Visual SLAM - an System for RGB-D Cameras

Nguyễn Bá Công
Mã Số Sinh Viên: 22127046
nbcong22@clc.fitus.edu.vn

Nguyễn Huỳnh Hải Đăng
Mã Số Sinh Viên: 22127052
nhhdang22@clc.fitus.edu.vn

Đặng Trần Anh Khoa
Mã Số Sinh Viên: 22127024
dtakhoa22@clc.fitus.edu.vn

Abstract

NICE-SLAM (Neural Implicit Scalable Encoding for SLAM) giới thiệu một phương pháp mới cho việc xây dựng bản đồ 3D đồng thời và định vị (SLAM) thời gian thực bằng cách sử dụng đại diện ẩn thần kinh. Khác với các phương pháp SLAM truyền thống dựa vào việc so khớp đặc trưng điểm rời rạc, NICE-SLAM sử dụng mã hóa cảnh dựa trên lưới phân cấp, cho phép tối ưu hóa cục bộ các đặc trưng cảnh. Phương pháp dựa trên học sâu này mang lại nhiều lợi thế, bao gồm khả năng mở rộng cao, hiệu suất thời gian thực và tính bền vững với nhiễu và môi trường động. Bằng cách kết hợp bộ giải mã ẩn liên tục được huấn luyện trước và lưới đặc trưng đa cấp, NICE-SLAM nâng cao cả độ chính xác của bản đồ và độ chính xác trong việc theo dõi, ngay cả trong các cảnh lớn và phức tạp. Việc sử dụng bộ kết xuất khả di giúp tối ưu hóa chi tiết hình học cảnh và vị trí camera bằng cách tối thiểu hóa mất mát khi tái tạo độ sâu và màu sắc, từ đó duy trì độ trung thực cao trong các bản đồ 3D. Kết quả thử nghiệm cho thấy NICE-SLAM vượt trội hơn các hệ thống SLAM truyền thống, đạt kết quả tốt hơn trong môi trường trong nhà quy mô lớn và duy trì tính bền vững trong môi trường có đối tượng động. Điều này làm cho NICE-SLAM trở thành một giải pháp hứa hẹn cho các ứng dụng điều hướng tự động, khám phá robot và thực tế tăng cường, nơi mà việc xây dựng bản đồ thời gian thực, quy mô lớn là rất quan trọng.

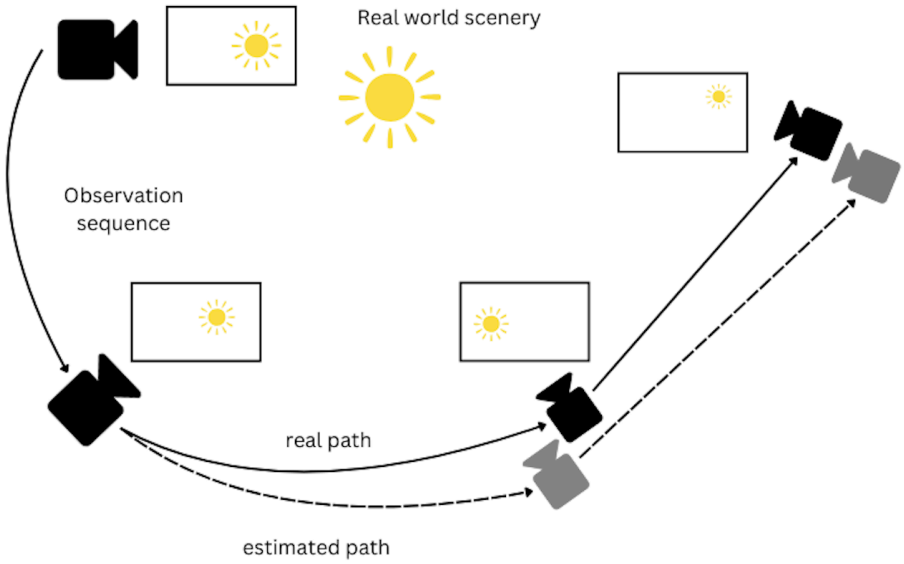
Mục lục

1 Introduction

Trong một không gian chưa biết trước, việc để một robot di chuyển tự do xung quanh cần xây dựng một bản đề khu vực đấy đồng thời xác định vị trí chính nó trong bản đồ đấy. Để giải quyết quá trình này, một phương pháp tên là Simultaneous Localization And Mapping - SLAM (Định vị và Lập bản đồ Đồng thời).

SLAM là một bài toán được sự quan tâm to lớn của cộng đồng thị giác máy tính, hơn nữa nó còn đang được ứng dụng mở rộng trong các lĩnh vực khác như công nghệ theo dõi cử động tay và nhận diện cử chỉ tay, công nghệ tích hợp LiDAR.

Về lý thuyết, bài toán SLAM đã có phương án giải quyết. Tuy nhiên, thực tế cho thấy còn nhiều vấn đề phát sinh. Các khó khăn đến từ việc xử lý thông tin thời gian thực mà một hệ thống điều hướng cần đáp ứng. Với môi trường ngoài trời, bài toán sẽ được hệ thống GPS giải quyết và cung cấp vị trí chính xác để robot làm việc. Ngược lại, nếu ở trong nhà nơi GPS không có sẵn và có vẻ phức tạp kèm độ tin cậy kém, việc nhận định vị trí chính xác trở nên khó khăn hơn và cần giải pháp thay thế.



Hình 1: Quá trình xác định vị trí camera qua SLAM.

Các thiết bị cảm biến thường được dùng trong hệ thống SLAM là:

- **Máy quét La-de**: Thiết bị cho độ chính xác cao trong việc thu thập thông tin độ sâu. Nhưng giá thành cao và nặng nên không thuận tiện trong việc di chuyển thiết bị.
- **Camera độ sâu**: Camera với cảm biến hỗ trợ tốt cho nghiên cứu áp dụng SLAM, rẻ hơn máy La-de. Camera đem lại nhiều thông tin hình ảnh hơn và không phải

tính toán lại thông tin độ sâu như camera thông thường. Điểm mạnh lớn nhất là Camera độ sâu đem lại ứng dụng thực tiễn cao.

1.1 Động lực nghiên cứu

Các hệ thống SLAM truyền thống chủ yếu dựa vào việc **trích xuất đặc trưng hình ảnh** và **khớp điểm** để ước tính vị trí của camera, sử dụng các thuật toán như **RANSAC** để xử lý các sai số trong quá trình so khớp. Mặc dù hiệu quả trong các môi trường nhỏ và có độ phức tạp thấp, các phương pháp này có một số hạn chế lớn khi phải xử lý các cảnh phức tạp và quy mô lớn. Việc khớp các đặc trưng rời rạc trong các môi trường có nhiều đối tượng động hay nhiễu có thể dẫn đến sai số lớn trong bản đồ và gây khó khăn trong việc duy trì độ chính xác của vị trí camera theo thời gian. Hơn nữa, việc sử dụng các mô hình đại diện cảnh vật rời rạc không đủ linh hoạt để xử lý các thay đổi trong cảnh vật và khó mở rộng cho các môi trường lớn.

