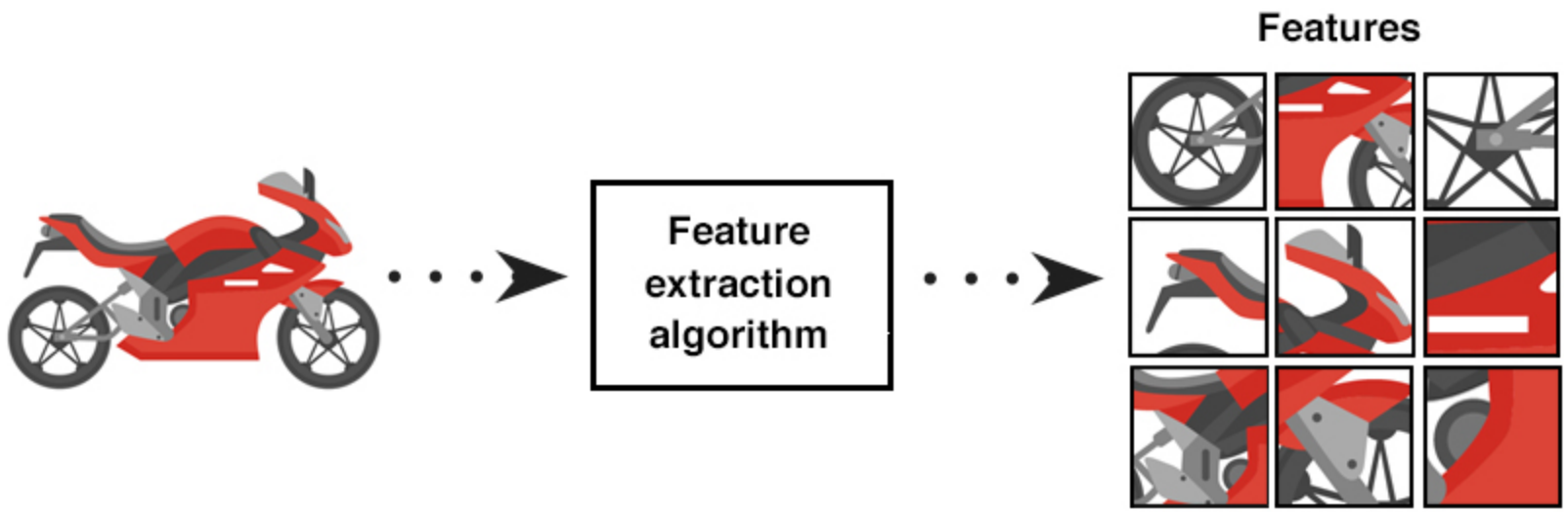
Phần 1: Giai đoạn 2 - Trích xuất Đặc trưng Cấp thấp Mạnh Mẽ (Nền tảng Perception)

Giới thiệu Giai đoạn 2: Đặc trưng là gì và tại sao lại cần?

Các điểm ảnh thô chứa quá nhiều thông tin và nhạy cảm với nhiễu, thay đổi ánh sáng. Giai đoạn 2 tìm ra các điểm/khu vực **độc đáo và ổn định (distinctive & robust)** trong ảnh hoặc dữ liệu 3D (đặc trưng). Những đặc trưng này giúp theo dõi chuyển động của camera/robot hoặc tìm các điểm tương ứng giữa các hình ảnh khác nhau.
Ước tính thời gian học: 1 giờ


Features
Feature extraction algorithm

Nền tảng Toán: Đại số Tuyến tính cho Biến đổi & Đặc trưng

Đại số Tuyến tính là ngôn ngữ của các phép biến đổi hình học và biểu diễn vector. Điểm, vector hướng, và các phép biến đổi (quay, tịnh tiến, tỷ lệ) đều được biểu diễn bằng ma trận và vector. Bộ mô tả đặc trưng (feature descriptor) thường là vector.
Khái niệm chính: **Vector, Ma trận, Phép nhân ma trận, Biến đổi affin/hình học.**
Ước tính thời gian học: 7 giờ

