Báo cáo Kỹ thuật: Chi Tiết Giai Đoạn 2 và 3 - Trích Xuất Đặc trưng và Nhận thức Cấp trung (Tập trung Toán học, Hiệu suất Thời gian)

Phần này tập trung phân tích chuyên sâu Giai đoạn 2 và Giai đoạn 3 trong quy trình Thị giác Máy tính cho robot, làm rõ vai trò của **toán học**, các thuật toán **hiện đại (SoTA)** có tính đến **độ phức tạp thời gian (time complexity)**, và cách chúng liên kết để xử lý dữ liệu **đa trường**.

Phần 1: Giai đoạn 2 - Trích xuất Đặc trưng Cấp thấp Mạnh Mẽ và Hiệu quả

- **Mục đích:** Tìm và mô tả các yếu tố cơ bản, ổn định (feature points, keypoints, descriptors) trong dữ liệu ảnh và dữ liệu 3D. Những đặc trưng này là nền tảng cho việc theo dõi chuyển động, định vị và tạo bản đồ (odometry và SLAM). Thay vì chỉ phát hiện cạnh (như Canny), chúng ta cần các đặc trưng *khác biệt* có thể *khớp nối* được giữa các khung hình hoặc các góc nhìn khác nhau.

- **Nền tảng Toán học Chủ chốt:**

    - **Đại số Tuyến tính (Linear Algebra):**
        - *Tầm quan trọng:* Biểu diễn và thực hiện các phép biến đổi hình học. Biểu diễn đặc trưng dưới dạng vector (descriptor). Nền tảng cho các phương pháp phân tích thành phần (PCA) để nén hoặc cải thiện đặc trưng. **Phép nhân ma trận** là phép tính cốt lõi trong nhiều mô hình học sâu (được sử dụng để trích xuất đặc trưng).
    - **Xác suất và Thống kê (Probability and Statistics):**
        - *Tầm quan trọng:* Mô hình hóa nhiễu trong dữ liệu cảm biến. Các thuật toán lấy mẫu (sampling) ngẫu nhiên để tìm mô hình phù hợp (**RANSAC** - dù thường dùng ở Giai đoạn 3/4 nhưng ý tưởng thống kê từ đây). So sánh các bộ mô tả (descriptor) thường dùng các tiêu chí khoảng cách thống kê (ví dụ: khoảng cách Hamming cho bộ mô tả nhị phân). Hiểu phân phối dữ liệu đặc trưng để phân biệt giữa đặc trưng tốt và kém.
    - **Giải Tích (Calculus):**
        - *Tầm quan trọng:* Tính gradient để phát hiện những thay đổi mạnh về cường độ (liên quan gốc tới việc tìm biên/góc). Nền tảng cho các thuật toán tối ưu hóa để tinh chỉnh vị trí đặc trưng (ví dụ: phát hiện đặc trưng dưới mức pixel - sub-pixel). **Đạo hàm** và **Gradient Descent** là nền tảng để huấn luyện các mô hình học sâu phát hiện đặc trưng (SuperPoint).

- **Hiệu suất Thời gian (Time Complexity) & Lựa chọn Thuật toán:** Độ phức tạp thời gian ở giai đoạn này rất quan trọng vì nó thường là bước đầu tiên xử lý dữ liệu thô tốc độ cao từ cảm biến. Robot di chuyển nhanh đòi hỏi đặc trưng phải được trích xuất với độ trễ thấp. Chúng ta cần các thuật toán có độ phức tạp *thấp* và có thể chạy *song song* trên GPU/NPU.

- **Các Thuật toán Đáng chú ý (Bỏ qua Canny):**

    1. **ORB (Oriented FAST and Rotated BRIEF)**
        - *Mô tả:* Một trong những thuật toán phát hiện và mô tả đặc trưng **Nhanh và Mạnh mẽ (robust)**. Kết hợp bộ phát hiện **FAST** (cực nhanh dựa trên so sánh pixel) và bộ mô tả nhị phân **BRIEF** đã được định hướng để kháng với xoay (**rotated BRIEF** - R-BRIEF).
        - *Nền tảng Toán học:*
            - **FAST Detector:** Dựa trên so sánh ngưỡng **độ sáng pixel** (thống kê đơn giản, không có toán phức tạp trực tiếp ở đây).