Chính những hạn chế này của SLAM truyền thống đã thúc đẩy sự ra đời của **NICE-SLAM** (Neural Implicit Scalable Encoding for SLAM). NICE-SLAM sử dụng **đại diện ẩn thần kinh** để mô hình hóa hình học và màu sắc của cảnh vật, thay vì sử dụng các đặc trưng hình học rời rạc như trong các phương pháp truyền thống. Phương pháp này sử dụng **lưới phân cấp** để mã hóa thông tin cảnh vật, giúp hệ thống có thể tối ưu hóa cục bộ và thích ứng linh hoạt với các cảnh phức tạp mà không gặp phải vấn đề về mở rộng quy mô hoặc thiếu chính xác. NICE-SLAM giải quyết được các vấn đề mà SLAM truyền thống gặp phải, đặc biệt là trong việc **xử lý môi trường động**, **tái tạo các cảnh lớn**, và duy trì hiệu suất theo **thời gian thực**.

Với việc thay thế các phương pháp cũ bằng các đại diện cảnh liên tục và khả năng tối ưu hóa thông qua các bộ giải mã ẩn thần kinh, NICE-SLAM không chỉ cải thiện độ chính xác mà còn làm tăng khả năng **mở rộng** và **xử lý môi trường động**. Điều này giúp hệ thống hoạt động tốt hơn trong các tình huống thực tế phức tạp, nơi SLAM truyền thống không còn hiệu quả, mở ra khả năng ứng dụng mạnh mẽ hơn trong các lĩnh vực như robot tự hành, xe tự lái và thực tế ảo, nơi các môi trường thường xuyên thay đổi và yêu cầu khả năng tái tạo cảnh vật chính xác trong thời gian thực.

1.2 Outline

2 Các công trình nghiên cứu liên quan

Trong những năm gần đây, nhiều nghiên cứu đã được thực hiện để giải quyết các vấn đề liên quan đến việc xây dựng bản đồ 3D và định vị đồng thời (SLAM) trong các môi trường phức tạp, đặc biệt là khi sử dụng dữ liệu từ các cảm biến RGB-D. Một số phương pháp truyền thống đã đạt được những thành công nhất định, tuy nhiên, chúng vẫn gặp phải nhiều hạn chế khi xử lý các cảnh lớn và động.

Một trong những nghiên cứu đáng chú ý là KinectFusion của Newcombe et al. (2011), nghiên cứu này giới thiệu một phương pháp xây dựng bản đồ dày đặc theo thời gian thực với sự hỗ trợ của camera Kinect, giúp tái tạo các bề mặt 3D chi tiết từ các ảnh RGB và độ sâu . Hệ thống này đã mở ra hướng đi mới trong việc tái tạo không gian 3D với độ chính xác cao, nhưng vẫn gặp phải khó khăn trong việc duy trì hiệu suất khi môi trường có sự thay đổi hoặc thiếu hụt dữ liệu.

Các nghiên cứu gần đây hơn, chẳng hạn như DTAM của Newcombe et al. (2011) và ElasticFusion của Whelan et al. (2015) , đã tiếp tục phát triển các phương pháp SLAM dày đặc mà không cần đồ thị vị trí, mang lại kết quả tốt hơn trong môi trường không có điểm đánh dấu rõ ràng. Tuy nhiên, các phương pháp này vẫn gặp vấn đề khi mở rộng quy mô và xử lý môi trường phức tạp.

Với sự phát triển của học sâu và các mô hình đại diện ẩn, các phương pháp mới như NICE-SLAM và Nerf (Mildenhall et al., 2020) đã mở ra một kỷ nguyên mới trong việc mô hình

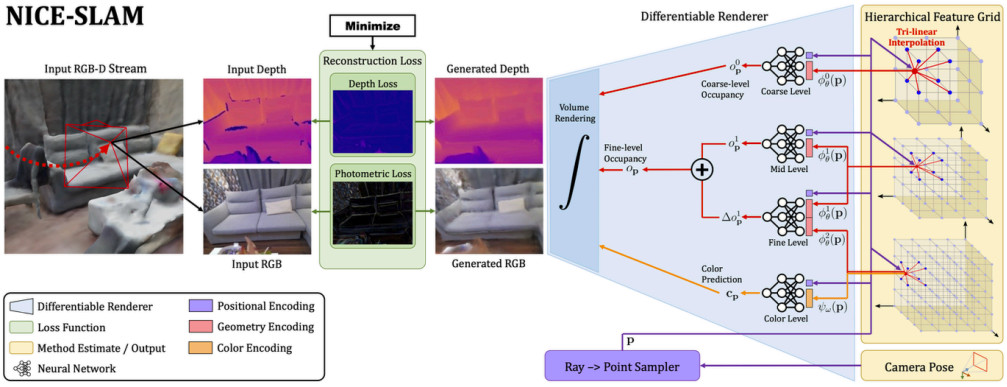
hóa và tái tạo hình học 3D. NICE-SLAM sử dụng đại diện ẩn thần kinh để xây dựng bản đồ 3D một cách chính xác hơn, vượt qua được nhiều hạn chế của các phương pháp SLAM truyền thống. Đồng thời, việc sử dụng các mạng thần kinh sâu giúp hệ thống có thể thích ứng với các thay đổi trong cảnh vật và tăng tính ổn định khi xử lý dữ liệu nhiễu hoặc thiếu hụt.

Bên cạnh đó, các nghiên cứu về Nerf (Neural Radiance Fields) như Mildenhall et al. (2020) và Martin-Brualla et al. (2021) đã đưa ra các phương pháp sử dụng các trường độ sáng thần kinh để tái tạo cảnh vật một cách chi tiết từ các bộ ảnh chụp khác nhau, cho phép tổng hợp hình ảnh và hình học 3D ở độ phân giải cao .

Những tiến bộ này trong việc sử dụng các mô hình đại diện ẩn và học sâu cho thấy tiềm năng lớn của các phương pháp SLAM mới trong việc cải thiện hiệu quả và độ chính xác của việc tái tạo 3D trong các môi trường phức tạp và động.

3 Phương pháp

Chúng tôi cung cấp tổng quan về phương pháp của chúng tôi trong Hình 2. Chúng tôi biểu diễn hình học và ngoại hình của cảnh bằng cách sử dụng bốn lưới đặc trưng và các bộ giải mã tương ứng của chúng (Mục 3.1). Chúng tôi truy vết các tia quan sát cho mỗi điểm ảnh bằng cách sử dụng hiệu chuẩn máy ảnh đã ước tính. Bằng cách lấy mẫu các điểm dọc theo tia quan sát và truy vấn mạng, chúng tôi có thể hiển thị cả giá trị độ sâu và màu sắc của tia này (Mục 3.2). Bằng cách giảm thiểu các tổn thất tái hiển thị cho độ sâu và màu sắc, chúng tôi có thể tối ưu hóa cả vị trí máy ảnh và hình học cảnh theo cách luân phiên (Mục 3.3) cho các khung hình chính được chọn (Mục 3.4).



