theo tỉ số trên tổng trọng lượng kết nối đến tất cả các đỉnh trong đồ thị theo công thức (1).

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \quad (1)$$

Trong đó, $assoc(X, V) = \sum_{u \in X, v \in V} w(u, v)$ là tổng trọng lượng của các kết nối từ đỉnh trong A đến tất cả các đỉnh trong đồ thị.

Với hơn hai vùng thì tiêu chuẩn Ncut được mở rộng theo công thức (2).

$$Ncut(A_i / i = 1..n) = \sum_{i=1}^N \frac{cut(A_i, \bar{A}_i)}{assoc(A_i, V)} \quad (2)$$

Giả sử tập đỉnh V là một vùng của đồ thị, cần tách thành hai tập con A, B. Đặt x là vector có $N=|V|$ chiều, với $x_i=1$ nếu đỉnh $i \in A$ và ngược lại $x_i=-1$. Đặt $d(i) = \sum_j w(i, j)$ là tổng kết nối từ đỉnh i đến tất cả các đỉnh khác.

Đặt $W=(w_{ij})$ là ma trận đối xứng cỡ $N \times N$ và $D=diag(d_1, d_2, \dots, d_n)$ là ma trận đường chéo kích thước $N \times N$. Qua nhiều bước biến đổi toán học, tiêu chuẩn Ncut trở thành vấn đề trong giải bài toán đại số theo công thức (3).

$$Ncut(A, B) = \frac{y^T(D - W)y}{y^T D y}, y_i \in \{1, b\}, y^T D 1 = 0 \quad (3)$$

Trong đó, $y = (1 + x) - b(1 - x)$ và $b = k / (1 - k)$, với $k = \sum_{x_i > 0} d_i / \sum_i d_i$.

Ta có thể cực tiểu Ncut (A,B) theo công thức (4).

$$\min_{A, B} Ncut(A, B) = \min_y \frac{y^T(D - W)y}{y^T D y} \quad (4)$$

Phương trình này có thể được giải quyết bằng việc giải bài toán trị riêng tổng quát theo công thức (5).

$$(D - W)y = \lambda D y \quad (5)$$

Thuật toán Ncut Giải thuật gom nhóm trong phân đoạn ảnh:

Cho 1 ảnh I. Xây dựng đồ thị trọng số $G = (V, E)$ với mỗi node của đồ thị là một điểm ảnh của ảnh. Đặt N là số node của đồ thị $N = |V|$.

Bước 1: Xây dựng ma trận tương đồng đối xứng W kích thước $N \times N$ với $W(i, j) = w_{ij}$ theo công thức (6).

$$w_{ij} = \exp \frac{-\|F(i) - F(j)\|_2^2}{\sigma_f^2} * \begin{cases} \exp \frac{-\|X(i) - X(j)\|_2^2}{\sigma_x^2} & \text{nếu } \|X(i) - X(j)\|_2 < r \\ 0 & \text{nếu } \|X(i) - X(j)\|_2 \geq r \end{cases} \quad (6)$$

Trong đó, $X(i)$ là tọa độ không gian của ảnh gốc I, và $F(i)$ là vector đặc trưng được định nghĩa như sau:

$F(i) = I$ đối với tập điểm phân đoạn.

$F(i) = I(i)$ giá trị cường độ cho những ảnh xám.

$F(i) = [v, u, \sin(h), v, s, \cos(h)](i)$ với h, s, v là giá trị HSV đối với phân đoạn ảnh màu.

Đặt $d_i = \sum_j w(i, j)$ là tổng các kết nối từ đỉnh i đến tất cả các đỉnh còn lại. Xây dựng ma trận đường chéo D với d_i là giá trị đường chéo.

Bước 2: Giải hệ trị riêng tổng quát $(D - W)x = \lambda D x$ và tìm vector riêng có trị riêng nhỏ thứ hai.