        - **R-BRIEF Descriptor:** Dựa trên một tập hợp các phép so sánh ngẫu nhiên cặp pixel trong vùng lân cận của đặc trưng. Việc thêm định hướng sử dụng mô men (liên quan tích phân/giải tích) để xác định hướng chính. Bộ mô tả là vector **nhị phân**, việc so sánh dùng khoảng cách **Hamming** (dựa trên phép XOR - một dạng thống kê/lý thuyết thông tin đơn giản).
    - *Độ phức tạp Thời gian:* **Thấp**, **Rất nhanh** cho phát hiện và mô tả. **Phát hiện:** Gần O(N) (linear với số pixel), mặc dù trong thực tế có overhead. **Mô tả:** O(k * N) nơi k là số điểm so sánh trong BRIEF. **Khớp đặc trưng (Matching):** Có thể nhanh dùng **khoảng cách Hamming** và cấu trúc dữ liệu như **LSH (Locality-Sensitive Hashing - liên quan Xác suất/Thống kê)** hoặc dùng **Brute Force Matching** với khoảng cách Hamming. Phù hợp với **thời gian thực**.
    - *Kết nối Đa trường:* Có thể chạy trên ảnh màu hoặc ảnh xám từ camera RGB-D. Với ảnh depth, có thể tính gradient depth để phát hiện "đặc trưng độ sâu" tương tự.
2. **SuperPoint: Live Visual Localization from Learnable Features**
    - *Mô tả:* Một phương pháp **học sâu (Deep Learning)** trích xuất cả **điểm đặc trưng (keypoints)** và **bộ mô tả (descriptors)** mạnh mẽ chỉ trong một lần chuyển tiếp mạng (forward pass). Điểm mạnh là tính lặp lại cao của các điểm đặc trưng ngay cả khi thay đổi góc nhìn đáng kể.
    - *Nền tảng Toán học:*
        - **CNNs (Convolutional Neural Networks):** Cốt lõi dựa trên **Đại số Tuyến tính** (phép tích chập - convolution, là phép nhân ma trận) và **Giải Tích** (tính toán gradient, quy tắc chuỗi) để huấn luyện qua **Gradient Descent** (phương pháp **tối ưu hóa**). Kiến trúc cụ thể có thể bao gồm các khối chuẩn hóa (**normalization** - liên quan Thống kê), các hàm kích hoạt phi tuyến.
        - **Multi-task Learning Loss:** Hàm mất mát kết hợp từ hai nhiệm vụ: dự đoán điểm đặc trưng (có thể là phân loại pixel - Xác suất) và dự đoán bộ mô tả (sử dụng loss function so sánh các bộ mô tả - ví dụ **Triplet Loss, Hinge Loss**, dựa trên khoảng cách Euclid/Cosine - Đại số Tuyến tính và Tối ưu hóa).
    - *Độ phức tạp Thời gian:* Tính toán **NẶNG HƠN** ORB truyền thống và **yêu cầu GPU/NPU** để đạt thời gian thực. Tuy nhiên, sau khi huấn luyện, thời gian **suy luận (inference)** khá nhanh và song song hóa tốt trên phần cứng phù hợp. So với các phương pháp học sâu cũ hơn, SuperPoint được thiết kế để đủ nhanh cho nhiều ứng dụng SLAM/VO thời gian thực. **Khớp đặc trưng:** Giống như ORB, dựa trên khoảng cách (Euclid cho SuperPoint descriptor) và các phương pháp khớp hiệu quả (Brute Force with ratio test - Thống kê, FLANN - KD-trees/kmeans liên quan Đại số Tuyến tính và Thuật toán cây).
    - *Kết nối Đa trường:* Ban đầu được thiết kế cho ảnh xám, nhưng có thể huấn luyện hoặc điều chỉnh để xử lý đầu vào nhiều kênh (ví dụ: ảnh RGB + kênh gradient depth, hoặc xử lý đám mây điểm trực tiếp bằng các mạng như PointNet++) để trích xuất đặc trưng 3D học được.
3. **Methods for 3D Point Cloud Feature Extraction:**
    - *Mô tả:* Trích xuất các đặc trưng từ dữ liệu đám mây điểm (LiDAR, camera độ sâu) mà không cần chiếu về 2D. Ví dụ các đặc trưng cục bộ tại mỗi điểm.
    - *Nền tảng Toán học:*
        - **Đại số Tuyến tính:** Phân tích lân cận điểm (**Principal Component Analysis - PCA**) để xác định mặt phẳng/đường cong cục bộ. Tính vector pháp tuyến (**normal vector**) tại mỗi điểm.