Nền tảng Toán: Giải Tích cho Gradient và Tối ưu

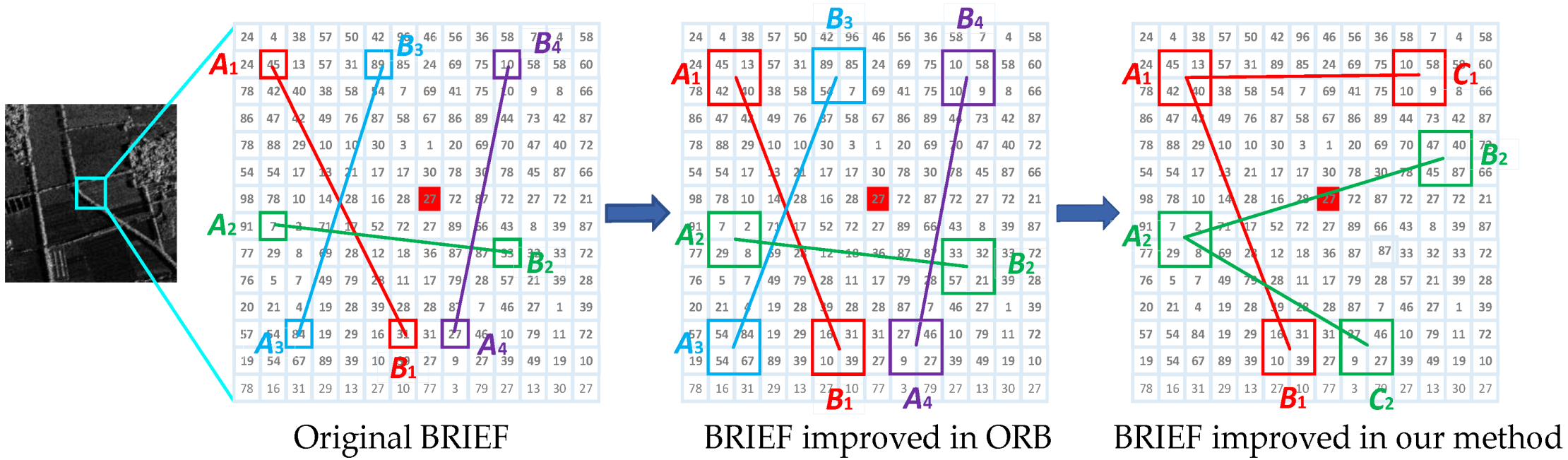
Giải Tích giúp xác định sự thay đổi và nền tảng cho việc tối ưu. Phát hiện các khu vực có sự thay đổi mạnh về cường độ (như biên) liên quan đến đạo hàm và gradient. Đặc biệt, Gradient Descent, phương pháp tối ưu cốt lõi, dựa trên việc tính gradient của hàm mất mát để điều chỉnh tham số mô hình học sâu trích xuất đặc trưng.
Khái niệm chính: **Đạo hàm, Gradient, Gradient Descent.**
Ước tính thời gian học: 3 giờ

Nền tảng Toán: Xác suất & Thống kê Cơ bản

Xác suất & Thống kê giúp mô hình hóa sự không chắc chắn và nhiễu. Việc so sánh các bộ mô tả đặc trưng thường dùng tiêu chí khoảng cách thống kê (ví dụ khoảng cách Hamming). Phân tích thống kê lân cận điểm trong dữ liệu 3D cũng thuộc về mảng này.
Khái niệm chính: **Xác suất, Phân phối (đơn giản), Khoảng cách thống kê.**
Ước tính thời gian học: 3 giờ

Thuật toán: ORB (Oriented FAST and Rotated BRIEF)