Bước 3: Dùng vector riêng này chia đôi đồ thị. Trong trường hợp lý tưởng, vector riêng này chỉ nhận hai giá trị rời rạc, và dựa vào dấu để xác định phân vùng (ví dụ $A = \{V_i / y_i > 0\}$ và $B = \{V_i / y_i < 0\}$). Tuy nhiên y nhận những giá trị thực liên tục, vì thế cần chọn điểm chia phù hợp bằng phương pháp heuristic để tìm kiếm điểm chia sao cho giá trị Ncut cực tiểu.

Bước 4: Lặp lại đệ qui chia đôi mỗi phân vùng cho đến khi giá trị Ncut lớn hơn ngưỡng cho trước.

3. Thuật toán Neut với mô hình song song cho phân đoạn ảnh

Quá trình song song với mô hình clustering sẽ mất nhiều thời gian chuyển dữ liệu giữa các node nên mô hình này không hữu ích cho phân đoạn ảnh, vì phải tính ma trận nhiều lần. Mô hình song song trên CPU sẽ không đủ bộ nhớ để tính toán nên phải đợi chờ các task ở các CPU khác. Với mô hình song song trên GPU cho thấy nó hiệu quả hơn CPU rất nhiều trong nhiều thuật toán phức tạp đòi hỏi hiệu năng cao. Ngoài ra, song song với công nghệ phần cứng chuyên biệt GPU, các ngôn ngữ hỗ trợ lập trình song song hóa cũng được phát triển và hỗ trợ mạnh mẽ trên thiết bị GPU như CUDA, Matlab hỗ trợ lập trình song song trên GPU.

Hiện thực phân đoạn ảnh bằng Neut trên GPU bởi kiến trúc CUDA [21]-[25]. Cụ thể, theo nhóm tác giả Senthilnath và cộng sự [22] thực hiện song song trên CUDA trong quá trình tính ma trận tương đồng W . Ngoài ra, nhóm tác giả Huang [21] sử dụng thư viện CUBLAS trong NVIDIA để tăng tốc tính vector riêng và trị riêng từ ma trận trong công thức (7) để phân đoạn ảnh. Kết quả thuật toán đáng giá hiệu suất song song cho kết quả phân đoạn nhanh khoảng 2.3 lần. Tuy nhiên, khi kích thước hình ảnh lớn quá trình tính vector riêng và trị riêng giảm độ chính xác. Do đó phương pháp trên cần bổ sung kỹ thuật tinh chỉnh trong quá trình tính vector riêng.

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}x = \lambda x \tag{7}$$

$$\text{Hoặc } Mx = \lambda x \text{ với } M = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}$$

Nhóm tác giả [26]-[29] phát triển một cách tiếp cận mới để cung cấp hiệu quả và phân đoạn mạnh mẽ các hình ảnh màu. Bằng cách kết hợp phân đoạn Mean shift (MS) [26], watersheds [27] với phương pháp phân vùng Neut giảm độ phức tạp tính toán và khả thi để xử lý phân đoạn ảnh theo thời gian thực. Nó xử lý trước một hình ảnh bằng cách sử dụng thuật toán MS để tạo thành các vùng được phân đoạn nhằm bảo toàn các đặc tính hình ảnh. Sau đó, các vùng được phân đoạn được biểu diễn bằng cách sử dụng cấu trúc đồ thị và phương pháp Neut được áp dụng để thực hiện phân nhóm được tối ưu hóa toàn cục. Ngoài ra, việc phân cụm hình ảnh bằng cách sử dụng các vùng được phân đoạn, thay vì các pixel hình ảnh, cũng làm giảm độ nhạy với nhiễu và dẫn đến hiệu suất phân đoạn hình ảnh được nâng cao.

Ngoài ra, nhóm tác giả [30] sử dụng phương pháp song song Multiscale Neut để xây dựng ma trận tương đồng bằng cách tạo ma trận phân vùng X với ma trận tương đồng W của thuật toán Neut. Kết quả so sánh với các thuật toán khác cũng đã được nhiều bài báo so sánh thực nghiệm tại [30]-[32].