        - **Hình học 3D:** Tính khoảng cách giữa các điểm, góc giữa các pháp tuyến.
        - **Thống kê:** Phân tích phân phối điểm trong vùng lân cận. Ví dụ: **Fast Point Feature Histograms (FPFH)** tạo biểu đồ dựa trên quan hệ giữa các điểm và pháp tuyến lân cận (thuộc về Thống kê/Đại số).
    - *Độ phức tạp Thời gian:* Phụ thuộc vào kích thước đám mây điểm và kích thước vùng lân cận xét đến. Các phương pháp dựa trên lân cận có thể từ O(N log k) đến O(Nk) hoặc O(N log N) tùy thuộc vào cách tìm lân cận (KD-tree). Phương pháp dựa trên học sâu (PointNet++, KPConv) phụ thuộc kiến trúc mạng và yêu cầu GPU.
    - *Kết nối Đa trường:* ĐÂY là xử lý *trực tiếp* trên dữ liệu 3D, kết hợp liền mạch dữ liệu depth/LiDAR. Có thể kết hợp đặc trưng 3D này với đặc trưng 2D (từ ảnh màu) thông qua chiếu/ánh xạ.

- **Kết nối sang Giai đoạn 3:** Các đặc trưng (điểm đặc trưng, bộ mô tả, đặc trưng đám mây điểm) từ Giai đoạn 2 được sử dụng làm *đầu vào* cho các thuật toán Giai đoạn 3 như **khớp đặc trưng** (feature matching) để ước lượng chuyển động (để từ đó phát hiện vật thể mới, phân đoạn), hoặc sử dụng để xây dựng các biểu diễn 3D sơ khai, hoặc cung cấp thông tin cấu trúc cho các mạng phân đoạn/phát hiện.

- **Các Khái niệm/Thuật toán cần Kiểm tra & Nghiên cứu tập trung (Giai đoạn 2):**

    - **ORB:** Code, ứng dụng trong Open-source (ví dụ: ORB-SLAM). Hiểu cơ chế **FAST** và **BRIEF/rBRIEF**. Khớp **Hamming distance**.
    - **SuperPoint:** Code (PyTorch/TensorFlow), mô hình được huấn luyện trước. Cách dùng mô hình đã huấn luyện. **Hiểu kiến trúc CNN** cơ bản của nó.
    - **Đặc trưng 3D:** Cơ bản về **PCA** và tính **Normal Vector**. Ý tưởng của **FPFH**. Sử dụng **thư viện PCL** cho xử lý đám mây điểm và trích xuất đặc trưng 3D cơ bản.
    - **Độ phức tạp:** Nắm được loại thuật toán nào **Nhanh** (ORB) và loại nào cần **GPU** (SuperPoint, đặc trưng 3D học sâu).
    - **Toán:** Hiểu ý nghĩa của ma trận trong biến đổi và vector đặc trưng. Hiểu vai trò của Xác suất/Thống kê trong so sánh/khớp đặc trưng.

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

- **Mục đích:** Từ các đặc trưng thô và dữ liệu đa trường, tổ chức thông tin để nhận dạng, định vị các vật thể cụ thể, phân loại các vùng ảnh (segmentation), và xây dựng các biểu diễn 3D của môi trường. Giai đoạn này cung cấp các "thành phần" chính cho hệ thống suy luận cấp cao hơn.