ORB là sự kết hợp hiệu quả giữa thuật toán phát hiện điểm FAST cực nhanh và bộ mô tả nhị phân BRIEF có tính đến hướng để chống xoay. Nhanh và miễn phí, rất phổ biến cho **Visual SLAM** thời gian thực trên phần cứng hạn chế.
Nguyên lý Toán: Phát hiện **FAST** dựa trên so sánh ngưỡng độ sáng. Mô tả **rBRIEF** dựa trên cặp điểm ngẫu nhiên và hướng tính từ mô men ảnh.
Độ phức tạp Thời gian: **O(N)** (N số pixel) cho phát hiện, mô tả cũng rất nhanh. **Hiệu suất Thời gian thực tốt**.
Ước tính thời gian học: 5 giờ



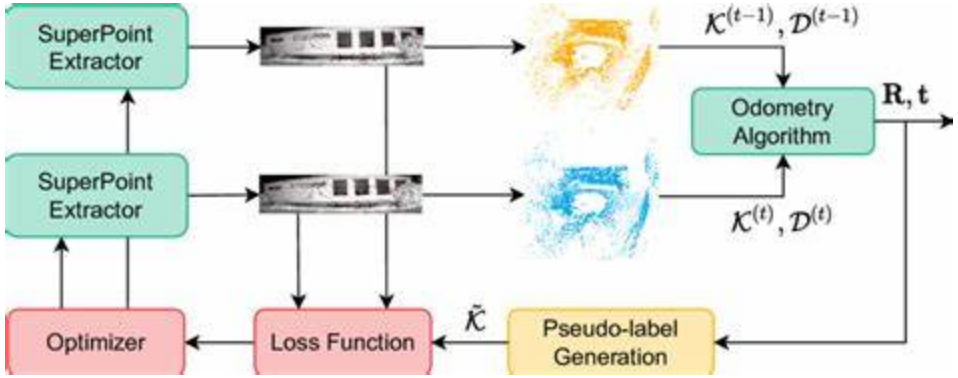
Original BRIEF

BRIEF improved in ORB

BRIEF improved in our method

Thuật toán: SuperPoint

Một phương pháp học sâu (dựa trên **CNN**) phát hiện và tạo bộ mô tả đặc trưng cùng lúc chỉ với một lần suy luận. SuperPoint tìm các đặc trưng có tính lặp lại cao giữa các góc nhìn khác nhau, mang lại hiệu quả cao hơn nhiều thuật toán truyền thống trong việc khớp đặc trưng.
Nguyên lý Toán: Sử dụng **CNN** (dựa trên **Đại số Tuyến tính**), được huấn luyện bằng **Tối ưu hóa (Gradient Descent)** và hàm mất mát đa nhiệm (đa phần dựa trên **Xác suất và khoảng cách**).
Độ phức tạp Thời gian: Yêu cầu **GPU/NPU** cho suy luận. **Thời gian thực khả thi** với phần cứng mạnh.
Ước tính thời gian học: 7 giờ