4. Thực nghiệm và đánh giá kết quả

Môi trường cài đặt: Chúng tôi thực nghiệm trên máy tính sử dụng hệ điều hành Windows 10 Home bản 64 bit, RAM 16 GB, GPU (NVIDIA GeForce GTX 1660Ti) 6GB, Chip Intel Core (TM) i7-9750H CPU @ 2.6GHz. Ngôn ngữ lập trình python phiên bản 3.6.2.

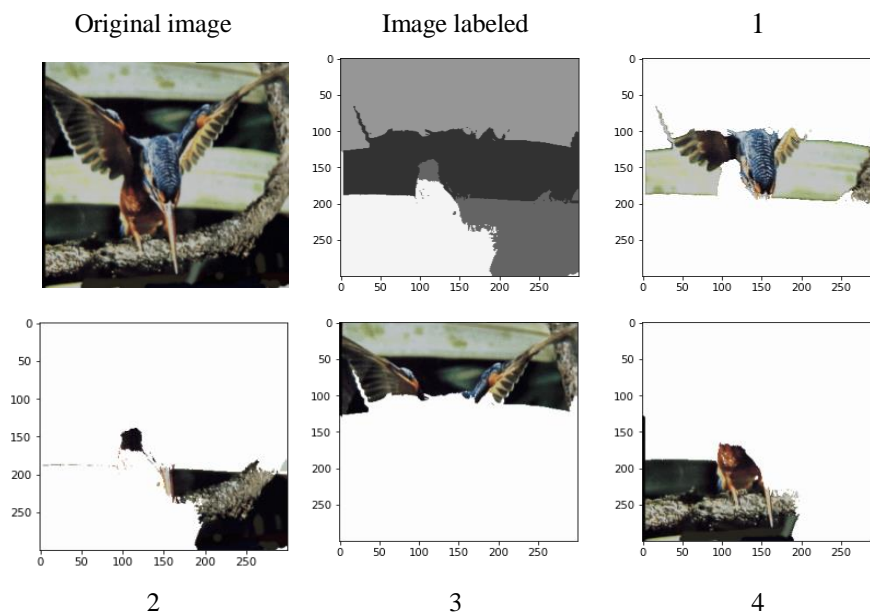
Bảng 1. Thời gian tìm trị riêng trong thuật toán Neut trên CPU và song song trên GPU

Kích thước ảnh	Thời gian		
	Tính trị riêng trong thuật toán Neut trên CPU	Tính trị riêng với k-mean trong thuật toán Neut trên CPU	Tính trị riêng trong thuật toán Neut trên GPU
128x128	1,6	0,5	0,6
140x140	2,3	0,8	0,8
200x200	6,2	2,3	0,9
256x256	17,3	3,5	0,9
512x512	NA	17,8	1,2
1024x1024	NA	70,4	2,8

Chúng tôi thực nghiệm trên tập dữ liệu ảnh [33] cho quá trình thực thi phần tính vector riêng bằng thuật toán Neut trên CPU [20] và song song trên GPU. Chúng tôi cài đặt thực thi với ảnh có kích cỡ tăng

dẫn như bảng 1 và cố định với số phân vùng là 4 phân vùng, tương ứng với việc tìm 4 trị riêng và vector riêng của ma trận W . Bảng 1 cho thấy thời gian tìm trị riêng song song GPU ít hơn đáng kể so với thời gian thực hiện trên CPU. Và khi kích cỡ ảnh tăng dần thì thời gian tính toán song song trên GPU tăng không nhiều so với thực hiện trên CPU. Ngoài ra, trong hình 3, 4 và 5 cho thấy kết quả giống nhau khi thực thi phân đoạn ảnh với 4 phân đoạn trên CPU và GPU.