- **Nền tảng Toán học Chủ chốt:**

    - **Đại số Tuyến tính (Linear Algebra):**
        - *Tầm quan trọng:* Các phép nhân ma trận trong **CNNs** (cốt lõi của các mô hình phát hiện/phân đoạn học sâu). Biểu diễn các **phép biến đổi hình học 3D** để kết hợp thông tin 2D và 3D. Phép chiếu (projection). Biểu diễn và xử lý **đám mây điểm**. Giải các hệ phương trình để tìm ra mô hình phù hợp (ví dụ: tìm mặt phẳng trong đám mây điểm với RANSAC).
    - **Giải Tích (Calculus):**
        - *Tầm quan trọng:* **Tính toán Gradient** cho **huấn luyện** các mô hình học sâu (**backpropagation**). Tối thiểu hóa các hàm mất mát cho **phát hiện, phân đoạn, ước**

lượng độ sâu, hoặc khớp mô hình 3D.

- **Tối ưu hóa (Optimization):**
  - *Tầm quan trọng:* Tìm các **tham số tốt nhất** cho các mô hình học máy. Giải quyết bài toán khớp 3D (ví dụ: **ICP** tìm phép biến đổi 3D tốt nhất giữa hai đám mây điểm). Huấn luyện mạng nơ-ron (Gradient Descent và các biến thể là thuật toán tối ưu).
- **Xác suất và Thống kê (Probability and Statistics):**
  - *Tầm quan trọng:* Các tầng cuối cùng của mạng phân loại/phân đoạn thường đưa ra **xác suất** thuộc về các lớp (**softmax**). Các mô hình nhận diện thường kèm theo **điểm tin cậy (confidence score)** (thống kê). Các kỹ thuật như **RANSAC** (tìm mô hình phù hợp bất chấp outlier) dựa trên **lấy mẫu ngẫu nhiên và xác suất**. Phân tích **thống kê** dữ liệu điểm (mean, covariance) trong xử lý đám mây điểm.
- **Hình học Đa Hình Chiếu & Hình học 3D (Multiple View Geometry & 3D Geometry):**
  - *Tầm quan trọng:* **Ước lượng độ sâu từ stereo** dựa trên **epipolar geometry**. Tái tạo cấu trúc 3D. Làm việc với các biểu diễn 3D (điểm, lưới - mesh, khối - voxel). Chuyển đổi giữa các hệ tọa độ camera, robot và thế giới. Nền tảng để hiểu cách các dữ liệu 2D và 3D liên quan đến nhau.
- **Lý thuyết Đồ thị (Graph Theory):**
  - *Tầm quan trọng:* Một số phương pháp phân đoạn ảnh cổ điển (**Graph Cuts**) hoặc phân đoạn trên đám mây điểm coi các pixel/điểm là các node và mối quan hệ lân cận/tương đồng là các edge. Có thể biểu diễn các mối quan hệ giữa các phần tử trong scene graph (dù Scene Graph Gen thường nằm giữa GĐ 3 và 4).

- **Hiệu suất Thời gian (Time Complexity) & Tích hợp Đa trường:** Giai đoạn này tính toán rất **nặng**, đặc biệt với các mô hình học sâu. **Hiệu suất thời gian thực** đòi hỏi sử dụng các kiến trúc mạng **nhẹ (lightweight)**, **tăng tốc phần cứng (GPU/NPU)** và các thuật toán **tối ưu** cho dữ liệu 3D. **Tích hợp đa trường** là chìa khóa để cung cấp thông tin toàn diện hơn, giúp các thuật toán ở đây hoạt động mạnh mẽ và chính xác hơn trong các điều kiện khó khăn (ví dụ: dùng độ sâu để hỗ trợ phân đoạn khi màu sắc không đủ, dùng LiDAR cho 3D mapping tầm xa).

