

Tính toán song song dựa trên GPU cho VANET: Hiện trạng và triển vọng tương lai

Abinash Borah
Khoa Khoa học Máy tính
Đại học bang Oklahoma
Stillwater, Oklahoma, Mỹ
aborah@okstate.edu

Anirudh Paranjothi
Khoa Khoa học Máy tính
Đại học bang Oklahoma
Stillwater, Oklahoma, Mỹ
anirudh.paranjothi@okstate.edu

trừu tượng—Mạng ad-hoc phương tiện (VANET) yêu cầu xử lý nhanh khối lượng dữ liệu lớn được truyền bởi các nút phương tiện để đáp ứng yêu cầu về độ trễ xử lý thấp và thông lượng cao. Để đáp ứng những yêu cầu đó thông qua việc tăng tốc độ tính toán, các kỹ thuật tính toán song song được áp dụng trong nhiều lĩnh vực khác nhau. Cách tiếp cận như vậy là sử dụng tính toán song song dựa trên Bộ xử lý đồ họa (GPU). Để sử dụng hiệu quả GPU, những thách thức của kiến trúc này cần được giải quyết. Trong bài viết này, chúng tôi xem xét các kỹ thuật dựa trên GPU tiên tiến nhất cho VANET, trình bày cách phân loại và xác định các hạn chế của chúng. Chúng tôi cũng đề xuất các hướng đi trong tương lai hướng tới việc sử dụng và tối ưu hóa hơn nữa tính toán song song dựa trên GPU cho VANET. Theo hiểu biết tốt nhất của chúng tôi, đây là đánh giá đầu tiên về tính toán song song dựa trên GPU cho VANET.

Từ khóa—VANET, ITS, tính toán song song, GPU

tôi. tôi GIỚI THIỆU

Mạng ad-hoc dành cho xe cộ (VANET) đóng vai trò là nền tảng của hệ thống giao thông thông minh (ITS). VANET là một loại mạng ad-hoc di động đặc biệt. VANET hỗ trợ giao tiếp giữa phương tiện với phương tiện (V2V) và phương tiện với cơ sở hạ tầng (V2I) để đảm bảo an toàn đường bộ và cải thiện điều kiện lái xe bằng cách cung cấp các dịch vụ khác nhau, từ giám sát tình trạng giao thông theo thời gian thực đến dịch vụ thông tin giải trí. Trong VANET, các thiết bị liên lạc được gọi là Bộ phận bên đường (RSU) được triển khai tại các điểm quan trọng được chọn dọc theo đường và các phương tiện được gắn các thiết bị được gọi là Bộ phận trên xe (OBU). Các phương tiện sử dụng OBU sẽ giao tiếp với nhau (giao tiếp V2V) và với RSU (giao tiếp V2I) [1].

Trong VANET, các phương tiện phát đi các thông báo đèn hiệu định kỳ với các thông tin như vị trí, tốc độ, danh tính, v.v. Ngoài ra, các phương tiện còn phát đi các thông báo an toàn khi xảy ra sự kiện như va chạm để cảnh báo cho các phương tiện khác. Tất cả những tin nhắn này về cơ bản dẫn đến một lượng lớn thông tin được phổ biến trên mạng. Song song đó, các kỹ thuật tính toán để giải quyết các vấn đề khác nhau trong VANET phụ thuộc rất nhiều vào các thông điệp được truyền bởi các phương tiện và do đó, lượng dữ liệu cần xử lý rất cao, đặc biệt khi mật độ của các nút trong mạng tăng lên. Hơn nữa, thông lượng cao và độ trễ xử lý thấp được mong muốn trong các ứng dụng VANET khác nhau khi xét đến tính chất quan trọng về mặt thời gian của chúng [2].

Tính toán song song nhiều lõi của Bộ xử lý đồ họa (GPU) được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau để tăng tốc tính toán. GPU là bộ đồng xử lý song song có thể hỗ trợ

tính toán song song lớn với thông lượng lệnh cao. Một số giải pháp được đề xuất trong tài liệu cho các vấn đề khác nhau trong VANET trong đó đạt được tốc độ tính toán bằng cách sử dụng GPU. Đặc điểm của kiến trúc GPU mang đến những thách thức nhất định (được thảo luận ở phần II) trong việc song song hóa dựa trên GPU. Để đạt được hiệu suất tối ưu từ GPU, việc xử lý hiệu quả những thách thức này là rất quan trọng. Ở cấp độ phần cứng, các đề xuất về RSU được trang bị GPU và sử dụng thiết bị có GPU làm OBU cũng đã được khám phá trong tài liệu. Đáng chú ý là hiệu suất của các kỹ thuật tính toán cho VANET thường được đánh giá bằng mô phỏng và một số nghiên cứu nhấn mạnh việc tăng tốc mô phỏng cho VANET sử dụng GPU. Một quan điểm khác xem xét tầm quan trọng của mô phỏng là các giải pháp tính toán song song dựa trên GPU cho VANET phải dễ dàng tích hợp vào các trình mô phỏng mạng thường được sử dụng.

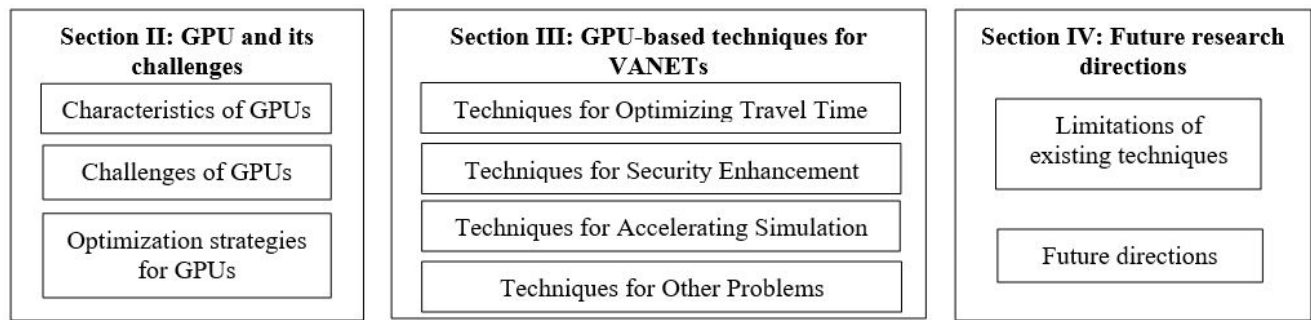
Trong bài viết này, chúng tôi xem xét các giải pháp tính toán song song dựa trên GPU tiên tiến nhất cho VANET, phân loại chúng thành bốn loại khác nhau, viz., 1) Kỹ thuật tối ưu hóa thời gian di chuyển, 2) Kỹ thuật tăng cường bảo mật, 3) Kỹ thuật để tăng tốc mô phỏng, và 4) Kỹ thuật giải các bài toán khác. Chúng tôi xác định thêm những hạn chế của những công việc hiện có này và đề xuất các hướng đi trong tương lai hướng tới việc sử dụng và tối ưu hóa hơn nữa tính toán dựa trên GPU cho VANET. Theo hiểu biết tốt nhất của chúng tôi, đây là đánh giá đầu tiên về điện toán dựa trên GPU cho VANET và chúng tôi tin rằng nó sẽ là động lực hướng tới việc khám phá sâu hơn về điện toán song song dựa trên GPU cho VANET. Hình 1 trình bày phạm vi và phân loại của bài viết này.