Hình 2: Tổng quan hệ thống. Phương pháp của chúng tôi nhận một luồng ảnh RGB-D làm đầu vào và xuất ra cả vị trí máy ảnh cùng với một biểu diễn cảnh đã học dưới dạng lưới đặc trưng phân cấp. Từ phải sang trái, quy trình của chúng tôi có thể được hiểu như một mô hình sinh tạo ra các ảnh độ sâu và màu từ một biểu diễn cảnh và vị trí máy ảnh đã cho. Trong thời gian thử nghiệm, chúng tôi ước tính cả biểu diễn cảnh và vị trí máy ảnh bằng cách giải quyết bài toán nghịch đảo thông qua việc lan truyền ngược lỗi tái tạo ảnh và độ sâu qua một bộ kết xuất có thể phân biệt (từ trái sang phải). Cả hai thực thể này được ước tính trong một tối ưu hóa thay phiên: Lập bản đồ: Việc lan truyền ngược chỉ cập nhật biểu diễn cảnh phân cấp; Theo dõi: Việc lan truyền ngược chỉ cập nhật vị trí máy ảnh. Để dễ đọc hơn, chúng tôi kết hợp lưới chi tiết cao dùng để mã hóa hình học với lưới màu có kích thước tương đương và hiển thị chúng như một lưới duy nhất với hai thuộc tính (đỏ và cam).

3.1 Biểu diễn cảnh phân cấp

Bây giờ chúng tôi giới thiệu biểu diễn cảnh phân cấp của chúng tôi kết hợp các đặc trưng lưới đa cấp với các bộ giải mã được huấn luyện trước cho các dự đoán độ chiếm dụng. Hình học được mã hóa thành ba lưới đặc trưng Φ_θ^l và các bộ giải mã MLP tương ứng f^l, trong đó $l \in \{0, 1, 2\}$ được gọi là chi tiết cảnh ở mức thô, trung và tinh. Ngoài ra, chúng tôi cũng có một lưới đặc trưng duy nhất ψ_ω và bộ giải mã g_ω để mô hình hóa ngoại hình của cảnh. Ở đây θ và ω chỉ ra các tham số có thể tối ưu hóa cho hình học và màu sắc, tức là các đặc trưng trong lưới và trọng số trong bộ giải mã màu.

Biểu diễn hình học mức mid và fine. Hình học cảnh quan sát được biểu diễn trong các lưới đặc trưng mức trung và tinh. Trong quá trình tái tạo, chúng tôi sử dụng hai lưới này trong cách tiếp cận từ thô đến tinh, trong đó hình học được tái tạo đầu tiên bằng cách tối ưu hóa lưới đặc trưng mức trung, tiếp theo là sự tinh chỉnh sử dụng mức tinh. Trong cài đặt, chúng tôi sử dụng lưới voxel với độ dài cạnh lần lượt là 32cm và 16cm, ngoại trừ TUM RGB-D, chúng tôi sử dụng 16cm và 8cm. Đối với mức trung, các đặc trưng được giải mã trực tiếp thành giá trị độ chiếm dụng bằng cách sử dụng MLP liên kết f1. Đối với bất kỳ điểm p $\in \mathbb{R}^3$, chúng tôi nhận được độ chiếm dụng là:

$$o_1(p) = f_1\big(p, \phi_1^\theta(p)\big) \tag{1}$$

trong đó $\Phi_\theta^l(p)$ biểu thị rằng lưới đặc trưng được nội suy tam tuyến tại điểm p. Độ phân giải tương đối thấp cho phép chúng tôi tối ưu hóa hiệu quả các đặc trưng lưới để phù hợp với các quan sát. Để nắm bắt các chi tiết tần số cao nhỏ hơn trong hình học cảnh, chúng tôi thêm vào các đặc trưng mức tinh theo cách dư thừa. Cụ thể, bộ giải mã đặc trưng mức tinh lấy đầu vào cả đặc trưng mức trung tương ứng và đặc trưng mức tinh và xuất ra một độ lệch từ độ chiếm dụng mức trung, tức là:

$$\Delta o_1(p) = f_2\big(p, \phi_1^\theta(p), \phi_2^\theta(p)\big) \tag{2}$$

trong đó độ chiếm dụng cuối cùng cho một điểm được cho bởi:

$$o(p) = o_1(p) + \Delta o_1(p) \tag{3}$$

Lưu ý rằng chúng tôi cố định các bộ giải mã được huấn luyện trước f^1 và f^2, và chỉ tối ưu hóa các lưới đặc trưng Φ_θ^1 và Φ_θ^2 trong suốt quá trình tối ưu hóa. Chúng tôi chứng minh rằng điều này giúp ổn định quá trình tối ưu hóa và học hình học nhất quán.

Biểu diễn hình học mức thô. Lưới đặc trưng mức thô nhằm nắm bắt hình học cấp cao của cảnh (ví dụ: tường, sàn, v.v.), và được tối ưu hóa độc lập với mức trung và tinh. Mục tiêu của lưới thô là có thể dự đoán giá trị độ chiếm dụng xấp xỉ bên ngoài hình học đã được quan sát (được mã hóa trong các mức trung/tinh), ngay cả khi mỗi voxel thô chỉ được quan sát một phần. Vì lý do này, chúng tôi sử dụng độ phân giải rất thấp, với độ dài cạnh là 2m trong cài đặt. Tương tự như lưới mức trung, chúng tôi giải mã trực tiếp thành giá trị độ chiếm dụng bằng cách nội suy các đặc trưng và chuyển qua MLP f^0, tức là:

$$o_0(p) = f_0\big(p, \phi_0^\theta(p)\big) \tag{4}$$

Trong quá trình theo dõi, các giá trị độ chiếm dụng mức thô chỉ được sử dụng để dự đoán các phần cảnh chưa được quan sát trước đó. Hình học được dự báo này cho phép chúng tôi theo dõi ngay cả khi một phần lớn của hình ảnh hiện tại chưa được thấy trước đó.

Huấn luyện trước các bộ giải mã đặc trưng. Trong khuôn khổ của chúng tôi, chúng tôi sử dụng ba MLP cố định khác nhau để giải mã các đặc trưng lưới thành giá trị độ chiếm dụng. Các bộ giải mã mức thô và trung được huấn luyện trước như một phần của ConvONet bao gồm một bộ mã hóa CNN và một bộ giải mã MLP. Chúng tôi huấn luyện cả bộ mã hóa/giải mã bằng cách sử dụng hàm mất mát entropy chéo nhị phân giữa giá trị dự đoán và giá trị thực tế, giống như trong [37]. Sau khi huấn luyện, chúng tôi chỉ sử dụng bộ giải mã MLP, vì chúng tôi sẽ tối ưu hóa trực tiếp các đặc trưng để phù hợp với các quan sát trong đường dẫn tái tạo của chúng tôi. Bằng cách này, bộ giải mã được huấn luyện trước có thể tận dụng các đặc tính cụ thể về độ phân giải được học từ tập huấn luyện, khi giải mã các đặc trưng đã tối ưu hóa của chúng tôi.