Trong thuật toán Ncut cần giải quyết vấn đề tìm trị riêng của ma trận trọng số W . Dưới góc nhìn của phương pháp phân vùng quang phổ, bài toán đưa về giải quyết vấn đề tìm k vector riêng nhỏ nhất của ma trận Laplace của ma trận trọng số W . Vì ma trận trọng số W là ma trận thưa đối xứng có kích cỡ tương đối lớn nên thời gian tính vector riêng chiếm khá nhiều thời gian trong toàn bộ quá trình phân đoạn ảnh Ncut. Đặc biệt với kích thước ảnh càng lớn thì kích cỡ ma trận cũng tăng đáng kể, vì vậy cần tăng hiệu suất bằng phương pháp song song hóa Lanczos cho phép tìm vector riêng của ma trận W trên GPU và sau đó hiệu chỉnh giá trị xấp xỉ trong quá trình tìm vector riêng.



Hình 3. Kết quả phân đoạn ảnh với Ncut trên CPU

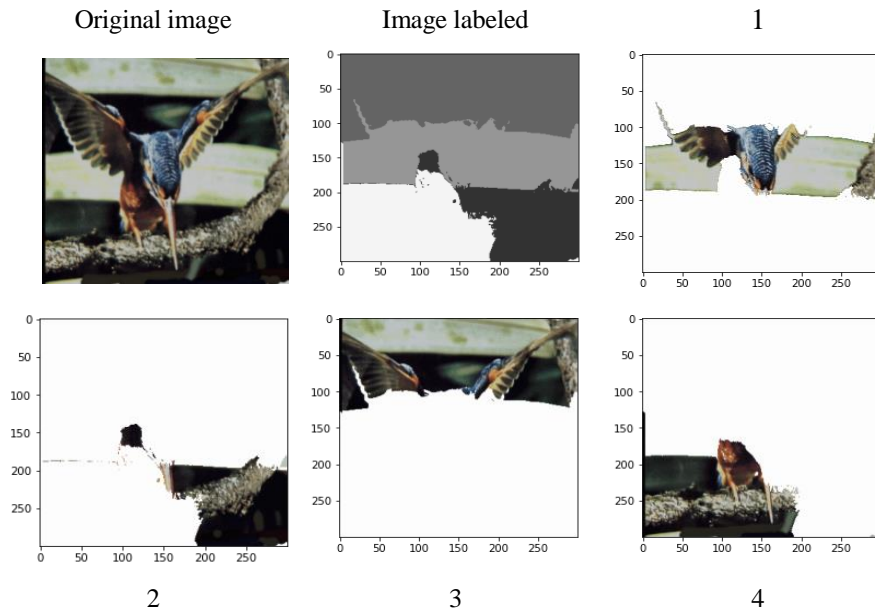
Gom nhóm là bước sau cùng để trích ra các phân vùng từ ma trận vector riêng với giá trị thực trong quá trình phân đoạn của thuật toán Ncut. Việc song song hóa thuật toán gom nhóm cũng rất cần thiết cho việc tăng hiệu suất phân đoạn.