Phần còn lại của bài viết này được tổ chức như sau: Phần II xem xét các khái niệm chính về GPU cùng với những thách thức và chiến lược tối ưu hóa của chúng; Phần III thảo luận về các kỹ thuật dựa trên GPU cho VANET theo phân loại trên; Phần IV đưa ra những hướng nghiên cứu tiếp theo và cuối cùng là Phần V đưa ra kết luận.

II. GPU VÀ NÓ CHALLENGES

Trong phần này, chúng tôi trình bày tài liệu cơ bản cần thiết về GPU để thực hiện phần còn lại của bài viết cũng như những thách thức cần giải quyết để đạt được hiệu suất tối ưu trong tính toán song song dựa trên GPU.

GPU là bộ đồng xử lý song song cao có sẵn trong hầu hết các thiết bị, từ điện thoại thông minh đến siêu máy tính, để kết xuất đồ họa. Có thể sử dụng quy mô lớn



Hình 1. Hình trên cho thấy phạm vi và cách phân loại của bài viết này.

tính song song của GPU cho tính toán đa năng. GPU hoạt động trên mô hình Nhiều dữ liệu hướng dẫn đơn để xử lý song song khai thác tính song song ở cấp độ dữ liệu bằng cách thực hiện đồng thời một thao tác trên nhiều thành phần dữ liệu. GPU được kết nối với CPU thông qua giao diện thông lượng tương đối thấp như bus PCIe. Trong bài viết này, chúng tôi sử dụng các thuật ngữ của CUDA, một phần mở rộng của C/C++ để lập trình GPU [3]. Một khung lập trình thường được sử dụng khác cho GPU là OpenCL (Ngôn ngữ điện toán mở) [4]. Các hàm được gọi là kernel [3] được gọi từ CPU để thực thi trên GPU. Việc gọi kernel sẽ khởi chạy một mạng lưới các luồng GPU đồng thời, được nhóm thành các khối. GPU và CPU có không gian bộ nhớ khác nhau. Vì vậy cần phải sao chép dữ liệu đầu vào từ bộ nhớ CPU sang bộ nhớ GPU để xử lý trên GPU. Tương tự, kết quả tính toán cần được sao chép ngược từ bộ nhớ GPU sang bộ nhớ CPU. Bộ nhớ của GPU có tổ chức phân cấp. Mỗi luồng có các thanh ghi riêng, các luồng trong một khối hợp tác bằng cách sử dụng bộ nhớ chia sẻ trên toàn khối và tất cả các luồng trên các khối khác nhau có quyền truy cập vào bộ nhớ chung chậm hơn nhưng lớn hơn nhiều.

Để khai thác tối ưu tính song song của GPU, bắt buộc phải giải quyết các thách thức của kiến trúc này. Vì các luồng trong một khối chia sẻ bộ nhớ dùng chung nhanh hơn nên để khai thác tính năng này, cách tiếp cận phổ biến là sao chép các phần dữ liệu từ bộ nhớ chung sang bộ nhớ dùng chung theo cách thức từng giai đoạn và xử lý chúng. Nó làm giảm quyền truy cập vào bộ nhớ chung chậm hơn. Mỗi khi có quyền truy cập bộ nhớ chung, một số lượng lớn các luồng sẽ thực hiện việc đọc bộ nhớ. Để giải quyết tình trạng chậm của bộ nhớ chung, GPU có bộ đệm có thể khai thác vị trí không gian của các truy cập bộ nhớ chung để giảm lưu lượng truy cập thông qua bộ điều khiển bộ nhớ. Để tận dụng các bộ nhớ đệm như vậy, các luồng GPU cần truy cập vào bộ nhớ chung theo các mẫu tôn trọng vị trí không gian, tức là các luồng liên tiếp truy cập vào các vị trí bộ nhớ lân cận. Điều này được gọi là kết hợp bộ nhớ toàn cầu [5].

Một thách thức khác là giao diện CPU-GPU thông lượng thấp. Để giải quyết vấn đề này, cần phải tối ưu hóa lưu lượng giao tiếp thông qua giao diện CPU-GPU. Điều này có thể đạt được bằng cách sắp xếp lại dữ liệu đầu vào trong bộ nhớ CPU nếu cần và chỉ truyền dữ liệu cần thiết sang GPU. Cần phải cân bằng số lượng và kích thước của các khối để truyền dữ liệu vì việc truyền dữ liệu theo một số lượng nhỏ các khối lớn có thể không đủ để khiến GPU bận rộn, trong khi một số lượng lớn các khối nhỏ thì không tối ưu do chi phí liên quan.

Cân bằng tải là một vấn đề quan trọng khác, đề cập đến việc đảm bảo rằng tất cả các luồng của GPU thực hiện cùng một lượng công việc để thời gian thực thi của một luồng thực hiện được nhiều hơn.

công việc không làm chậm thời gian thực thi của toàn bộ hàm kernel. Truyền dữ liệu chồng chéo giữa CPU và GPU với các tính toán trên GPU là một chiến lược tối ưu hóa khác. Trong đó, một đoạn dữ liệu có thể được chuyển từ CPU sang GPU và một hạt nhân có thể được khởi chạy để hoạt động trên đó; trong khi GPU thực hiện các tính toán trên đoạn đó, một đoạn khác có thể được chuyển từ CPU. Theo cách tương tự, việc truyền dữ liệu GPU sang CPU cũng có thể bị chồng chéo với các tính toán trên GPU. Mức độ song song là một thách thức khác trong việc đạt được hiệu suất tối ưu của GPU khi việc khởi chạy các luồng phát sinh một số chi phí. Việc khởi chạy quá nhiều luồng hơn mức cần thiết có thể làm tăng chi phí hoạt động và do đó vô hiệu hóa một phần hiệu suất đạt được. Mặt khác, việc khởi chạy không đủ số lượng luồng có thể không mang lại tốc độ tăng tốc khả thi lý tưởng.