- **Các Thuật toán Đáng chú ý:**

1. **Object Detection (Real-time Focus)**
   - *Mô tả:* Vẽ hộp giới hạn (bounding box) và phân loại các vật thể trong ảnh. Các mô hình SoTA hiện tại dựa trên **học sâu**.
   - *Thuật toán SoTA (Real-time):* **YOLO (You Only Look Only - các phiên bản V5, V7, V8 là SoTA về tốc độ), SSD (Single Shot Detector)**.
   - *Nền tảng Toán học:*
     - **CNNs:** Cốt lõi **Linear Algebra, Calculus, Optimization** để huấn luyện.
     - **Anchor Boxes/Prediction Heads:** Các kỹ thuật để dự đoán vị trí và kích thước hộp giới hạn, kèm theo xác suất lớp. Sử dụng nhiều lớp tích chập cuối để đưa ra dự đoán tại nhiều vị trí và tỷ lệ (scales) khác nhau. Kết quả dự đoán xác suất sử dụng **softmax** (**Probability**).
   - *Độ phức tạp Thời gian:* **Rất nhanh** cho các mô hình một giai đoạn (**one-stage** như YOLO, SSD), gần như **O(Input Image Size)**. Yêu cầu **GPU/NPU**. Tốc độ suy luận được tối ưu cao.
   - *Tích hợp Đa trường:* Các kiến trúc như **PointPillars, SECOND, CenterPoint** trực tiếp xử lý dữ liệu **LiDAR** hoặc kết hợp chiếu LiDAR/Depth với ảnh RGB để phát hiện vật thể 3D,

hoặc sử dụng mạng 2D huấn luyện trên dữ liệu RGB-D. Điều này cải thiện đáng kể khả năng phát hiện vật thể ở 3D và trong điều kiện thiếu sáng.

2. **Image Segmentation (Semantic and Instance) (Real-time Focus)**
    - *Mô tả:* Phân loại pixel (semantic seg) hoặc phát hiện và che phủ từng cá thể vật thể (instance seg). Quan trọng để robot hiểu đâu là vật thể thao tác, đâu là nền, đâu là vùng di chuyển được.
    - *Thuật toán SoTA (Real-time Semantic Seg):* **BiSeNet, ERFNet, MobileNetV3+Seg Head**.
    - *Thuật toán SoTA (Real-time Instance Seg):* **YOLACT, CenterMask** (thường chậm hơn Semantic Seg nhưng SoTA cải thiện tốc độ).
    - *Nền tảng Toán học:*
        - **CNNs / FCNs / U-Nets:** Cốt lõi là các khối **Linear Algebra, Calculus, Optimization**. Sử dụng kiến trúc mạng đặc biệt với các lớp giải chập (**deconvolution/transposed convolution**) hoặc upsampling để đưa về độ phân giải ảnh gốc và phân loại từng pixel (Semantic Seg: **Softmax** cho mỗi pixel - **Probability**). Instance Seg phức tạp hơn, kết hợp detection và mask generation.
    - *Độ phức tạp Thời gian:* Với các mô hình **lightweight**, có thể đạt **thời gian thực** trên **GPU/NPU**. Thường là O(Input Image Size), nhưng hằng số lớn hơn so với Detection đơn thuần do output chi tiết hơn (mask per-pixel).
    - *Tích hợp Đa trường:* Kết hợp RGB và Depth/Spectral làm kênh đầu vào cho mạng, hoặc dùng kiến trúc nhiều nhánh xử lý riêng từng modality và kết hợp feature maps. Dữ liệu độ sâu rất hữu ích để tách các vật thể gần nhau trong không gian 3D mà 2D khó phân biệt. Point cloud semantic segmentation (**PointNet++, RandLA-Net**) trực tiếp phân loại điểm 3D.
3. **3D Perception (Reconstruction, Registration)**
    - *Mô tả:* Xây dựng lại mô hình 3D của cảnh, hoặc căn chỉnh (align) các quét 3D lại với nhau.
    - *Thuật toán SoTA:*
        - **Stereo Matching: PSMNet, RAFT-Stereo** (dựa trên học sâu).
        - **Point Cloud Registration: ICP (Iterative Closest Point)** - Classic and still widely used, and its variants like **Generalized ICP (G-ICP)**. **PointNetLK, ICP based on learned features** (dựa trên học sâu).
    - *Nền tảng Toán học:*
        - **Stereo Matching: Epipolar Geometry (Hình học Đa hình chiếu)**. **Optimization** (ví dụ **Semi-Global Matching** giải bài toán năng lượng trên đồ thị hoặc học sâu huấn luyện với loss function). **Statistics/Correlation** để so sánh vùng ảnh.
        - **ICP:** Thuật toán lặp dựa trên **Bình phương nhỏ nhất (Least Squares)** để tìm phép biến đổi vật rắn (**Rigid Body Transformation - Đại số Tuyến tính, SE(3)**) giữa hai đám mây điểm sao cho khoảng cách trung bình giữa các điểm tương ứng là nhỏ nhất.
        - **Học sâu: CNNs, Optimization, Linear Algebra** như đã mô tả.
    - *Độ phức tạp Thời gian:*
        - **Học sâu Stereo/3D:** Yêu cầu **GPU/NPU**. Thời gian xử lý phụ thuộc kiến trúc và độ phân giải/kích thước dữ liệu.
        - **ICP:** Độ phức tạp cho một lần lặp có thể là O(N) (linear với số điểm), nhưng số lần lặp **không xác định trước** (tới khi hội tụ). Hiệu suất phụ thuộc vào số điểm, cách tìm correspondences và thuật toán tối ưu. Biến thể của ICP như Point-to-Plane ICP thường nhanh hơn. Cần cấu trúc dữ liệu **không gian** hiệu quả (KD-tree, Octree -