Chiến lược tương tự được sử dụng để huấn luyện trước bộ giải mã mức tinh, ngoại trừ việc chúng tôi đơn giản nối đặc trưng $\Phi_\theta^1(p)$ từ mức trung cùng với đặc trưng mức tinh $\Phi_\theta^2(p)$ trước khi đưa vào bộ giải mã. Biểu diễn màu sắc. Trong khi chúng tôi chủ yếu quan tâm đến hình học cảnh, chúng tôi cũng mã hóa thông tin màu sắc cho phép chúng tôi hiển thị hình ảnh RGB, cung cấp tín hiệu bổ sung cho việc theo dõi. Để mã hóa màu sắc trong cảnh, chúng tôi áp dụng một lưới đặc trưng khác ψω và bộ giải mã gω:

$$c_p = g_\omega(p, \psi_\omega(p)), \tag{5}$$

Trong đó ω chỉ ra các tham số có thể học được trong quá trình tối ưu hóa. Khác với hình học có kiến thức tiên nghiệm mạnh mẽ, chúng tôi nhận thấy qua thực nghiệm rằng việc tối ưu hóa đồng thời các đặc trưng màu ψω và bộ giải mã gω cải thiện hiệu suất theo dõi (xem Bảng 5). Lưu ý rằng, tương tự như iMAP [46], điều này có thể dẫn đến vấn đề quên và màu sắc chỉ nhất quán cục bộ. Nếu chúng tôi muốn hiển thị màu sắc cho toàn bộ cảnh, nó có thể được tối ưu hóa toàn cục như một bước hậu xử lý.

Thiết kế mạng. Đối với tất cả các bộ giải mã MLP, chúng tôi sử dụng kích thước đặc trưng ẩn là 32 và 5 khối được kết nối đầy đủ. Ngoại trừ biểu diễn hình học mức thô, chúng tôi áp dụng mã hóa vị trí Gaussian có thể học được [46, 49] cho p trước khi đưa vào các bộ giải mã MLP. Chúng tôi nhận thấy điều này cho phép khám phá các chi tiết tần số cao cho cả hình học và ngoại hình.

3.2 Hiển thị độ sâu và màu sắc

Lấy cảm hứng từ thành công gần đây của kỹ thuật hiển thị thể tích trong NeRF [25], chúng tôi đề xuất cũng sử dụng một quy trình hiển thị khả vi phân tích hợp độ chiếm dụng và màu sắc dự đoán từ biểu diễn cảnh của chúng tôi trong Mục 3.1.

Với các tham số nội của máy ảnh và vị trí máy ảnh hiện tại, chúng tôi có thể tính toán hướng quan sát r của tọa độ điểm ảnh. Chúng tôi trước tiên lấy mẫu dọc theo tia này N_{strat} điểm cho việc lấy mẫu phân tầng, và cũng lấy mẫu đồng đều N_{imp} điểm gần độ sâu. Tổng cộng chúng tôi lấy mẫu $N = N_{\text{strat}} + N_{\text{imp}}$ điểm cho mỗi tia. Chính thức hơn, để $p_i = 0 + d_i r$, i ∈ {1, ..., N} biểu thị các điểm lấy mẫu trên tia r với gốc máy ảnh o, và di tương ứng với giá trị độ sâu của p_i dọc theo tia này. Đối với mỗi điểm p_i, chúng tôi có thể tính toán xác suất độ chiếm dụng mức thô $o^0_{p_i}$, xác suất độ chiếm dụng mức tinh o_{p_i}, và giá trị màu sắc c_{p_i} sử dụng Phương trình (4), Phương trình (3), và Phương trình (5). Tương tự như [34], chúng tôi mô hình hóa xác suất kết thúc tia tại điểm p_i là $w^i_c = o_{p_i} \prod_{j=1}^{i-1}\left(1 - o_{p_j}\right)$ cho mức thô, và$w^i_f = o_{p_i} \prod_{j=1}^{i-1}\left(1 - o_{p_j}\right)$ cho mức tinh.

Cuối cùng đối với mỗi tia, độ sâu ở cả mức thô và tinh, và màu sắc có thể được hiển thị như sau:

$$\begin{aligned} \hat{D}^c &= \sum_{i=1}^{N} w^c_i d_i, \\ \hat{D}^f &= \sum_{i=1}^{N} w^f_i d_i, \\ \hat{I} &= \sum_{i=1}^{N} w^f_i c_i \end{aligned} \tag{6}$$

Hơn nữa, chúng tôi cũng tính toán phương sai độ sâu dọc theo tia:

$$\begin{aligned} \hat{D}^c_{var} &= \sum_{i=1}^{N} w^c_i \left(\hat{D}^c - d_i\right)^2, \\ \hat{D}^f_{var} &= \sum_{i=1}^{N} w^f_i \left(\hat{D}^f - d_i\right)^2 \quad (7) \end{aligned} \tag{7}$$

3.3. Lập bản đồ và theo dõi

Trong phần này, chúng tôi cung cấp chi tiết về việc tối ưu hóa các tham số hình học θ và ngoại hình ω của biểu diễn cảnh phân cấp của chúng tôi, và của các vị trí máy ảnh.

Lập bản đồ. Để tối ưu hóa biểu diễn cảnh đã đề cập trong Mục 3.1, chúng tôi lấy mẫu đồng đều tổng cộng M điểm ảnh từ khung hình hiện tại và các khung hình chính được chọn. Tiếp theo, chúng tôi thực hiện tối ưu hóa theo từng giai đoạn để giảm thiểu các tổn thất hình học và quang học.

Tổn thất hình học đơn giản là tổn thất L_1 giữa các quan sát và độ sâu dự đoán ở mức thô hoặc tinh:

$$L_g^l = \frac{1}{M}\sum_{m=1}^{M}\left|D_m - \hat{D}_m^l\right|, \quad l \in \{c, f\} \tag{8}$$

Tổn thất quang học cũng là tổn thất L_1 giữa các giá trị màu sắc được hiển thị và quan sát cho M điểm ảnh được lấy mẫu:

$$L_p = \frac{1}{M}\sum_{m=1}^{M}\left|I_m - \hat{I}_m\right| \tag{9}$$

Ở giai đoạn đầu tiên, chúng tôi chỉ tối ưu hóa lưới đặc trưng mức mid Φ_θ^1 bằng cách sử dụng tổn thất hình học L_g^l trong Phương trình (8). Tiếp theo, chúng tôi tối ưu hóa đồng thời cả đặc trưng mức mid-fine Φ_θ^1, Φ_θ^2 với cùng tổn thất độ sâu mức tinh L_g^l. Cuối cùng, chúng tôi tiến hành điều chỉnh bó cục bộ (BA) để tối ưu hóa đồng thời các lưới đặc trưng ở tất cả các mức, bộ giải mã màu, cũng như các tham số ngoại của máy ảnh $\{R_i, t_i\}$ của K khung hình chính được chọn:

$$\min_{\theta,\omega,\{R_i,t_i\}}\left(L_g^c + L_g^f + \lambda_p L_p\right) \tag{10}$$

trong đó λ_p là hệ số trọng số tổn thất.

Sơ đồ tối ưu hóa đa giai đoạn này dẫn đến sự hội tụ tốt hơn vì các đặc trưng ngoại hình và mức tinh có độ phân giải cao hơn có thể dựa vào hình học đã được tinh chỉnh từ lưới đặc trưng mức trung.

Lưu ý rằng chúng tôi song song hóa hệ thống của chúng tôi trong ba luồng để tăng tốc quá trình tối ưu hóa: một luồng cho lập bản đồ mức thô, một cho tối ưu hóa hình học và màu sắc mức trung & tinh, và một luồng khác cho theo dõi máy ảnh.