III. GPU-DỰA TRÊN KỸ THUẬT CHO VANETs

Trong phần này, chúng tôi thảo luận về các giải pháp điện toán song song dựa trên GPU cho VANET, theo phân loại được hiển thị trong Hình 1, trong các phần phụ sau: A. Kỹ thuật tối ưu hóa thời gian di chuyển, B. Kỹ thuật tăng cường bảo mật, C. Kỹ thuật tăng tốc mô phỏng và D. Kỹ thuật giải các bài toán khác. Cuối cùng, chúng tôi cung cấp bản tóm tắt các kỹ thuật được xem xét trong Bảng I.

A. Kỹ thuật tối ưu hóa thời gian di lại

Định tuyến phương tiện hiệu quả là điều quan trọng trong VANET để nâng cao trải nghiệm lái xe bằng cách giảm thời gian di chuyển. Tìm các tuyến đường ngắn nhất, không tắc nghẽn và giảm thiểu thời gian chờ đợi tại các nút giao thông là một số cách tiếp cận phổ biến để giải quyết vấn đề này. Một số công trình xem xét việc song song hóa dựa trên GPU của một số thuật toán nổi tiếp nhất định để giảm thời gian di chuyển. Tuy nhiên, tất cả các kỹ thuật được thảo luận trong tiểu mục này là sự song song hóa của một số thuật toán nổi tiếp hiện có; không có cái nào trong số chúng được thiết kế để nhằm mục tiêu cụ thể đến kiến trúc song song của GPU, vốn mong muốn đạt được đầy đủ lợi ích của việc song song hóa dựa trên GPU.

Trong số các phương pháp tiếp cận tập trung vào các tuyến đường ngắn nhất và không bị tắc nghẽn, việc triển khai song song dựa trên GPU của thuật toán Tối ưu hóa đàn kiến đã sửa đổi (MACO) [6], CUDA_MACO là một trong số đó. Thuật toán MACO giúp giảm thời gian di chuyển bằng cách tránh các tuyến đường tắc nghẽn trong quá trình định tuyến xe. Tuy nhiên, khi số lượng phương tiện trong mạng tăng lên thì thời gian xử lý của MACO cũng tăng lên dẫn đến thời gian di chuyển tăng lên. Để giải quyết vấn đề này, CUDA_MACO được đề xuất trong đó tất cả các giai đoạn của MACO được song song trên GPU.

Một thuật toán khác để giảm thời gian di chuyển bằng cách tránh các tuyến đường tắc nghẽn, GHAPO-P (Tối ưu hóa hạt kiến lai được tăng tốc GPU) được trình bày trong [8]. GHAPO-P một lần nữa là sự song song hóa dựa trên GPU của thuật toán HAPO-P [9] giúp giảm thời gian di chuyển bằng cách tránh tắc nghẽn trên đường kết hợp với kiểm soát giao thông ưu tiên. GHAPO-P giảm thời gian xử lý bằng xử lý song song so với HAPO-P.

IHAPO-G [10] là một thuật toán khác để giảm thời gian di chuyển thông qua tránh tắc nghẽn song song với thuật toán IHAPO. IHAPO kết hợp tối ưu hóa MACO và bày hạt [11] để giảm thời gian di chuyển và IHAPO-G giảm thời gian di chuyển hơn nữa bằng cách sử dụng xử lý song song trên GPU.

Thuật toán được đề xuất trong [12] đề cập đến việc đồng thời tìm đường đi ngắn nhất cho tất cả các phương tiện tới đích. Điều này đạt được thông qua việc triển khai song song thuật toán Hedetniemi để giải bài toán đường đi ngắn nhất cho tất cả các cặp. Thuật toán Hedetniemi liên tục tính lũy thừa của ma trận kề cho đến khi hai lũy thừa ma trận liên tiếp khớp nhau. Thuật toán song song tăng tốc quá trình nhân ma trận bằng GPU để giảm thời gian xử lý.

Thuật toán di truyền song song dựa trên GPU cho Bài toán nhân viên bán hàng du lịch (TSP) được đề xuất trong [13]. Thuật toán có thể được tích hợp vào mô hình điện toán đám mây xe cộ (VCC) và được sử dụng để đưa ra quyết định định tuyến trong hệ thống quản lý giao thông. Đầu tiên, các tác giả đề xuất một phương pháp song song hóa các hoạt động chính của thuật toán di truyền, sau đó áp dụng nó để giải TSP và cuối cùng đề xuất tích hợp với VCC để quản lý lưu lượng.

Một thuật toán khác dựa trên MACO là MACO-P (MACO ưu tiên), thuật toán này cũng xem xét các tín hiệu giao thông ngoài việc tìm ra tuyến đường tối ưu và do đó cải thiện thời gian di chuyển. Một phiên bản song song của thuật toán này GMACO-P (MACO ưu tiên được GPU hỗ trợ) được đề xuất trong [14] song song hóa các giai đoạn khác nhau của MACO-P trên GPU.

Thuật toán ưu tiên song song trong [15] và thuật toán điều khiển giao thông thích ứng song song trong [16] giảm độ dài hàng đợi trung bình tại các nút giao thông, do đó giảm thiểu thời gian chờ tổng thể. Các thuật toán này song song với các thuật toán nối tiếp của chúng, sử dụng mật độ phương tiện hiện tại tại một thời điểm nhất định để chọn giai đoạn tiếp theo nhằm phân bổ thời gian đèn xanh. Các phiên bản song song chạy tất cả các phần phụ của thuật toán nối tiếp trên GPU, do đó giảm thời gian tính toán để đưa ra quyết định nhanh hơn về phân bổ thời gian đèn xanh.

Khung xử lý dữ liệu Mạng xe thông minh (IVN) hỗ trợ đô thị, Hooker được đề xuất trong [17] khám phá việc sử dụng một số thuật toán đô thị để giải quyết việc lập kế hoạch đường đi và quản lý nút giao thông cho IVN. Các tối ưu hóa dựa trên đô thị khác nhau được sử dụng trong công việc này để tối ưu hóa hiệu suất GPU. Việc tối ưu hóa tạo điều kiện thuận lợi cho việc truy cập bộ nhớ toàn cầu GPU được kết hợp và chồng chéo giao tiếp CPU-GPU với tính toán trên GPU.