liên quan đến Thuật toán cây/Lý thuyết Đồ thị không trực tiếp nhưng ý tưởng tổ chức dữ liệu) để tìm điểm lân cận nhanh chóng.
    - *Tích hợp Đa trường:* Kết quả từ Object Detection/Segmentation 2D có thể giúp **giảm tập điểm** cho các thuật toán ICP (chỉ khớp những điểm thuộc vật thể nhất định). Dữ liệu RGB có thể tạo texture cho mô hình 3D. Dữ liệu từ nhiều nguồn 3D (stereo, LiDAR) cần được **đăng ký** (sử dụng các biến thể của ICP hoặc các phương pháp SLAM).

- **Kết nối sang Giai đoạn 4:** Đầu ra của Giai đoạn 3 (ví dụ: danh sách vật thể đã phát hiện kèm vị trí 3D, bản đồ môi trường 3D thưa/dày, mask phân đoạn các vật thể) là *đầu vào trực tiếp* cho các hệ thống **Theo dõi vật thể (Object Tracking)**, **Định vị (Localization)**, và **Lập bản đồ (Mapping)** trong Giai đoạn 4. Vị trí của vật thể trong không gian 3D cho phép robot tương tác với chúng. Bản đồ môi trường là cần thiết cho việc **lập kế hoạch đường đi (path planning)**.

- **Các Khái niệm/Thuật toán cần Kiểm tra & Nghiên cứu tập trung (Giai đoạn 3):**

    - **Học sâu cho Detection/Segmentation:** Lựa chọn kiến trúc **lightweight** (YOLO-series, MobileNet). Hiểu **CNNs**, hàm mất mát, huấn luyện. Tối ưu **suy luận** với TensorRT/OpenVINO.
    - **Ước lượng 3D/Xử lý Point Cloud:** Sử dụng **PCL** cho các tác vụ cơ bản (đọc/ghi, lọc, biến đổi). Hiểu nguyên lý của **ICP** và ứng dụng trong đăng ký đám mây điểm.
    - **Kết hợp dữ liệu:** Cách truyền và xử lý dữ liệu nhiều kênh trong mạng nơ-ron (RGB-D). Cách kết hợp thông tin 2D và 3D.
    - **Độ phức tạp:** Luôn nghĩ về hiệu quả trên **GPU/NPU**. So sánh tốc độ giữa các thuật toán SoTA cho cùng một nhiệm vụ.
    - **Toán: Nắm vững các phép biến đổi 3D (Linear Algebra, Geometric Algebra) là cực kỳ quan trọng.** Hiểu **Bình phương nhỏ nhất** trong ICP. Hiểu output **xác suất** từ các mạng học sâu.