Theo dõi máy ảnh. Ngoài việc tối ưu hóa biểu diễn cảnh, chúng tôi cũng chạy song song việc theo dõi máy ảnh để tối ưu hóa các vị trí của máy ảnh cho khung hình hiện tại, tức là quay và dịch chuyển {R, t}. Để làm điều này, chúng tôi lấy mẫu M_t điểm ảnh trong khung hình hiện tại và áp dụng cùng tổn thất quang học trong Phương trình (9) nhưng sử dụng tổn thất hình học được sửa đổi:

$$L_{g-var} = \frac{1}{M_t}\sum_{m=1}^{M_t}\left|D_m - \hat{D}_m^c\right|\frac{1}{\sqrt{\hat{D}_{var}^c}} + \left|D_m - \hat{D}_m^f\right|\frac{1}{\sqrt{\hat{D}_{var}^f}} \tag{11}$$

Tổn thất được sửa đổi giảm trọng số các vùng kém chắc chắn trong hình học được tái tạo [46, 62], ví dụ như các cạnh đối tượng. Việc theo dõi máy ảnh cuối cùng được xây dựng như bài toán tối thiểu hóa sau:

$$\min_{R,t}\left(L_{g-var} + \lambda_{pt} L_p\right) \tag{12}$$

Lưới đặc trưng thô có khả năng thực hiện dự đoán hình học cảnh trong phạm vi ngắn. Hình học ngoại suy này cung cấp tín hiệu có ý nghĩa cho việc theo dõi khi máy ảnh di chuyển

vào các khu vực chưa được quan sát trước đó. Điều này làm cho hệ thống mạnh mẽ hơn đối với mất khung hình đột ngột hoặc chuyển động máy ảnh nhanh. Chúng tôi cung cấp các thí nghiệm trong tài liệu bổ sung.

Khả năng chống chịu các đối tượng động. Để làm cho việc tối ưu hóa mạnh mẽ hơn đối với các đối tượng động trong quá trình theo dõi, chúng tôi lọc các điểm ảnh có tổn thất tái hiển thị độ sâu/màu sắc lớn. Cụ thể, chúng tôi loại bỏ bất kỳ điểm ảnh nào khỏi quá trình tối ưu hóa khi tổn thất Phương trình (12) lớn hơn 10× giá trị trung vị của tổn thất của tất cả các điểm ảnh trong khung hình hiện tại. Hình 6 cho thấy một ví dụ trong đó một đối tượng động bị bỏ qua vì nó không có mặt trong hình ảnh RGB và độ sâu được hiển thị. Lưu ý rằng đối với nhiệm vụ này, chúng tôi chỉ tối ưu hóa biểu diễn cảnh trong quá trình lập bản đồ. Việc tối ưu hóa đồng thời các tham số máy ảnh và biểu diễn cảnh trong môi trường động là không tầm thường, và chúng tôi xem xét nó như một hướng thú vị trong tương lai.

3.3 Lựa chọn khung hình chính

Tương tự như các hệ thống SLAM khác, chúng tôi liên tục tối ưu hóa biểu diễn cảnh phân cấp của chúng tôi với một tập hợp các khung hình chính được chọn. Chúng tôi duy trì một danh sách khung hình chính toàn cục theo tinh thần của iMAP [46], nơi chúng tôi tăng dần thêm các khung hình chính mới dựa trên lợi ích thông tin. Tuy nhiên, trái ngược với iMAP [46], chúng tôi chỉ bao gồm các khung hình chính có sự chồng lấp trực quan với khung hình hiện tại khi tối ưu hóa hình học cảnh. Điều này có thể thực hiện được vì chúng tôi có khả năng thực hiện các cập nhật cục bộ cho biểu diễn dựa trên lưới của chúng tôi, và chúng tôi không gặp phải các vấn đề quên như [46]. Chiến lược lựa chọn khung hình chính này không chỉ đảm bảo hình học bên ngoài góc nhìn hiện tại vẫn tĩnh, mà còn dẫn đến một bài toán tối ưu hóa rất hiệu quả vì chúng tôi chỉ tối ưu hóa các tham số cần thiết mỗi lần. Trong thực tế, chúng tôi trước tiên lấy mẫu ngẫu nhiên các điểm ảnh và chiếu ngược các độ sâu tương ứng bằng cách sử dụng vị trí máy ảnh đã tối ưu hóa. Sau đó, chúng tôi chiếu đám mây điểm đến mọi khung hình chính trong danh sách khung hình chính toàn cục. Từ những khung hình chính có điểm được chiếu lên, chúng tôi chọn ngẫu nhiên K - 2 khung hình. Ngoài ra, chúng tôi cũng bao gồm khung hình chính gần đây nhất và khung hình hiện tại trong quá trình tối ưu hóa biểu diễn cảnh, tạo thành tổng số K khung hình hoạt động. Vui lòng tham khảo Mục 4.4 để biết nghiên cứu loại bỏ về chiến lược lựa chọn khung hình chính.

4 Thí nghiệm

Chúng tôi đánh giá hệ thống SLAM của mình trên nhiều bộ dữ liệu khác nhau, cả thực tế và tổng hợp, với các kích thước và độ phức tạp khác nhau. Chúng tôi cũng tiến hành một nghiên cứu loại trừ toàn diện để hỗ trợ các lựa chọn thiết kế của mình.

4.1 Cài đặt thí nghiệm

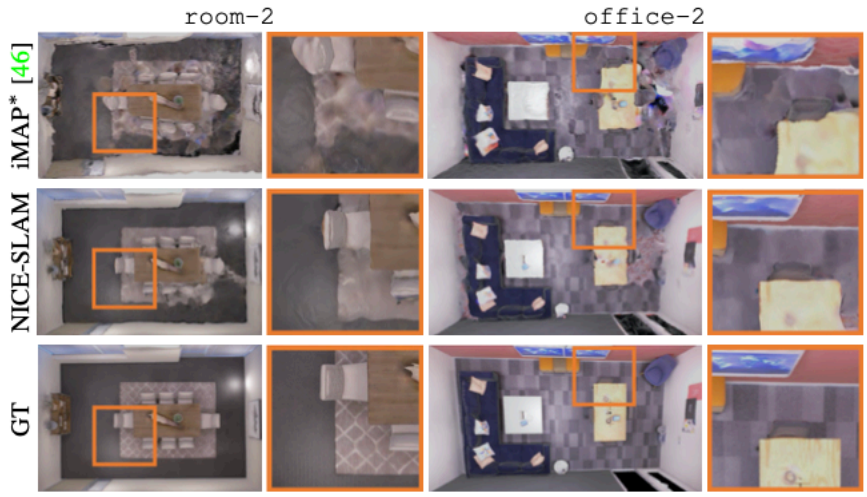
Bộ dữ liệu. Chúng tôi xem xét 5 bộ dữ liệu đa năng: Replica [44], ScanNet [13], bộ dữ liệu TUM RGB-D [45], bộ dữ liệu Co-Fusion [39], cùng với một bộ dữ liệu tự chụp từ một căn hộ lớn với nhiều phòng. Chúng tôi thực hiện các bước tiền xử lý giống như cho bộ dữ liệu TUM RGB-D theo [53].