Toán & Thuật toán: Làm việc với Đặc trưng 3D từ Đám mây điểm

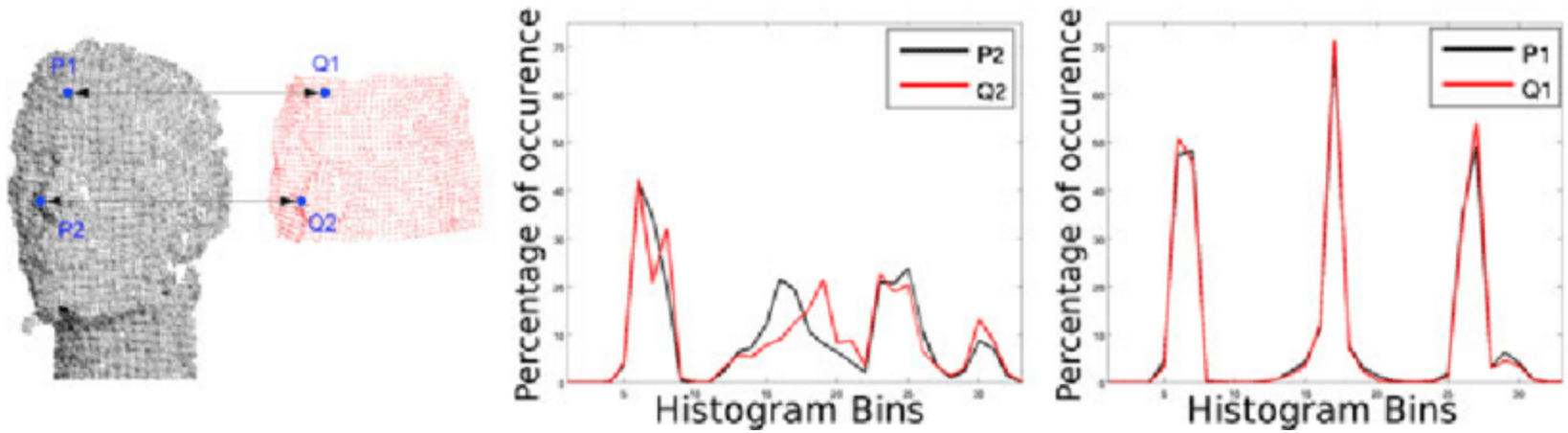
Xử lý dữ liệu từ **LiDAR** hoặc **Depth Camera** đòi hỏi kỹ thuật làm việc trực tiếp trên **đám mây điểm (Point Cloud)**. Các đặc trưng có thể được trích xuất tại các điểm trong không gian 3D.

Nền tảng Toán: **Đại số Tuyến tính** (phân tích lân cận điểm - **PCA** để tính pháp tuyến), **Hình học 3D** (tính khoảng cách, góc).

Ước tính thời gian học: 4 giờ

Thuật toán: FPFH (Fast Point Feature Histograms)

Một bộ mô tả đặc trưng cho **đám mây điểm 3D**. Nó ghi lại hình dạng cục bộ của đám mây điểm tại một điểm quan tâm bằng cách phân tích mối quan hệ không gian (độ lệch góc giữa các vector pháp tuyến, sự khác biệt về vector pháp tuyến) giữa điểm đó và các điểm lân cận. Biểu diễn dạng **Histogram** (liên quan **Thống kê**).
Nguyên lý Toán: Tính **Normal Vectors** (pháp tuyến) dựa trên **PCA**. Tính toán các quan hệ góc và biểu diễn bằng **histogram**.
Độ phức tạp Thời gian: Phụ thuộc kích thước lân cận và cách tìm điểm lân cận (thường dùng KD-Tree: O(N log k) - N điểm, k lân cận). **Khá hiệu quả**.
Ước tính thời gian học: 4 giờ


P1
Q1
P2
Q2
Percentage of occurence
Histogram Bins
P2
Q2
Percentage of occurence
Histogram Bins
P1
Q1

Chuyển tiếp Giai đoạn 2 -> Giai đoạn 3: Xây dựng Sự Hiểu biết

Đặc trưng thô là nền tảng. Giai đoạn 3 sẽ nhóm, phân loại và diễn giải những đặc trưng này, kết hợp thêm thông tin không gian 3D từ cảm biến để tạo ra những biểu diễn có ý nghĩa hơn: vật thể, phân đoạn, cấu trúc không gian.
Ước tính thời gian học: 1 giờ

Phần 2: Giai đoạn 3 - Nhận thức Cấp trung: Vật thể, Không gian, Cấu trúc

Giới thiệu Giai đoạn 3: Nhìn xa hơn các Đặc trưng

Giai đoạn 3 kết hợp các đặc trưng cấp thấp và dữ liệu thô/đã xử lý để nhận diện và định vị các "đối tượng" quan trọng trong môi trường: các loại vật thể cụ thể (người, xe, ghế...), các khu vực có ý nghĩa (sàn, tường), và cấu trúc 3D của cảnh. Đây là "mắt" của robot bắt đầu "hiểu" cảnh.
Ước tính thời gian học: 1 giờ