Bộ Dữ liệu (Datasets) cho Kiểm thử (Giai đoạn 2 & 3 - Liên kết Multi-field và Robotics):

Các bộ dữ liệu tốt nhất cho việc phát triển quy trình này thường có:

1. **Dữ liệu Multi-sensor:** Ít nhất RGB-D (camera màu + độ sâu), lý tưởng là thêm IMU, LiDAR.
2. **Ground Truth:** Thông tin chính xác để đánh giá:
    - Tư thế robot (**ground truth poses**) hoặc quỹ đạo camera/robot.
    - Annotaion vật thể (bounding boxes, segmentation masks).
    - Mô hình 3D hoặc bản đồ chính xác của môi trường.
3. **Cảnh quan Robot thực tế:** Môi trường trong nhà/ngoài trời mà robot có thể hoạt động.

- **Các Bộ Dữ liệu Đáng chú ý:**
    - **TUM RGB-D Dataset:** Chứa video RGB-D và **ground truth tư thế** (dùng hệ thống Motion Capture) cho camera di chuyển trong các căn phòng. **Lý tưởng** cho phát triển và kiểm tra Visual Odometry, SLAM (Visual/RGB-D), ước lượng tư thế, và các thuật toán cần thông tin 3D ở tầm gần-trung. Rất tốt để kiểm tra **tốc độ và độ chính xác của các thuật toán Giai đoạn 2 và 3** trên dữ liệu RGB-D.
    - **KITTI Dataset:** Chứa dữ liệu từ xe tự lái (bao gồm **Camera stereo, LiDAR, IMU, GPS**, ground truth cho odometry/SLAM và detection 3D). **Lý tưởng** cho phát triển và kiểm tra thuật toán trên **dữ liệu ngoài trời quy mô lớn và đa cảm biến**, đặc biệt cho **LiDAR-based** và **Stereo-based** SLAM/odometry, **phát hiện vật thể 3D**. Độ phức tạp dữ liệu và môi trường lớn.
    - **EuRoC MAV Dataset:** Dữ liệu từ flycam (Micro Aerial Vehicle) bao gồm **Camera stereo và IMU chính xác cao**. **Lý tưởng** cho phát triển và kiểm tra **Visual-Inertial Odometry (VIO)** và **SLAM**

(rất quan trọng cho robot bay hoặc robot di chuyển nhanh cần độ trễ thấp).

- **ScanNet / MatterPort3D:** Các bộ dữ liệu **RGB-D trong nhà** quy mô lớn, có **annotations cho phân đoạn ngữ nghĩa, phân đoạn thể hiện, mô hình 3D voxel/mesh**. Tuy **không có ground truth tư thế tốc độ cao** như TUM/KITTI/EuRoC, chúng rất **lý tưởng** để phát triển và kiểm tra các thuật toán **phát hiện và phân đoạn vật thể (cả 2D và 3D)** trên dữ liệu trong nhà, xây dựng **bản đồ ngữ nghĩa 3D**.
- **COCO / Cityscapes:** Các bộ dữ liệu ảnh 2D lớn với annotations chi tiết cho **Object Detection và Segmentation**. Dù không có dữ liệu 3D trực tiếp, chúng rất hữu ích để **huấn luyện** các mô hình học sâu cho Giai đoạn 3 (detection, segmentation) mà sau đó có thể áp dụng hoặc fine-tune trên dữ liệu đa trường. Cityscapes đặc biệt có lợi cho robot di chuyển trên đường (ảnh ngoại cảnh, semantic segmentation của đường, vỉa hè, phương tiện).

Chọn bộ dữ liệu phù hợp giúp bạn tập trung vào đúng khía cạnh của pipeline và đánh giá hiệu quả của thuật toán trên dữ liệu gần giống với môi trường hoạt động của robot mục tiêu.