5. Kết luận

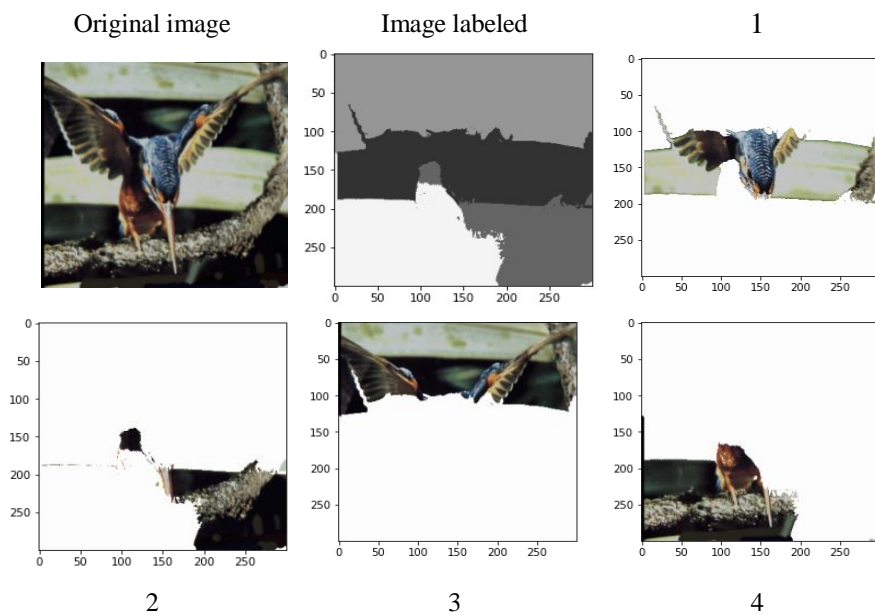
Khi chuyển từ việc tính toán trên CPU sang tính toán song song trên GPU, chúng ta quan sát thấy rằng thời gian tìm trị riêng trên GPU ít hơn đáng kể so với thời gian thực hiện trên CPU. Điều này cho thấy sức mạnh tính toán của GPU trong việc xử lý các phép tính ma trận lớn cần thiết trong thuật toán Ncut. Khi làm việc với ảnh kích thước lớn, sự chênh lệch này trở nên đặc biệt quan trọng để cải thiện hiệu suất và tăng tốc quá trình phân đoạn ảnh.

Chúng ta quan sát thấy rằng kết quả của quá trình phân đoạn ảnh trên CPU và GPU giống nhau. Điều này chứng tỏ rằng việc tính toán song song trên GPU không ảnh hưởng đến chất lượng của quá trình phân đoạn ảnh, cho phép chúng ta chuyển đổi sang việc tính toán song song trên GPU mà không lo ngại về sự thay đổi về kết quả.

Khi làm việc với ảnh kích thước lớn, việc tối ưu hóa hiệu suất thông qua tính toán song song là cần thiết. Trong bài toán Ncut, việc tính toán vector riêng của ma trận trọng số W là một phần quan trọng của quá trình phân đoạn ảnh, và sử dụng GPU để tăng tốc quá trình này có thể giúp tối ưu hóa tổng thời gian thực hiện và làm cho ứng dụng Ncut trở nên hiệu quả hơn.



Hình 4. Kết quả phân đoạn ảnh với Neut trên GPU



Hình 5. Kết quả phân đoạn ảnh với Neut sử dụng gom cụm k-mean trên CPU

Lời cảm ơn

Xung đột lợi ích

Các tác giả tuyên bố không có xung đột lợi ích trong bài báo này.

TÀI LIỆU THAM KHẢO

- [1] Z. Lv, Y. Hu, and H. Zhong, "Parallel K-Means Clustering of Remote Sensing Images Based on MapReduce," in *Proc. WISM 2010: Web Information Systems and Mining, International Conference on Web Information Systems and Mining*, Springer, Berlin, Heidelberg, vol. 6318, 2010, pp. 162-170.
- [2] B. Liu, S. He, and D. He, "A Spark-Based Parallel Fuzzy C -Means Segmentation Algorithm for Agricultural Image Big Data," *IEEE access*, vol. 7, pp. 42169-42180, 2019.
- [3] J. Cao, L. Chen, and M. Wang, "Implementing a Parallel Image Edge Detection Algorithm Based on the Otsu-Canny Operator on the Hadoop Platform," *Computational Intelligence and Neuroscience*, 2018, doi: 10.1155/2018/3598284.