Ngoại trừ công việc trong [17], không có kỹ thuật nào được thảo luận trong tiểu mục này áp dụng bất kỳ chiến lược nào để tối ưu hóa hiệu suất của GPU, do đó bỏ lỡ cơ hội tăng tốc bổ sung. Hơn nữa, không ai trong số họ đề xuất mức độ

về tính song song cần xem xét, tức là nên khởi chạy bao nhiêu luồng song song trong GPU để thuật toán đạt được hiệu suất tối ưu. Điều này rất quan trọng trong bất kỳ mô hình lập trình song song nào vì việc khởi chạy nhiều luồng hơn mức cần thiết sẽ làm giảm hiệu suất do chi phí liên quan. Ngoài ra, không có kỹ thuật nào trong số này được tích hợp với các trình mô phỏng mạng thường được sử dụng; sự tích hợp như vậy có thể mang lại lợi ích cho các ứng dụng mới hơn của các kỹ thuật này trong nghiên cứu VANET.

B. Kỹ thuật nâng cao bảo mật

Bảo mật là một trong những thách thức lớn trong việc triển khai VANET. Bảo mật là trạng thái giữ cho mạng không bị đe dọa hoặc tấn công. Điều quan trọng là phải đảm bảo tính bảo mật của VANET một cách đầy đủ, vì VANET là mạng có tính di động cao và năng động, nơi cấu trúc liên kết thay đổi thường xuyên. Trong tiểu mục này, chúng tôi thảo luận về một nhóm công việc nhằm nâng cao tính bảo mật trong VANET bằng GPU. Tuy nhiên, những kỹ thuật này sử dụng tối thiểu hoặc không sử dụng các chiến lược tối ưu hóa để nâng cao hiệu suất của GPU bằng cách giải quyết các thách thức của chúng.

Trong số các giải pháp nâng cao bảo mật dựa trên GPU cho VANET, trong [18], khung tăng tốc phần cứng mật mã HAA được thiết kế cho Xác thực nhanh (RA), một sơ đồ chữ ký ngoại tuyến-trực tuyến nhận biết độ trễ cho các hệ thống chỉ huy và điều khiển. HAA sử dụng Định lý số dư Trung Quốc và thuật toán nhân Montgomery để kích hoạt tính năng song song trên GPU. Trong quá trình song song hóa dựa trên GPU được đề xuất, quyền truy cập bộ nhớ hợp nhất cho bộ nhớ chung GPU được đảm bảo để tối ưu hóa hiệu suất.

Một cách tiếp cận kết hợp sử dụng đồng thời tính năng song song hóa dựa trên GPU và dựa trên CPU cho Thuật toán chữ ký số đường cong Elliptic (ECDSA) để xác minh chữ ký phương tiện trong RSU được trình bày trong [19]. Ở đầu CPU, việc xác minh ECDSA dựa trên OpenSSL được thực hiện trong mỗi luồng và mỗi lần một chữ ký được xác minh; ở đầu GPU, nhiều chữ ký được xác minh đồng thời.

Công trình trong [20] đề xuất một hệ thống dựa trên sự tin cậy lai MobileTrust để chia sẻ tài nguyên an toàn trong VANET. Mục tiêu của nó là đánh giá hành vi của một thực thể sau khi thiết lập liên lạc an toàn bằng mật mã. Hệ thống này là một hệ thống điện toán tin cậy tập trung sử dụng công nghệ đám mây và 5G để thiết lập niềm tin. Tính năng song song hóa dựa trên GPU được áp dụng trong hệ thống này để thu thập dữ liệu song song cho các đoạn đường khác nhau.

Trong số ba kỹ thuật nêu trên được thảo luận trong tiểu mục này, chỉ HAA sử dụng một số chiến lược tối ưu hóa cho GPU bằng cách sử dụng quyền truy cập bộ nhớ toàn cầu hợp nhất. Không có kỹ thuật nào phân tích tác động của việc thay đổi mức độ song song cũng như không cung cấp bất kỳ hướng dẫn nào về điều đó. Hơn nữa, những kỹ thuật này không cung cấp bất kỳ sự tích hợp nào với các công cụ mô phỏng mạng.

C. Kỹ thuật tăng tốc mô phỏng

Đánh giá hiệu suất của mạng là điều quan trọng để điều tra mọi vấn đề có thể tồn tại trong mạng và VANET cũng không ngoại lệ. Cách phù hợp nhất cho nhiệm vụ này là triển khai các mô phỏng mang lại kết quả gần nhất với các quan sát trong thế giới thực. Nhiều công cụ mô phỏng khác nhau được sử dụng để mô phỏng và đánh giá hiệu suất của VANET. Song song

việc xử lý có thể hữu ích trong việc giảm thời gian cần thiết để thực hiện các nghiên cứu mô phỏng. Mặc dù có sự hỗ trợ tối thiểu cho việc song song hóa đa lõi trong một số trình mô phỏng thường được sử dụng nhưng không có sự hỗ trợ nào cho việc song song hóa dựa trên GPU. Do đó, một số phương pháp sử dụng song song hóa dựa trên GPU được đề xuất để tăng tốc mô phỏng VANET, sẽ được thảo luận trong tiểu mục này. Tuy nhiên, vấn đề là những cách tiếp cận này không giải quyết được những thách thức của kiến trúc GPU.

Công cụ mô phỏng VANET tăng tốc GPU sử dụng trình mô phỏng NS-3 được đề xuất trong [21] trong đó mô-đun di động của NS-3 được sửa đổi để tích hợp với GPU sử dụng CUDA. Cụ thể, tất cả tính toán khoảng cách Euclidean theo cặp giữa các nút trên đường cao tốc đều được thực hiện song song trên GPU, trong đó một luồng GPU được phân bổ để tính toán khoảng cách giữa một cặp nút.