Cơ sở so sánh. Chúng tôi so sánh với TSDF-Fusion [11] sử dụng các vị trí camera của mình với độ phân giải lưới voxel 256^3 (kết quả với độ phân giải cao hơn được báo cáo trong tài liệu bổ sung), DI-Fusion [16] sử dụng cài đặt chính thức của họ, cùng với iMAP của chúng tôi [46] (phiên bản tái triển khai): iMAP*. Phiên bản tái triển khai của chúng tôi có hiệu suất tương tự như iMAP gốc trong cả tái cấu trúc cảnh và theo dõi camera.

Đo lường. Chúng tôi sử dụng cả chỉ số 2D và 3D để đánh giá hình học cảnh. Đối với chỉ số 2D, chúng tôi đánh giá mất mát L_1 trên 1000 bản đồ độ sâu lấy mẫu ngẫu nhiên từ cả mô hình tái cấu trúc và mô hình thật. Để so sánh công bằng, chúng tôi áp dụng bộ giải đối

xứng [2] cho DI-Fusion [16] và TSDF-Fusion để lấp đầy các lỗ độ sâu trước khi tính toán mất mát L_1 trung bình. Đối với các chỉ số 3D, chúng tôi làm theo [46] và xem xét Độ chính xác [cm], Hoàn thành [cm], và Tỷ lệ hoàn thành [$< 5cm$], ngoại trừ việc chúng tôi loại bỏ các khu vực chưa được nhìn thấy không nằm trong phạm vi quan sát của bất kỳ camera nào. Để đánh giá theo dõi camera, chúng tôi sử dụng ATE RMSE [45]. Nếu không được chỉ định, mặc định chúng tôi báo cáo kết quả trung bình của 5 lần chạy.

Chi tiết triển khai. Chúng tôi chạy hệ thống SLAM của mình trên một PC để bàn với CPU Intel i7-10700K 3.80GHz và GPU NVIDIA RTX 3090. Trong tất cả các thí nghiệm của mình, chúng tôi sử dụng số điểm lấy mẫu trên một tia $N_{\text{strat}} = 32$ và $N_{\text{imp}} = 16$, trọng số mất mát quang học $\lambda_p = 0.2$ và $\lambda_{\text{pt}} = 0.5$. Đối với các bộ dữ liệu tổng hợp quy mô nhỏ (Replica và Co-Fusion), chúng tôi chọn K = 5 keyframes và lấy mẫu M = 1000 và Mt = 200 điểm ảnh. Đối với các bộ dữ liệu thực tế quy mô lớn (ScanNet và cảnh tự ghi lại của chúng tôi), chúng tôi sử dụng K = 10, M = 5000, $M_t = 1000$.



Hình 3: Kết quả tái tạo trên bộ dữ liệu Replica [44]. iMAP* đề cập đến việc tái triển khai iMAP của chúng tôi.

Đối với bộ dữ liệu khó khăn TUM RGB-D, chúng tôi sử dụng K = 10, M = 5000, Mt = 5000. Đối với việc tái triển khai iMAP* của chúng tôi, chúng tôi tuân theo tất cả các siêu tham số đã đề cập trong [46] ngoại trừ việc chúng tôi đặt số điểm ảnh lấy mẫu là 5000 vì nó mang lại hiệu suất tốt hơn trong cả tái tạo và theo dõi.

Bảng 1: iMAP* chỉ đến việc tái triển khai lại iMAP của chúng tôi. TSDF-Fusion sử dụng vị trí camera từ NICE-SLAM.

Method	**Reconstruction Results (average over 8 scenes)**			
2-5	Mem. (MB)	Depth L1 ↓	Acc. ↓	Comp. ↓
TSDF-Fusion [11]	67.10	7.57	1.60	3.49
iMAP [46]	1.04	23.33	6.95	5.33
iMAP* [46]	3.78	23.53	19.40	10.19
DI-Fusion [16]	12.02	3.53	2.85	3.00
NICE-SLAM	12.96	2.85	2.65	3.00

4.2 Đánh giá về Bản đồ hóa và Theo dõi

Đánh giá trên Replica [44]. Để đánh giá trên tập dữ liệu Replica [44], chúng tôi sử dụng cùng chuỗi ảnh RGB-D đã được render do tác giả của iMAP cung cấp. Với biểu diễn cảnh theo phân cấp, phương pháp của chúng tôi có thể tái tạo hình học một cách chính xác trong số vòng lặp giới hạn. Như được trình bày trong Bảng 1, NICE-SLAM vượt trội hơn đáng kể so với các phương pháp nền tảng ở hầu hết các chỉ số, đồng thời duy trì mức tiêu thụ bộ nhớ hợp lý. Về mặt chất lượng, như thể hiện ở Hình 3, phương pháp của chúng tôi tạo ra hình học sắc nét hơn và ít tạo ra các hiện tượng nhiễu (artifact) hơn.

Đánh giá trên TUM RGB-D [45]. Chúng tôi cũng đánh giá hiệu suất theo dõi camera trên tập dữ liệu nhỏ TUM RGB-D. Như thể hiện trong Bảng 2, phương pháp của chúng tôi vượt trội hơn iMAP và DI-Fusion mặc dù về thiết kế, phương pháp của chúng tôi phù hợp hơn với các cảnh lớn. Có thể nhận thấy rằng các phương pháp hiện đại nhất trong việc theo dõi (ví dụ: BAD-SLAM [42], ORB-SLAM2 [26]) vẫn vượt trội hơn các phương pháp dựa trên biểu diễn cảnh ẩn (iMAP [46] và phương pháp của chúng tôi). Tuy nhiên, phương pháp của chúng tôi đã rút ngắn đáng kể khoảng cách giữa hai nhóm phương pháp này, đồng thời vẫn giữ được lợi thế về biểu diễn của mô hình cảnh ẩn.

Bảng 2: **Kết quả theo dõi camera trên ScanNet [13].** Kết quả theo dõi camera trên bộ dữ liệu TUM RGB-D [45]. ATE RMSE [cm] (↓) được sử dụng làm chỉ số đánh giá. NICE-SLAM giảm khoảng cách giữa các phương pháp SLAM với đại diện ẩn và các phương pháp truyền thống. Chúng tôi báo cáo kết quả tốt nhất trong 5 lần chạy cho tất cả các phương pháp trong bảng này. Các số liệu cho iMAP, BAD-SLAM, Kintinuous và ORB-SLAM2 được lấy từ [46].

Scene ID	**fr1/desk**	**fr2/xyz**	**fr3/office**
iMAP [46]	4.9	2.0	5.8
iMAP* [46]	7.2	2.1	9.0
DI-Fusion [16]	4.4	2.3	15.6
NICE-SLAM	2.7	1.8	3.0
BAD-SLAM [42]	1.7	1.1	1.7
Kintinuous [59]	3.7	2.9	3.0
ORB-SLAM2 [26]	1.6	0.4	1.0

Đánh giá trên ScanNet [13]. Chúng tôi chọn nhiều cảnh lớn từ tập dữ liệu ScanNet [13] để đánh giá khả năng mở rộng của các phương pháp khác nhau. Về mặt hình học, như thể hiện ở Hình 4, có thể thấy rõ rằng NICE-SLAM tạo ra hình học sắc nét và chi tiết hơn so với TSDF-Fusion, DI-Fusion và iMAP∗. Về theo dõi, có thể quan sát thấy rằng iMAP∗ và DI-Fusion hoặc là thất bại hoàn toàn hoặc xuất hiện lỗi trôi lớn, trong khi phương pháp của chúng tôi thành công trong việc tái tạo toàn bộ cảnh. Về mặt định lượng, kết quả theo dõi của chúng tôi cũng chính xác hơn đáng kể so với cả DI-Fusion và iMAP∗ như thể hiện trong Bảng 3.

