ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH:

KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG TÁI TẠO VẬT THỂ 3D TỪ ẢNH 2D HỖ TRỢ TRONG THIẾT KẾ ĐỒ HỌA

Người hướng dẫn: **ThS. MAI VĂN HÀ**
Sinh viên thực hiện: **NGUYỄN HOÀNG QUÂN**
Số thẻ sinh viên: **102200281**
Lớp**:** **20TCLC_KHDL**

Đà Nẵng, 06/2025

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

CHUYÊN NGÀNH:

KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG TÁI TẠO VẬT THỂ 3D TỪ ẢNH 2D HỖ TRỢ TRONG THIẾT KẾ ĐỒ HỌA

Người hướng dẫn: **ThS. MAI VĂN HÀ**
Sinh viên thực hiện: **NGUYỄN HOÀNG QUÂN**
Số thẻ sinh viên: **102200281**
Lớp**:** **20TCLC_KHDL**

Đà Nẵng, 06/2025

TÓM TẮT

Tên đề tài: Xây dựng hệ thống tái tạo vật thể 3D từ ảnh 2D hỗ trợ trong thiết kế đồ họa

Sinh viên thực hiện: Nguyễn Hoàng Quân

Số thẻ SV: 102200281

Lớp: 20TCLC_KHDL

Sự phát triển nhanh chóng của thị giác máy tính và trí tuệ nhân tạo trong những năm gần đây đã mở ra nhiều hướng đi mới trong ngành công nghiệp thiết kế đồ họa, giải trí số và mô phỏng không gian ảo. Trong đó, tái tạo vật thể 3D từ ảnh 2D đang nổi lên như một giải pháp đầy tiềm năng nhằm đơn giản hóa quá trình dựng hình truyền thống, vốn đòi hỏi kỹ năng chuyên sâu và nhiều công sức thủ công. Với nhu cầu ngày càng tăng về nội dung 3D chất lượng cao trong các lĩnh vực như game, hoạt hình, kiến trúc và thực tế ảo, việc xây dựng các hệ thống hỗ trợ tái tạo 3D một cách tự động, nhanh chóng và chính xác đang trở thành một yêu cầu cấp thiết.

Tuy nhiên, tái tạo hình học 3D từ ảnh 2D là một bài toán phức tạp. Hệ thống cần phải suy diễn được thông tin chiều sâu, hình dạng và cấu trúc không gian từ hình ảnh phẳng – điều vốn dĩ khó khăn ngay cả đối với con người. Hơn nữa, việc cân bằng giữa tốc độ xử lý, độ chính xác của mô hình đầu ra và khả năng dễ sử dụng với người không chuyên cũng là những thách thức lớn trong việc triển khai thực tiễn.

Để giải quyết bài toán này, em xây dựng một hệ thống tái tạo vật thể 3D dựa trên phương pháp "image to point cloud" sử dụng mạng nơ-ron tích chập (CNN). Phương pháp này cho phép hệ thống học và suy luận hình dạng vật thể từ ảnh đầu vào, sau đó chuyển đổi thành dữ liệu point cloud. Tiếp theo, một bước hậu xử lý được thực hiện để tái cấu trúc point cloud thành mô hình mesh hoàn chỉnh, giúp cải thiện độ chính xác và chất lượng hình học của vật thể 3D.

Hệ thống được thiết kế hướng tới tính trực quan, dễ sử dụng và phù hợp với nhu cầu của người dùng trong ngành thiết kế đồ họa, góp phần thu hẹp khoảng cách giữa công nghệ 3D hiện đại và người dùng phổ thông, đồng thời mở rộng tiềm năng ứng dụng của AI trong sáng tạo nội dung số.

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Hoàng Quân

Số thẻ sinh viên: 102200281

Lớp: 20TCLC_KHDL

Khoa: Công nghệ Thông tin

Ngành: Khoa học dữ liệu và Trí tuệ nhân tạo

1. *Tên đề tài đồ án:* Xây dựng hệ thống tái tạo vật thể 3D từ ảnh 2D hỗ trợ trong thiết kế đồ họa
2. *Đề tài thuộc diện:* ☐ *Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện*
3. *Các số liệu và dữ liệu ban đầu:* Không có
4. *Nội dung các phần thuyết minh và tính toán:*

   Chương 1: Tổng quan đề tài – Trình bày các nội dung chính của đề tài

   Chương 2: Cơ sở lý thuyết – Trình bày một số lý thuyết quan trọng trong bài toán

   Chương 3: Giải pháp đề xuất – Trình bày quy trình và giải pháp cho dự án.