Phương pháp mô phỏng truyền sóng vô tuyến dựa trên GPU, dựa trên mô phỏng phóng tia được giới thiệu trong [21] nhằm mô hình hóa các kênh truyền thông V2V một cách xác định. Trong phương pháp hai giai đoạn này, giai đoạn đầu tiên sử dụng phương pháp phóng tia song song để tăng tốc quá trình tính toán khả năng hiển thị khi định cấu hình cảnh. Sau đó, các đường truyền được tính toán giữa máy phát và máy thu sẽ được chuyển sang pha thứ hai, nơi các đặc tính điện từ được tính toán cho mỗi đường truyền. Trong cả hai giai đoạn, xử lý song song trên GPU được sử dụng để tăng tốc quá trình tính toán.

Khung mô phỏng mạng xe cộ, Veneris, được đề xuất trong [23] được triển khai trên công cụ trò chơi Unity bao gồm mô hình xe thực tế và một tập hợp các hành vi lái xe và thay đổi làn đường mô phỏng động lực giao thông. Một trình mô phỏng truyền tia phóng dựa trên GPU, Opal được sử dụng ở Veneris để giảm thời gian phóng tia.

Không có kỹ thuật nào được xem xét trong tiểu mục này thảo luận về mức độ song song được áp dụng để có hiệu suất tối ưu cũng như không sử dụng bất kỳ chiến lược tối ưu hóa nào cho GPU. Trong khi trình mô phỏng VANET tăng tốc GPU được đề xuất trong [21] chỉ giới hạn ở một mô-đun duy nhất, hai đề xuất còn lại tập trung vào việc tăng tốc các kịch bản mô phỏng rất cụ thể thay vì một công cụ mô phỏng tăng tốc chung hơn có thể được sử dụng trong các mô phỏng đa dạng.

D. Kỹ thuật giải các bài toán khác trong VANET

Một số kỹ thuật tính toán song song dựa trên GPU khác đang giải quyết các vấn đề riêng biệt khác nhau cho VANET. Những kỹ thuật đó sẽ được thảo luận trong tiểu mục này. Trong số các kỹ thuật này, các kỹ thuật được đề xuất trong [24] và [25] sử dụng một số dạng chiến lược tối ưu hóa cho GPU bằng cách sử dụng bộ nhớ dùng chung của GPU. Ngoài ra, công trình trong [26] đã khám phá việc sử dụng nhiều GPU để đạt được tốc độ cao hơn thông qua việc song song hóa.

Trong [24], một thuật toán heuristics tham lam được đề xuất để lên lịch các nhiệm vụ xe cộ trong các đám mây xe cộ khác nhau tạo thành một hệ thống VCC, với giao tiếp V2I có sẵn theo cơ hội. Sơ đồ sắp xếp nhiệm vụ phương tiện được xây dựng dưới dạng Quy trình quyết định Markov (MDP) trong công việc này, trong đó thuật toán giá trị lặp được sử dụng để tính toán các quyết định tối ưu của MDP được tăng tốc trên GPU bằng thuật toán song song, Khối lập phân chia trạng thái (BSDI) .

Việc song song hóa phân loại gói dựa trên GPU cho Internet phương tiện được xác định bằng phần mềm (IoV) dựa trên SDN (Mạng được xác định bằng phần mềm) được đề xuất trong [25]. Các thuật toán Tìm kiếm không gian bộ dữ liệu (TSS) và Tìm kiếm không gian bộ dữ liệu cắt tỉa (TPS) để phân loại gói được song song hóa trong công việc này khi xem xét cấu trúc có khả năng song song hóa cao của chúng.

Công việc trong [26] trình bày một bộ định vị hợp tác phổ quát ba chiều (3D UCL) cho VANET trong không gian 3D dưới nhiều loại phép đo khác nhau. Phiên bản song song của 3D UCL sử dụng GPU được triển khai để giảm thời gian xử lý của phiên bản nối tiếp.

Một mô hình tối ưu hóa siêu dữ liệu, dựa trên tối ưu hóa nhóm hạt, giải quyết vấn đề chất lượng dịch vụ thấp do chuyển động của phương tiện và phạm vi bao phủ biên hạn chế trong cài đặt điện toán biên phương tiện được thảo luận trong [27]. Thuật toán GPUPSO được đề xuất sử dụng tính song song dựa trên GPU để tăng tốc các tính toán nhất định của mô hình tối ưu hóa.

Không có kỹ thuật nào trong bốn kỹ thuật trên thảo luận về mức độ song song được áp dụng để đạt hiệu suất tối ưu. Mặc dù các kỹ thuật trong [24] và [25] đã sử dụng bộ nhớ dùng chung của GPU để tối ưu hóa hiệu suất, nhưng các kỹ thuật trong [26] và [27] không sử dụng bất kỳ chiến lược tối ưu hóa nào cho GPU. Hơn nữa, không có kỹ thuật nào trong số này được tích hợp với các công cụ mô phỏng mạng.

Để kết thúc phần này, trong Bảng I bên dưới, chúng tôi trình bày tóm tắt các kỹ thuật được thảo luận trong bài viết này về vấn đề mà chúng giải quyết, các chiến lược tối ưu hóa được áp dụng cho GPU và các công cụ triển khai được sử dụng (điều này cũng chỉ rõ liệu có bất kỳ sự tích hợp nào được thực hiện với một mô phỏng mạng).

BẢNG I. TỔNG HỢP CỦA GPU-DỰA TRÊN PARALLEL COMPUTING KỸ THUẬT CHO VANETS

phần loại cation	Đề xuất	Vấn đề Địa chỉ	Tối ưu hóa bật cho GPU	Thực hiện Công cụ ation
Tối ưu hóa thời gian di chuyển	CUDA_MA CO2 [7]	Thời gian du lịch giảm bằng cách tránh tắc nghẽn <small>tuyến đường</small>	-	CUDA bộ công cụ 7.0, Bật C/C++ Thị giá Phòng thu 2010
	GHAPO-P <small>[số 8]</small>	Thời gian du lịch giảm bằng cách tránh trên đường sự tắc nghẽn kết hợp với giao thông ưu tiên điều khiển	-	CUDA bộ công cụ 7.5, C trên trực quan Phòng thu 2013
	IHAPO-G [10]	Thời gian du lịch giảm bằng cách tránh tắc nghẽn <small>tuyến đường</small>	-	CUDA bộ công cụ 7.5, C trên trực quan Phòng thu 2013
	dựa trên GPU của Hedetniemi thuật toán [12]	Thời gian du lịch giảm bớt bằng cách tìm tất cả các cặp ngắn nhất <small>con đường</small>	-	OpenCL
	Song song di truyền thuật toán cho TSP [13]	Làm xe quyết định định tuyến cho giao thông sự quản lý	-	CUDA bộ công cụ 8.0 (V8.0.61), C++
	GMACO-P [14]	Thời gian du lịch giảm bằng cách	-	CUDA bộ công cụ 7.5,