- [4] D. P. Augustine and P. Raj, "Performance Evaluation of Parallel Genetic Algorithm for Brain MRI Segmentation in Hadoop and Spark," *Indian Journal of Science and Technology*, vol. 9, no. 48, 2016, doi: 10.17485/ijst/2016/v9i48/140123.
- [5] M. N.t Akhtar, J. M. Saleh, and H. Awais, "Map-Reduce based tipping point scheduler for parallel image processing," *Expert Systems with Applications*, vol. 139, 2020, doi: 10.1016/j.eswa.2019.112848.
- [6] N. Wang, F. Chen, and B. Yu, "Segmentation of large-scale remotely sensed images on a Spark platform: A strategy for handling massive image tiles with the MapReduce model," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 137-147, 2020.
- [7] J. Sirotković, H. Dujmić, and V. Papić, "K-means image segmentation on massively parallel GPU architecture," in *Proc. 35th International Convention MIPRO*, Opatija, Croatia, 2012, pp. 489-494.
- [8] Q. B. Baker and K. Balhaf, "Exploiting GPUs to accelerate white blood cells segmentation in microscopic blood images," in *Proc. 8th International Conference on Information and Communication Systems (ICICS)*, Irbid, Jordan, 2017, pp. 136-140.
- [9] M. Baydoun, M. Dawi, and H. Ghaziri, "Enhanced Parallel Implementation of the K-Means Clustering Algorithm," in *Proc. 3rd International Conference on Advances in Computational Tools for Engineering Applications (ACTEA)*, IEEE, 2016, doi: 10.1109/ACTEA.2016.7560102.
- [10] M. Dalvand, A. Fathi, and A. Kamran, "Flooding region growing: a new parallel image segmentation model based on membrane computing," *Journal of Real-Time Image Processing*, vol. 18, pp. 37-55, 2021.
- [11] X. Wang, J. Pan, and S. Chu, "A Parallel Multi-Verse Optimizer for Application in Multilevel Image Segmentation," *IEEE Access*, vol. 8, pp. 32018-32030, 2020.
- [12] Y. Chen, J. Tao, and L. Liu, "Research of improving semantic image segmentation based on a feature fusion model," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 5033-5045, 2020.
- [13] L. Jiao, Y. Li, M. Gong, "Quantum-inspired immune clonal algorithm for global optimization," *IEEE Transaction on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1234-1253, 2008.
- [14] Y. Li, S. Feng, and X. Zhang, "SAR image segmentation based on quantum-inspired multiobjective evolutionary clustering algorithm," *Information Processing Letters*, vol. 114, pp. 287-293, 2014.
- [15] D. P. Hudedagaddi and B. K. Tripathy, "Quantum inspired computational intelligent techniques in image segmentation," *Quantum Inspired Computational Intelligence*, pp. 233-258, 2017.
- [16] S. Das, S. De, and S. Dey, "Magnetic Resonance Image Segmentation Using a Quantum-Inspired Modified Genetic Algorithm (QIANA) Based on FRM," John Wiley & Sons, 2020, doi: 10.1002/9781119551621.ch8.
- [17] S. Yuan, C. Wen, and B. Hang, "The dual-threshold quantum image segmentation algorithm and its simulation," *Quantum Information Processing*, vol. 19, no. 425, 2020, doi: 10.1007/s1128-020-02932-x.
- [18] J. Ghorpade, J. Parande, and M. Kulkarni, "GPGPU processing in cuda architecture," *Advanced Computing: An International Journal (ACIJ)*, vol. 3, no. 1, pp. 105-120, 2012.
- [19] J. I. Agulleiro, F. Vázquez, and E. M. Garzón, "Hybrid computing: CPU+GPU co-processing and its application to tomographic reconstruction," *Ultramicroscopy at SciVerse ScienceDirect*, vol. 115, pp. 109-114, 2012.
- [20] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions*, vol. 22, no. 8, pp. 888-905, 2000.
- [21] H. X. Lou and Y. S. Yuan, "Image segmentation based on normalized cut and CUDA parallel implementation," in *Proc. 5th IET International Conference on Wireless, Mobile and Multimedia Networks (ICWMMN 2013)*, Beijing, China, 2013, doi: 10.1049/cp.2013.2410
- [22] J. Senthilnath, S. Sindhu, and S. N. Omkar, "GPU-based normalized cuts for road extraction using satellite imagery," *Journal of Earth System Science*, vol. 123, pp. 1759-1769, 2014.
- [23] M. Naumov and T. Moon, "Parallel Spectral Graph Partitioning," NVIDIA Technical Report, 2016. [Online]. Available: <https://research.nvidia.com/sites/default/files/publications/nvr-2016-001.pdf>.
- [24] B. Catanzaro, B. Y. Su, and N. Sundaram, "Efficient, High-Quality Image Contour Detection," in *Proc. the IEEE 12th International Conference on Computer Vision*, Kyoto, Japan, 2010, doi: 10.1109/ICCV.2009.5459410.
- [25] Sattar and N. Safirin, "Parallel Algorithms for Scalable Graph Mining: Applications on Big Data and Machine Learning," University of New Orleans Theses and Dissertations, 2022. [Online]. Available: <https://scholarworks.uno.edu/td/3014/>.
- [26] W. Tao, H. Jin, and Y. Zhang, "Color Image Segmentation Based on Mean Shift and Normalized Cuts," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, pp. 1382-1389, 2007.
- [27] A. Fabjawska, "Normalized cuts and watersheds for image segmentation," in *IET Conference on Image Processing (IPR 2012)*, London, UK, 2012, doi: 10.1049/cp.2012.0440.
- [28] L. You, H. Jiang, and J. Hu, "GPU-accelerated Faster Mean Shift with euclidean distance metrics," in *Proc. the IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, Los Alamitos, CA, USA, 2022, doi: 10.1109/COMPSAC54236.2022.00037.
- [29] S. Soor and B. S. D. Sagar, "Segmentation of Multi-Band Images Using Watershed Arcs," *IEEE Signal Processing Letters*, vol. 29, pp. 2407-2411, 2022.
- [30] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, doi: 10.1109/CVPR.2005.332.
- [31] S. D. Kapade, S. M. Khairnar, and B. S. Chaudhari, "Enhanced graph based normalized cut methods for image segmentation," *ICTACT journal on image and video processing*, vol. 5, no. 2, pp. 907-911, 2014.
- [32] A. Challa, S. Danda, and B. S. D. Sagar, "Power Spectral Clustering," *Journal of Mathematical Imaging and Vision*, vol. 62, pp.1195-1213, 2020.
- [33] <https://ccia.ugr.es/cvg/dbimagenes/>