Toán Trọng tâm DL: Đại số Tuyến tính cho Các phép toán Mạng

Các thuật toán SoTA Giai đoạn 3 đa phần dựa trên Học sâu (CNN). **Phép tích chập (convolution)** và **phép nhân ma trận** là các phép tính cơ bản trong các lớp của mạng nơ-ron.
Khái niệm chính: **Tích chập (Convolution), Nhân ma trận, Biểu diễn feature maps.**
Ước tính thời gian học: 3 giờ

Toán Trọng tâm DL: Tối ưu hóa cho Huấn luyện Mô hình

Việc "dạy" mạng nơ-ron phát hiện/phân đoạn vật thể là một bài toán **tối ưu hóa**. Mục tiêu là tìm bộ tham số mạng (các trọng số) sao cho hàm mất mát (đo lường sự khác biệt giữa dự đoán và ground truth) là nhỏ nhất. **Gradient Descent** là thuật toán tối ưu chính được sử dụng.

Khái niệm chính: **Hàm mất mát (Loss function), Gradient Descent (và biến thể Adam/RMSprop).**

Ước tính thời gian học: 3 giờ

Toán Trọng tâm DL: Xác suất ở Đầu ra Mô hình

Lớp cuối của mạng nơ-ron cho các bài toán phân loại/phân đoạn thường đưa ra **phân phối xác suất** (thường dùng hàm **softmax**) cho từng lớp/đối tượng tại mỗi vị trí pixel/vùng. Kết quả phát hiện vật thể cũng kèm theo **điểm tin cậy (confidence score)** là xác suất vật thể thuộc về lớp đó.
Khái niệm chính: **Xác suất, Softmax, Điểm tin cậy.**
Ước tính thời gian học: 3 giờ

Toán: Hình học 3D & Hình học Đa Hình Chiếu

Làm việc với dữ liệu **đa trường** (đặc biệt RGB-D/Stereo) đòi hỏi hiểu cách ánh xạ giữa thế giới 3D và ảnh 2D (**phép chiếu**), cách ước lượng độ sâu từ hai ảnh (**Stereo Vision**, dựa trên **Epipolar Geometry**) và cách biểu diễn/biến đổi các vật thể trong không gian 3D. **Rất quan trọng** để kết hợp thông tin từ camera và cảm biến độ sâu/LiDAR.
Khái niệm chính: **Camera Pinhole Model, Hệ tọa độ Camera/Thế giới, Phép chiếu, Epipolar Geometry, Biến đổi vật rắn (SE(3)).**
Ước tính thời gian học: 7 giờ

Thuật toán: Phát hiện Vật thể Thời gian Thực - YOLO / SSD

Đây là các thuật toán **học sâu một giai đoạn (one-stage)** cực kỳ nhanh cho phép phát hiện vật thể trong ảnh với tốc độ cao. Chúng dự đoán trực tiếp vị trí và lớp của các vật thể chỉ sau một lần truyền qua mạng, lý tưởng cho các ứng dụng robot cần phản ứng nhanh.
Nguyên lý Toán: Dựa trên **CNNs** được huấn luyện bằng **Tối ưu hóa**. Đầu ra trực tiếp là các **bounding box** và **điểm tin cậy (xác suất)** lớp tại nhiều vị trí và kích thước (anchors/prior boxes).
Độ phức tạp Thời gian: **Rất thấp** cho mỗi khung hình (sau huấn luyện), **O(Image Size)**. Yêu cầu **GPU/NPU**. **Thời gian thực tốt nhất** hiện nay cho phát hiện 2D.
Ước tính thời gian học: 6 giờ