phân loại cation	Đề xuất	Vấn đề Địa chỉ	Tối ưu hóa bật cho GPU	Thực hiện Công cụ ation
		tránh tắc nghẽn tuyến đường và giảm chờ tín hiệu giao thông <small>thời gian</small>		C trên trực quan Phòng thu 2013
	Song song phủ đầu giao thông điều khiển thuật toán [15]	Thời gian chờ đợi giảm thiểu bằng cách giảm hàng đợi chiều dài tại Giao lộ	-	CUDA bộ công cụ 7.5, C trên trực quan Phòng thu 2013
	Song song thích nghi giao thông điều khiển thuật toán [16]	Thời gian chờ đợi giảm thiểu bằng cách giảm hàng đợi chiều dài tại Giao lộ	-	CUDA bộ công cụ 7.5, C trên trực quan Phòng thu 2013
	Gái điểm [17]	Đường đi của xe lập kế hoạch, ngã tư sự quản lý	kết hợp lại ký ức truy cập, CPU-GPU giao tiếp hóa, và tính toán <small>chồng lên nhau</small>	CUDA 8.0
Bảo vệ Sự năng cao	HA [18]	Tăng tốc nhanh chóng xác thực	kết hợp lại ký ức truy cập	CUDA
	Song song ECDSA [19]	Chữ ký xe xác minh trong RSU	-	OpenCL
	MobileTrust [20]	Dựa trên sự tin cậy chia sẻ tài nguyên	-	SUMO, C++, CUDA
Tăng tốc mô phỏng	tăng tốc NS-3 [21]	tăng tốc mô phỏng VANET trong trình mô phỏng NS-3	-	NS-3, CUDA
	dựa trên GPU cá đuôi- phóng mô phỏng [22]	Mô phỏng V2V giao tiếp kênh	-	CUDA 6.5, OptiX 3.7
	Veneris [23]	Mô phỏng của hợp tác xã lái xe tự động	-	Trò chơi đoàn kết động cơ, Đà mắt mèo, OMNET++
Các kỹ thuật khác	BSDI [24]	Nhiệm vụ xe cộ vị trí trong phân phối đám mây xe cộ	Sử dụng GPU đã chia sẻ ký ức	CUDA
	TSS song song và TPS [25]	Phân loại gói cho IoV dựa trên SDN	Sử dụng GPU đã chia sẻ ký ức	CUDA 9, C
	3D song song UCL [26]	hợp tác xã định vị để thu hồi xe chức vụ	-	CUDA Bộ công cụ v.8.0
	GPUPSO [27]	Chất lượng dịch vụ thấp ở rìa xe cộ tín học	-	CUDA

IV. DKIỆN NGHỊ CHO FUTURE NGHIÊN CỨU

Mặc dù có rất nhiều công trình sử dụng tính toán song song dựa trên GPU trong các lĩnh vực khác nhau, như chúng ta tìm thấy trong phần III, chỉ có một số lượng hạn chế các công trình như vậy tồn tại trong tài liệu về VANET. Ngoài ra, những công việc này còn có những hạn chế khác như chưa áp dụng được các chiến lược để đạt được hiệu quả tối ưu.

hiệu suất của GPU, thiếu tích hợp GPU với trình mô phỏng mạng, v.v. Chúng tôi tin rằng việc giải quyết hiệu quả những vấn đề này sẽ giúp cộng đồng nghiên cứu nhận ra tiềm năng thực sự của sức mạnh tính toán song song của GPU trong nghiên cứu VANET. Điều đáng chú ý là ở cấp độ phần cứng, một số đề xuất [28-30] xem xét việc sử dụng các thiết bị như điện thoại thông minh làm OBU. Vì điện thoại thông minh đã được trang bị GPU nên khả năng tính toán của GPU có thể được sử dụng để tính toán song song tại OBU. Một số công trình khác [31-33] khám phá việc sử dụng RSU dựa trên GPU. Dựa trên những quan sát của chúng tôi trong bài đánh giá này, tiếp theo chúng tôi đề xuất một số hướng nghiên cứu trong tương lai nhằm đáp ứng triển vọng sử dụng GPU cho VANET.

A. Chiến lược tối ưu hóa cho GPU

Như được thấy trong Bảng I, các chiến lược tối ưu hóa để đạt được tiềm năng tối đa của việc song song hóa dựa trên GPU được sử dụng ít trong các phương pháp hiện có cho VANET, mặc dù những tối ưu hóa như vậy có thể mang lại lợi ích hiệu suất đáng kể. Các đánh giá thực nghiệm trong [25] là sự xác nhận điều này. Việc áp dụng hiệu quả các chiến lược tối ưu hóa này để giải quyết các đặc điểm độc đáo của kiến trúc này sẽ cho phép hiện thực hóa các khả năng thực sự của điện toán song song dựa trên GPU trong nghiên cứu VANET.

B. Mức độ song song thích hợp

Các phương pháp dựa trên GPU hiện có cho VANET không xác định mức độ song song được áp dụng cũng như không thử nghiệm số lượng luồng GPU khác nhau trong các đánh giá thử nghiệm cho các phương pháp được đề xuất. Những đánh giá phù hợp về kỹ thuật song song với các mức độ song song khác nhau sẽ cho phép xác định mức độ song song chính xác cho các quy mô vấn đề khác nhau. Do đó, kỹ thuật dựa trên GPU sẽ phù hợp hơn với các ứng dụng trong đời thực dựa trên lượng dữ liệu mà ứng dụng cần xử lý.

C. Tích hợp GPU với Trình mô phỏng mạng

Có rất ít nỗ lực hướng tới việc tích hợp GPU với các công cụ mô phỏng mạng thường được sử dụng. Chúng tôi tin rằng sự tích hợp như vậy sẽ thúc đẩy các nhà nghiên cứu khám phá thêm các giải pháp dựa trên GPU cho VANET. Hơn nữa, nó có thể khuyến khích khám phá khả năng song song hóa dựa trên GPU cho nhiều vấn đề khác trong VANET bằng cách cung cấp hỗ trợ cho các đánh giá dựa trên mô phỏng. Ví dụ, không có nỗ lực song song hóa dựa trên GPU đối với một số vấn đề được nghiên cứu rộng rãi trong VANET, chẳng hạn như định tuyến gói, phân cụm, v.v.