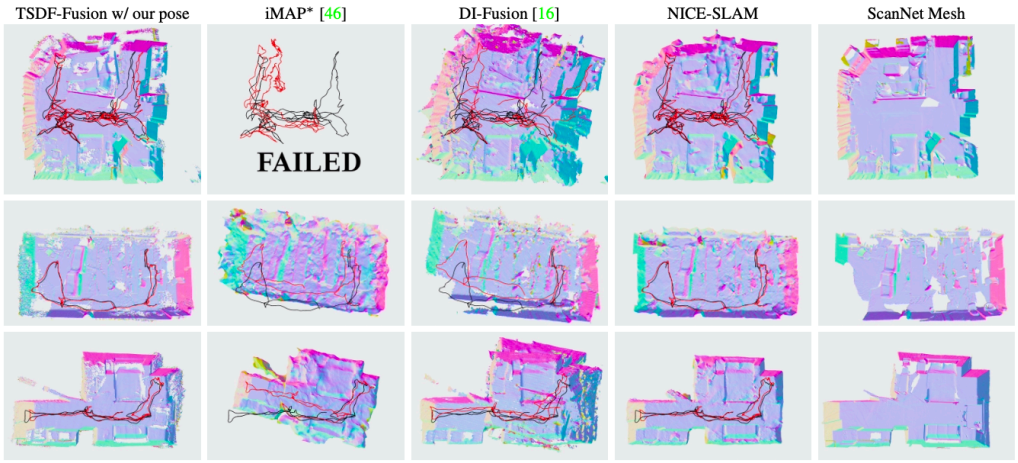
Đánh giá trên Cảnh Lớn hơn. Để đánh giá khả năng mở rộng của phương pháp, chúng tôi đã thu thập một chuỗi dữ liệu trong một căn hộ lớn với nhiều phòng. Hình 1 và Hình 5 cho thấy các bản dựng lại bằng NICE-SLAM, DI-Fusion [16] và iMAP∗[46]. Để tham khảo, chúng tôi cũng trình bày bản dựng 3D bằng công cụ offline Redwood [10] trong Open3D [69]. Có thể thấy rằng NICE-SLAM đạt được kết quả tương đương với phương pháp offline, trong khi iMAP∗ và DI-Fusion không thể dựng lại toàn bộ chuỗi dữ liệu.

Bảng 3: **Camera Tracking trên ScanNet**.Phương pháp của chúng tôi cho kết quả tốt hơn trên bộ dữ liệu này. ATE RMSE (↓) được sử dụng làm chỉ số đánh giá.

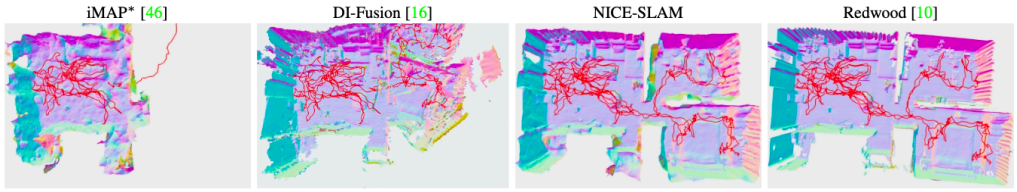
Scene ID	0000	0059	0106	0169	0181	0207	Avg
iMAP* [46]	55.95	32.06	17.50	70.51	32.10	11.91	36.67
DI-Fusion [16]	62.99	128.00	18.50	75.80	87.88	100.19	78.89
NICE-SLAM	8.64	12.25	8.09	10.28	12.93	5.59	9.63

4.3 Phân tích hiệu suất

Ngoài việc đánh giá tái tạo cảnh và theo dõi camera trên các bộ dữ liệu khác nhau, trong phần tiếp theo, chúng tôi cũng đánh giá các đặc điểm khác của pipeline được đề xuất.



Hình 4: Tái tạo 3D và Theo dõi trên ScanNet [13]. Đoạn đường màu đen là kết quả theo dõi từ ScanNet [13], trong khi đoạn đường màu đỏ là kết quả theo dõi của các phương pháp. Chúng tôi đã thử nghiệm nhiều tham số khác nhau cho iMAP* và trình bày các kết quả tốt nhất, nhưng hầu hết đều không đạt.



Hình 5: Tái tạo 3D và Theo dõi trên Căn hộ nhiều phòng. Đoạn đường theo dõi camera được hiển thị bằng màu đỏ. iMAP* và DI-Fusion không thể tái tạo toàn bộ chuỗi. Chúng tôi cũng cho thấy kết quả từ một phương pháp ngoại tuyến [10] để tham khảo.

Bảng 4: Tính toán và Thời gian chạy. Đại diện cảnh của chúng tôi không chỉ cải thiện chất lượng tái tạo và theo dõi mà còn nhanh hơn. Thời gian chạy cho iMAP được lấy từ [46].

Phương pháp	FLOPs [$\times 10^3$]↓	Theo dõi (ms)↓	Tạo bản đồ (ms)↓
iMAP [46]	443.91	101	448
NICE-SLAM	104.16	47	130

Độ phức tạp tính toán. Trước tiên, chúng tôi so sánh số lượng phép toán điểm nổi (FLOPs) cần thiết để truy vấn màu sắc và mật độ chiếm đóng/thể tích của một điểm 3D, xem Bảng 4. Phương pháp của chúng tôi yêu cầu chỉ 1/4 FLOPs so với iMAP. Đáng chú ý, FLOPs trong phương pháp của chúng tôi vẫn giữ nguyên ngay cả đối với những cảnh rất

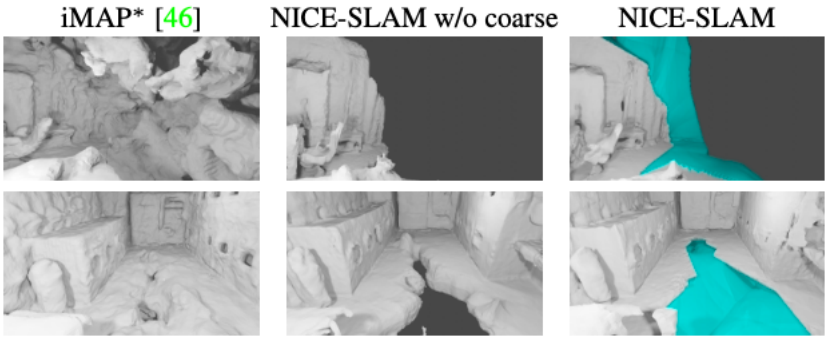
lớn. Ngược lại, do việc sử dụng một MLP duy nhất trong iMAP, giới hạn dung lượng của MLP có thể yêu cầu nhiều tham số hơn dẫn đến nhiều FLOPs hơn.