Thuật thuật toán: Phân đoạn Ngữ nghĩa Thời gian Thực - BiSeNet / ERFNet

Các kiến trúc mạng học sâu này được thiết kế đặc biệt để cân bằng giữa độ chính xác phân đoạn ngữ nghĩa và tốc độ suy luận. Chúng thường sử dụng các kiến trúc mạng tối ưu, ví dụ các nhánh xử lý song song (như BiSeNet) để nắm bắt cả thông tin ngữ cảnh rộng và chi tiết nhanh chóng. Quan trọng để robot biết đâu là vùng "đi lại", "tránh né".
Nguyên lý Toán: Dựa trên các kiến trúc **CNN/FCN/U-Net lightweight**, sử dụng các lớp **tích chập (Đại số)** và **giải chập/upsampling**. Output là **mask phân loại pixel** (dùng **softmax - Xác suất**). Huấn luyện dùng **Tối ưu hóa**.
Độ phức tạp Thời gian: **O(Image Size)**. **Đạt thời gian thực** trên **GPU/NPU** trung bình-yếu hơn so với Detection.
Ước tính thời gian học: 5 giờ

Thuật toán: Phân đoạn Cá thể Thời gian Thực - YOLACT

Phân đoạn cá thể (Instance Segmentation) không chỉ phân loại pixel mà còn tách riêng từng cá thể vật thể cùng loại. YOLACT là một trong những nỗ lực đầu tiên để đạt tốc độ thời gian thực cho Instance Segmentation bằng cách kết hợp nhánh dự đoán mask với các proposal của detector. Cần thiết khi robot cần tương tác với từng vật thể cụ thể (ví dụ gắp chai nước A chứ không phải B).
Nguyên lý Toán: Kết hợp **Phát hiện Vật thể** (dựa trên CNN, **Đại số Tuyến tính, Tối ưu hóa**) và tạo **mask nhị phân** cho từng proposal (sử dụng thêm các lớp tích chập nhỏ).
Độ phức tạp Thời gian: **Nhanh hơn** các phương pháp 2 giai đoạn (như Mask R-CNN), **gần thời gian thực** nhưng **chậm hơn** Semantic Segmentation/Detection thuần. Yêu cầu **GPU/NPU mạnh hơn**.
Ước tính thời gian học: 4 giờ

Toán: Nguyên lý RANSAC cho Khớp Mạnh mẽ

RANSAC (RANdom SAmple Consensus) là một kỹ thuật **thống kê** lặp lại để ước lượng các tham số của một mô hình toán học từ dữ liệu có lẫn rất nhiều "nhiễu" hoặc "ngoại lai (outliers)". Trong thị giác, nó thường dùng để tìm ma trận biến đổi giữa các điểm đặc trưng khớp nối, hoặc tìm mặt phẳng trong đám mây điểm, bỏ qua các điểm khớp sai hoặc điểm nhiễu.

Nguyên lý Toán: Dựa trên việc lấy **mẫu ngẫu nhiên** dữ liệu (phù hợp với **Xác suất**), ước lượng mô hình, đếm số điểm "đồng thuận" (inliers) nằm trong một ngưỡng (**Thống kê**), và lặp lại để tìm mô hình tốt nhất.

Ước tính thời gian học: 3 giờ

Thuật toán: RANSAC

Triển khai cụ thể của nguyên lý RANSAC. Được dùng rộng rãi trong **khớp đặc trưng** (ví dụ: tìm Homography hoặc Ma trận Cơ bản giữa hai tập điểm đặc trưng), **tìm mặt phẳng** trong đám mây điểm, v.v., bất cứ khi nào cần khớp một mô hình hình học trên dữ liệu nhiều nhiễu.
Nguyên lý Toán: **Thuật toán lặp** thực hiện nguyên lý **RANSAC**. Cần hiểu mô hình đang khớp (ví dụ: Homography, Planar model - dựa trên **Đại số Tuyến tính**), và tiêu chí tính "đồng thuận" (dựa trên **Hình học** và ngưỡng **Thống kê**).
Độ phức tạp Thời gian: Số lần lặp **không cố định**, phụ thuộc vào tỷ lệ outlier trong dữ liệu. Mỗi lần lặp tính toán nhanh. Nhìn chung **khá hiệu quả và cần thiết** cho tính mạnh mẽ.
Ước tính thời gian học: 5 giờ

Thuật toán: ICP (Iterative Closest Point)