D. Giải pháp học máy dựa trên GPU cho VANET

Nhiều giải pháp cho các vấn đề khác nhau trong VANET sử dụng kỹ thuật học máy (ML). GPU đã được cộng đồng ML sử dụng rộng rãi và các thư viện ML khác nhau đã được phát triển nhằm vào kiến trúc GPU. Các giải pháp dựa trên kỹ thuật ML cho VANET có thể sử dụng các thư viện ML dựa trên GPU hiện có để phát triển các giải pháp dựa trên GPU cho VANET. Ngoài ra, các giải pháp mới cho nhiều vấn đề khác nhau trong VANET có thể được thiết kế dựa trên các kỹ thuật ML dựa trên GPU có sẵn.

E. Khám phá các giải pháp đa GPU

Một lĩnh vực tiềm năng khác cần nghiên cứu sâu hơn là việc sử dụng nhiều GPU cùng nhau để cải thiện hiệu suất bổ sung thông qua quá trình song song hóa. Như được thể hiện bởi

đánh giá thực nghiệm trong [26], việc sử dụng hai card GPU mang lại hiệu suất tốt hơn cho kỹ thuật đề xuất của họ. Tuy nhiên, khi làm như vậy, sự cân bằng giữa lượng dữ liệu được xử lý trong GPU và số lượng GPU được sử dụng vẫn cần được duy trì.

V. CTUYÊN BỐ

Để kết luận, trong bài viết này, chúng tôi đã xem xét các kỹ thuật tính toán song song dựa trên GPU tiên tiến nhất nhằm giải quyết các vấn đề khác nhau trong VANET bằng cách phân loại chúng. Chúng tôi đã xác định thêm những hạn chế trong các phương pháp này và dựa trên đó đề xuất một số hướng đi trong tương lai để khám phá và khai thác tính toán song song dựa trên GPU cho VANET, bao gồm khả năng giải quyết các vấn đề chưa được xử lý trước đây với GPU.

RIỆU QUẢ

- [1] M. Bayat, M. Pournaghi, M. Rahimi và M. Barmshoory, "NERA: sơ đồ xác thực dựa trên RSU mới và hiệu quả cho VANET," *Wireless Networks*, 26, 2020, trang 3083-3098.
- [2] H. Shahwani, SA Shah, M. Ashraf, M. Akram, JP Jeong và J. Shin, "Một cuộc khảo sát toàn diện về việc phổ biến dữ liệu trong Mạng Ad Hoc dành cho xe cộ," *Truyền thông xe cộ* 34, 2022, 100420.
- [3] "Hướng dẫn lập trình CUDA C++" Tập đoàn NVIDIA, 2023. [Trực tuyến]. Có sẵn tại: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/> . [Truy cập ngày 5 tháng 7 năm 2023]
- [4] <https://www.khronos.org/api/opencl> [Truy cập ngày 5 tháng 7 năm 2023]
- [5] "Hướng dẫn thực hành tốt nhất CUDA C++," Tập đoàn NVIDIA, 2023. [Trực tuyến]. Có sẵn tại: <https://docs.nvidia.com/cuda/cuda-c-bestpractices-guide/> . [Truy cập ngày 5 tháng 7 năm 2023]
- [6] V. Jindal, H. Dhankani, R. Garg và P. Bedi, "MACO: ACO đã sửa đổi để giảm thời gian di chuyển trong VANET," trong *Kỷ yếu của Hội nghị chuyên đề quốc tế lần thứ ba về Phụ nữ trong lĩnh vực Máy tính và Tin học*, 2015, trang 0,97-102.
- [7] V. Jindal và P. Bedi, "Giảm thời gian di chuyển trong VANET bằng cách triển khai song song MACO (ACO đã sửa đổi)," trong *Những đổi mới trong ứng dụng và điện toán lấy cảm hứng từ sinh học: Kỷ yếu của Hội nghị quốc tế về đổi mới trong ứng dụng và điện toán lấy cảm hứng từ sinh học (IBICA 2015)* được tổ chức tại Kochi, Ấn Độ trong thời gian từ ngày 16 đến 18 tháng 12 năm 2015, trang 383-392. Nhà xuất bản Quốc tế Springer, 2016.
- [8] V. Jindal và P. Bedi, "GPU tăng tốc HAPO ưu tiên (GHAPO-P) để giảm thời gian chuyển đi trong VANETS," trong *Hội nghị quốc tế về những tiến bộ trong máy tính, truyền thông và tin học (ICACCI)*, trang 1569-1575 . IEEE, 2017.
- [9] V. Jindal, VR Sharma và P. Bedi, "Thuật toán tối ưu hóa hạt kiến lai ưu tiên (HAPO-P) cho giao thông thông minh." Trong *Hội nghị quốc tế về những tiến bộ trong máy tính, truyền thông và tin học (ICACCI)*, trang 1357-1363. IEEE, 2016.
- [10] V. Jindal và P. Bedi, "Thuật toán HAPO cải tiến sử dụng khai thác GPU (IHAPO-G) để phản hồi nhanh trong VANET." trong *Hội nghị quốc tế về những đổi mới gần đây trong kỹ thuật điện, điện tử và truyền thông (ICRIECE)*, trang 1150-1155. IEEE, 2018.
- [11] J. Kennedy và RC Eberhart, "Tối ưu hóa đàn hạt," trong *Hội nghị quốc tế về mạng thần kinh*, IEEE, Piscataway, NJ, 1995, trang 1942-1948.
- [12] SAM Ali và EH Al-Hemiyari, "Phương pháp triển khai song song để giải quyết vấn đề đường đi ngắn nhất cho mạng xe cộ," trong *Hội nghị quốc tế về công nghệ thông tin nhằm tăng cường hợp tác trực tuyến và ứng dụng khác (IT-ELA)*, trang 121-126. IEEE, 2020.
- [13] M. Abbasi, M. Rafiee, MR Khosravi, A. Jolfaei, VG Menon, và J. M. Koushyar, "Giải pháp thuật toán di truyền song song hiệu quả cho vấn đề định tuyến phương tiện trong việc triển khai đám mây của hệ thống giao thông thông minh," *Tạp chí Điện toán Đám mây* số 9, 2020, trang 1-14.
- [14] V. Jindal và P. Bedi, "GMACO-P: Thuật toán MACO ưu tiên được GPU hỗ trợ để kích hoạt Giao thông thông minh," bản in trước arXiv arXiv:2010.14244, 2020.