Tran Nhu Y. 2008: BSc degree in Mathematics and Computer Science, University of Science, Vietnam National University-Ho Chi Minh, Vietnam. 2013: MSc degree in Mathematics and Computer Science, University of Science, Vietnam National University-Ho Chi Minh, Vietnam. 2005 to now: Lecturer at Information Technology Faculty, University of Industry and Trade, Vietnam. Research interests: computer vision, image segmentation.
Email: ytn@hufi.edu.vn. ORCID: <https://orcid.org/0000-0002-8588-0355>



Huynh Trung Hieu received a Bachelor's and Master's degree from the Ho Chi Minh City University of Technology in 1998 and 2003, and then obtained a PhD in Computer Engineering from Chonnam National University, South Korea in 2009. Currently, he is an associate professor and works at the Ho Chi Minh City University of Industry. He is a member of IEEE and IEICE. His research focuses on artificial intelligence and its applications in medical data analysis. Email: hthieu@iecc.org. ORCID: <https://orcid.org/0000-0002-2097-0704>



Pham The Bao. 1995: B.S., 2000: MSc., 2009: Ph.D. degree in Computer Science from University of Science, Vietnam. 1995 ~ 2018: Lecturer and Professor in Department of Computer Science, Faculty of Mathematics & Computer Science, University of Science, Vietnam. 2007 ~ 2016: Vice Dean Faculty of Mathematics & Computer Science and head of Computer Science Department, University of Science, Vietnam. 2019 to now: Professor in Department of Computer Science, and Dean of Information Science Faculty, Sai Gon University, Vietnam. Chair of IC-IP Lab. Research interests: Image processing & pattern recognition, intelligent computing.

Email: ptbao@sgu.edu.vn. ORCID:  <https://orcid.org/0000-0002-4847-4366>