Thời gian chạy. Chúng tôi cũng so sánh trong Bảng 4 thời gian chạy cho theo dõi và tạo bản đồ sử dụng cùng số lượng mẫu pixel ($M_t = 200$ cho theo dõi và M = 1000 cho tạo bản đồ). Chúng tôi nhận thấy rằng phương pháp của chúng tôi nhanh hơn hơn 2x và 3x so với iMAP trong theo dõi và tạo bản đồ. Điều này chỉ ra lợi thế của việc sử dụng các lưới đặc trưng với bộ giải MLP nông thay vì một MLP nặng.

Độ bền với các đối tượng động. Ở đây, chúng tôi xem xét bộ dữ liệu Co-Fusion [39] chứa các đối tượng di chuyển động. Như minh họa trong Hình 6, phương pháp của chúng tôi đúng đắn nhận diện và bỏ qua các mẫu pixel rơi vào đối tượng động trong quá trình tối ưu hóa, giúp cải thiện mô hình đại diện cảnh (xem các RGB và độ sâu đã được render). Hơn nữa, chúng tôi cũng so sánh với iMAP* trên cùng một chuỗi cho theo dõi camera. Kết quả ATE RMSE của chúng tôi và iMAP* lần lượt là 1.6cm và 7.8cm, điều này rõ ràng chứng minh độ bền của chúng tôi với các đối tượng động.

Dự báo hình học và Lấp đầy lỗ. Như minh họa trong Hình 7, chúng tôi có thể hoàn thiện các khu vực cảnh chưa được quan sát nhờ vào việc sử dụng cảnh prior mức độ thô. Ngược lại, các khu vực chưa thấy được tái tạo lại bởi iMAP* rất ồn ào do không có thông tin prior cảnh nào được mã hóa trong iMAP*.

Hình 6: Độ bền với các đối tượng động. Chúng tôi hiển thị các pixel mẫu được chồng lên một hình ảnh có một đối tượng động ở giữa (bên trái), RGB đã render của chúng tôi (giữa) và độ sâu đã render của chúng tôi (phải) để minh họa khả năng xử lý môi trường động. Các mẫu pixel bị che khuất trong quá trình theo dõi được tô màu đen, trong khi các mẫu đã sử dụng được hiển thị bằng màu đỏ.



Hình 7: Dự báo hình học và Lấp đầy lỗ. Khu vực được tô màu trắng là khu vực có quan sát, và màu xanh lam chỉ ra khu vực chưa được quan sát nhưng đã được dự đoán. Nhờ vào việc sử dụng cảnh prior mức độ thô, phương pháp của chúng tôi có khả năng dự đoán tốt hơn so với iMAP*, điều này cũng cải thiện hiệu suất theo dõi của chúng tôi.

4.4 Nghiên cứu loại trừ

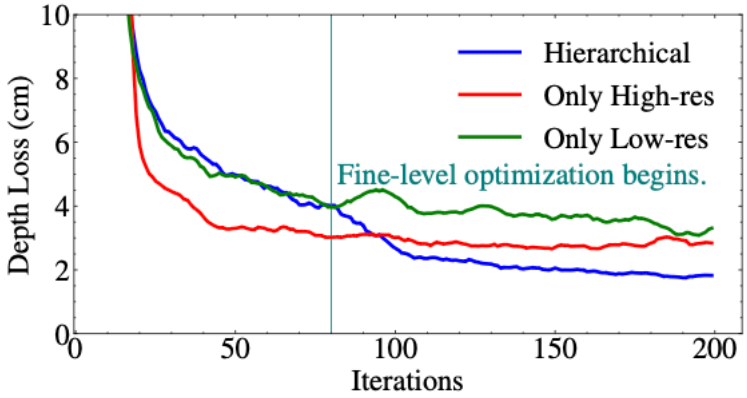
Trong phần này, chúng tôi nghiên cứu sự lựa chọn của kiến trúc phân cấp và tầm quan trọng của đại diện màu sắc.

Kiến trúc phân cấp. Hình 8 so sánh kiến trúc phân cấp của chúng tôi với: a) một lưới đặc trưng với độ phân giải giống như đại diện cấp thấp của chúng tôi (Chỉ độ phân giải cao); b) một lưới đặc trưng với độ phân giải mức trung bình (Chỉ độ phân giải thấp). Kiến trúc phân cấp của chúng tôi có thể nhanh chóng thêm các chi tiết hình học khi đại diện cấp thấp tham gia vào tối ưu hóa, điều này cũng dẫn đến việc hội tụ tốt hơn.

BA cục bộ. Chúng tôi xác minh hiệu quả của việc điều chỉnh theo nhóm cục bộ trên ScanNet [13]. Nếu chúng tôi không tối ưu hóa đồng thời các vị trí camera cho K keyframes cùng với đại diện cảnh (không có BA cục bộ trong Bảng 5), việc theo dõi camera không chỉ kém chính xác hơn mà còn kém bền vững.

Đại diện màu sắc. Trong Bảng 5, chúng tôi so sánh phương pháp của chúng tôi mà không có mất mát quang học L_p trong phương trình (9). Kết quả cho thấy mặc dù các màu sắc ước tính của chúng tôi không hoàn hảo do ngân sách tối ưu hóa hạn chế và thiếu điểm mẫu, việc học một đại diện màu sắc vẫn đóng vai trò quan trọng trong việc theo dõi camera chính xác.

Lựa chọn keyframe. Chúng tôi kiểm tra phương pháp của mình bằng cách sử dụng chiến lược lựa chọn keyframe của iMAP (với keyframe iMAP trong Bảng 5) nơi chúng chọn keyframes từ toàn bộ cảnh. Điều này là cần thiết cho iMAP để ngăn chặn MLP đơn giản của họ quên đi hình học trước đó. Tuy nhiên, nó cũng dẫn đến hội tụ chậm và theo dõi không chính xác.



Hình 8: Thử nghiệm với Kiến trúc phân cấp. Tối ưu hóa hình học trên một ảnh độ sâu duy nhất của Replica [44] với các kiến trúc khác nhau. Các đường cong đã được làm mịn để dễ dàng hình dung hơn.

Bảng 5: **Nghiên cứu loại trừ.** Chúng tôi nghiên cứu tính hữu ích của BA cục bộ, đại diện màu sắc, cũng như chiến lược lựa chọn keyframe của chúng tôi. Chúng tôi chạy mỗi cảnh 5 lần và tính toán giá trị trung bình và độ lệch chuẩn của ATE RMSE (↓) trên 6 cảnh trong ScanNet [13].

ATE RMSE (↓)	w/o Local-BA	w/o $\mathcal{L}_p$	w/ iMAP-keyframes	Full
Mean	37.74	32.02	12.10	9.63
Std.	30.97	21.98	3.38	0.62

Tham khảo

[1] J. T. Barron and B. Poole, *The Fast Bilateral Solver*. Springer, 2016, p. 5.

[2] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, “Neural RGB-D Surface Reconstruction,” p. 2, 2022.

A Phụ lục

[1]

[2]