- [15] V. Jindal và P. Bedi, "Giảm thời gian chờ đợi bằng thuật toán ưu tiên song song trong VANET," *Vehicular Communications* 7, 2017, trang 58-65.
- [16] V. Jindal và P. Bedi, "Kiểm soát giao thông thích ứng hiệu suất cao để đáp ứng hiệu quả trong các mạng đặc biệt dành cho xe cộ," *Tạp chí Quốc tế về Khoa học và Kỹ thuật Tính toán* 16, không. Ngày 4 tháng 4 năm 2018, trang: 390-400.
- [17] Z. Zheng và AK Bashir, "Xử lý dữ liệu mạng phương tiện thông minh hỗ trợ đô thị," *Giao dịch IEEE trên Hệ thống Giao thông Thông minh* 23, không. Ngày 5 tháng 1 năm 2022, trang 4726-4735.
- [18] A. Singla, A. Mudgerikar, I. Papapanagiotou và AA Yavuz, "Haa: Xác thực tăng tốc phần cứng cho internet vạn vật trong các mạng xe cộ quan trọng trong nhiệm vụ," trong *Hội nghị Truyền thông Quân sự IEEE*, trang 1298-1304. IEEE, 2015.
- [19] S. Lee, H. Seo, H. Kwon và H. Yoon, "Phương pháp kết hợp triển khai song song trên CPU-GPU để xác minh ECDSA tốc độ cao," *Tạp chí Siêu máy tính* 75, 2019, trang 4329- 4349.
- [20] G. Hatzivasilis, O. Soultatos, S. Ioannidis, G. Spanoudakis, V. Katos, và G. Demetriou, "MobileTrust: Tích hợp kiến trúc an toàn trong VANET," *Giao dịch ACM trên Hệ thống Vật lý Điện tử* 4, không. Ngày 3 năm 2020, trang 1-25.
- [21] S. Yip, M. Chok và A. Asaduzzaman, "Một trình mô phỏng mạng khu vực xe cộ được tăng tốc CUDA đầy hứa hẹn bằng cách sử dụng NS-3," trong *IEEE 33th Hội nghị Truyền thông và Máy tính Hiệu suất Quốc tế*, 2014, trang 1-2.
- [22] M. Schiller, A. Knoll, M. Mockler và T. Eibert, "Việc khởi chạy tia tăng tốc GPU cho các ổ đĩa thử nghiệm ảo có độ chính xác cao của các ứng dụng VANET," tại *Hội nghị quốc tế năm 2015 về Máy tính hiệu năng cao & Mô phỏng (HPCS)*), trang 262-268. IEEE, 2015.
- [23] E. Egea-Lopez, F. Losilla, J. Pascual-Garcia và JM Molina-Garcia-Pardo, "Mô phỏng mạng xe cộ với vật lý thực tế," *IEEE Access* 7, 2019, trang 44021-44036.
- [24] Y. Maalej, A. Abderrahim, M. Guizani và B. Hamdaoui, "CUDA tăng tốc lập lịch tác vụ trong các đám mây xe cộ với V2I có sẵn theo cơ hội," trong *Hội nghị Truyền thông Quốc tế của IEEE (ICC)*, 2017, trang 1-6 .
- [25] M. Abbasi, A. Najafi, M. Rafiee, MR Khosravi, VG Menon và G. Muhammad, "Xử lý luồng hiệu quả trong Internet phương tiện dựa trên SDN được hình dung bằng 5G sử dụng GPU," *Giao dịch IEEE trên Hệ thống Giao thông Thông minh* 22, không. Ngày 8 tháng 8 năm 2020, trang 5283-5292.
- [26] S. Wang và X. Jiang, "Định vị hợp tác ba chiều trong các mạng đặc biệt dành cho xe cộ," *Giao dịch IEEE trên Hệ thống Giao thông Thông minh* 22, không. Ngày 2 năm 2020, trang 937-950.
- [27] MA Alqarni, MH Mousa và MK Hussein, "Giảm tải tác vụ bằng cách sử dụng tối ưu hóa nhóm hạt dựa trên GPU để tính toán cạnh phương tiện hiệu suất cao," *Tạp chí Khoa học Thông tin và Máy tính của Đại học King Saud* 34, không. Ngày 10 tháng 10 năm 2022, trang 10356-10364.
- [28] MA Abdel-Aty, Q. Cai, S. Agarwal, Z. Islam, P. Li, S. Zhang, D. Hasan và J. Huang, "Sử dụng điện thoại thông minh làm Trình mô phỏng thiết bị trên máy bay (OBU) Nghiên cứu," năm 2020.
- [29] X. Li, Y. Feng, F. Wang và Q. Qian, "Khi điện thoại thông minh kết hợp với phương tiện giao thông: Một sơ đồ thiết bị tích hợp mới cho VANET," tại *Hội nghị Quốc tế về Máy tính và Công nghệ Thông tin của IEEE; Máy tính và Truyền thông Phổ biến; Máy tính đáng tin cậy, tự động và an toàn; Trí tuệ lan tỏa và Máy tính*, 2015, trang 1095-1100.
- [30] K. Su, H. Wu, W. Chang và Y. Chou, "Hệ thống liên lạc giữa xe với xe thông qua mạng wi-fi sử dụng điện thoại thông minh Android," trong *Hội nghị quốc tế về phương tiện và triển lãm được kết nối (ICCVE)*, 2012, trang 191-196.
- [31] GAG Castillo, E. Bonetto, D. Brevi, F. Scappatura, A. Sheikh và R. Scopigno, "Đánh giá độ trễ của nguyên mẫu ứng dụng an toàn ITS để bảo vệ người đi bộ qua đường," năm 91st Hội nghị Công nghệ Xe cộ, 2020, trang 1-5.
- [32] R. Ojala, J. Vepsäläinen, J. Hanhiova, V. Hirvisalo và K. Tammi, "Đơn vị bên đường dựa trên mạng thần kinh tích chập mới để định vị chính xác người đi bộ," *Giao dịch IEEE trên Hệ thống Giao thông Thông minh* 21, không. Ngày 9 năm 2019, trang 3756-3765.
- [33] D. Weimer, S. Köhler, C. Hellert, K. Doll, U. Brunsmann và R. Krzikalla, "Kiến trúc GPU để phát hiện người đi bộ đa cảm biến cố định tại các nút giao thông thông minh," trong *Hội nghị chuyên đề về phương tiện thông minh của IEEE*, 2011, trang . 89-94.