Thuật toán cổ điển nhưng cốt lõi để **đăng ký (align) hai đám mây điểm 3D**. ICP tìm phép biến đổi vật rắn (quay + tịnh tiến) tối ưu để chồng hai đám mây điểm lên nhau sao cho tổng khoảng cách giữa các điểm tương ứng (nearest neighbors) là nhỏ nhất. Rất quan trọng trong việc xây dựng bản đồ 3D từ nhiều lần quét hoặc định vị robot bằng cách khớp quét mới với bản đồ hiện có.
Nguyên lý Toán: Thuật thuật toán **lặp** sử dụng **Bình phương nhỏ nhất (Least Squares)** để **tối ưu hóa** phép biến đổi (**Rigid Body Transform - Đại số Tuyến tính, SE(3)**) dựa trên việc tìm các điểm lân cận gần nhất trong mỗi lần lặp (liên quan tìm kiếm không gian hiệu quả - cây dữ liệu như KD-Tree).
Độ phức tạp Thời gian: Độ phức tạp mỗi lần lặp O(N) (linear với số điểm), tổng thời gian phụ thuộc số lần lặp để hội tụ. Cần triển khai tìm lân cận hiệu quả. **Có thể chạy thời gian thực** trên đám mây điểm thưa hoặc được xử lý hiệu quả.
Ước tính thời gian học: 6 giờ

Thuật toán: Xử lý Đám mây điểm học được - PointNet++ / RandLA-Net

Các kiến trúc mạng học sâu được thiết kế để xử lý **trực tiếp dữ liệu đám mây điểm 3D**, không cần chuyển đổi sang định dạng khác (voxel hay ảnh 2D). Chúng học cách phân loại từng điểm (phân đoạn ngữ nghĩa) hoặc phân cụm các điểm thuộc cùng một vật thể (phân đoạn cá thể 3D), phát hiện vật thể 3D, v.v. Đây là SoTA cho nhận thức trên dữ liệu 3D nguyên thủy.
Nguyên lý Toán: Dựa trên kiến trúc mạng nơ-ron có khả năng xử lý dữ liệu không có cấu trúc lưới (non-grid structure) của đám mây điểm. Vẫn dựa trên **Đại số Tuyến tính, Tối ưu hóa**. RandLA-Net sử dụng **lấy mẫu ngẫu nhiên** (liên quan **Xác suất/Thống kê**) để giảm kích thước dữ liệu hiệu quả cho các đám mây điểm lớn.
Độ phức tạp Thời gian: Yêu cầu **GPU/NPU mạnh**. Độ phức tạp phụ thuộc kiến trúc, **thường tính toán nặng** hơn xử lý ảnh 2D. RandLA-Net được thiết kế để xử lý các đám mây điểm rất lớn hiệu quả hơn các mô hình cũ như PointNet++. **Thời gian thực là một thách thức** nhưng đang có nhiều tiến bộ.

Bộ Dữ liệu Quan trọng cho Giai đoạn 2 & 3

Để kiểm thử và huấn luyện các thuật toán trên, cần dữ liệu đa trường chất lượng cao.

- **TUM RGB-D:** RGB + Depth + Ground truth tư thế. Tuyệt vời cho VO/SLAM RGB-D, kiểm thử phát hiện/phân đoạn 2D-áp dụng-3D.
- **KITTI:** Stereo Camera + LiDAR + IMU + GPS + Ground truth cho odometry/SLAM/Detection 3D. Lý tưởng cho phát triển ngoài trời, xử lý LiDAR, Stereo Vision.
- **EuRoC MAV:** Stereo Camera + IMU chính xác cao + Ground truth tư thế. Tuyệt vời cho VI-SLAM/VIO.
- **ScanNet / MatterPort3D:** RGB-D + Annotation Segmentation + Mô hình 3D. Tốt cho phát triển nhận thức vật thể và phân đoạn trong nhà.

Chọn bộ dữ liệu phù hợp giúp bạn tập trung vào khía cạnh cảm biến và môi trường mà robot mục tiêu sẽ hoạt động.