   Chương 4: Đánh giá kết quả – Đánh giá hiệu suất mô hình dự đoán và ứng dụng

   Kết luận: Tổng kết đóng góp, kết quả, giới hạn và hương hướng phát triển
5. *Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):* Không có
6. *Họ tên người hướng dẫn:* ThS. Mai Văn Hà
7. *Ngày giao nhiệm vụ đồ án:* *……/……/2025*
8. *Ngày hoàn thành đồ án:* *……/……/2025*

Đà Nẵng, ngày tháng năm 2025

Trưởng Bộ môn …….. **Người hướng dẫn**

LỜI NÓI ĐẦU

Trước tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại học Bách khoa – Đại học Đà Nẵng, đặc biệt là các thầy cô trong Khoa Công nghệ Thông tin, những người đã tận tình giảng dạy và truyền đạt kiến thức quý báu trong suốt quá trình học tập. Những kiến thức và kinh nghiệm mà em tích lũy được tại đây chính là nền tảng vững chắc giúp em thực hiện đồ án tốt nghiệp này.

Em xin bày tỏ lòng biết ơn sâu sắc đến Thạc sĩ Mai Văn Hà – giảng viên hướng dẫn, người đã luôn theo sát, hỗ trợ và định hướng em trong suốt quá trình thực hiện đề tài. Sự chỉ dẫn tận tâm cùng những góp ý quý giá của thầy đã giúp em vượt qua nhiều khó khăn và hoàn thiện đồ án theo đúng mục tiêu đề ra.

Em cũng xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn bên cạnh động viên, hỗ trợ em về cả tinh thần lẫn vật chất trong suốt thời gian học tập và thực hiện đồ án. Sự đồng hành của mọi người là nguồn động lực to lớn giúp em cố gắng không ngừng.

Mặc dù đã nỗ lực hoàn thành đề tài trong khả năng và thời gian cho phép, em hiểu rằng bài báo cáo vẫn có thể còn một số thiếu sót. Em rất mong nhận được những ý kiến đóng góp và phản hồi từ quý thầy cô và các bạn để hoàn thiện hơn trong tương lai.

Xin chân thành cảm ơn!

CAM ĐOAN

Em xin cam đoan rằng đồ án tốt nghiệp này là công trình nghiên cứu và thực hiện của chính em, dưới sự hướng dẫn của Thạc sĩ Mai Văn Hà.

Toàn bộ các tài liệu tham khảo được sử dụng trong báo cáo đều đã được trích dẫn rõ ràng, bao gồm tên tác giả, tên tài liệu và thông tin xuất bản đầy đủ.

Em hoàn toàn chịu trách nhiệm nếu có bất kỳ hành vi đạo văn hay vi phạm quy định học thuật nào được phát hiện trong báo cáo này.

Sinh viên thực hiện

Nguyễn Hoàng Quân

MỤC LỤC

DANH SÁCH CÁC BẢNG

DANH SÁCH CÁC HÌNH VẼ

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

CHỮ VIẾT TẮT:

Chữ viết tắt	Giải nghĩa
3D	Three-dimension – Ba chiều
2D	Two-dimension – Hai chiều
AI	Artificial Intelligence – Trí tuệ nhân tạo
AR	Augmented Reality
VR	Virtual Reality
RGB	Red Green Blue – Ba kênh màu cơ bản trong ảnh màu
CNN	Convolutional Neural Network – Mạng nơ-ron tích chập
GPU	Graphics Processing Unit – Bộ xử lý đồ họa

MỞ ĐẦU

Trong bối cảnh nội dung 3D ngày càng trở thành nhu cầu thiết yếu trong các lĩnh vực như thiết kế đồ họa, trò chơi điện tử, hoạt hình và thực tế ảo, khả năng dựng lại mô hình ba chiều từ một hoặc vài ảnh 2D đang thu hút sự quan tâm mạnh mẽ từ cả giới nghiên cứu lẫn ngành công nghiệp sáng tạo. Trên thực tế, nhiều ứng dụng hiện nay đã hỗ trợ tái tạo mô hình 3D từ ảnh 2D, bao gồm cả phần mềm miễn phí lẫn thương mại. Tuy nhiên, phần lớn các ứng dụng này vẫn dựa trên các kỹ thuật hình học truyền thống, đòi hỏi sự can thiệp thủ công đáng kể từ người dùng để đạt được kết quả mong muốn.

Việc ứng dụng các phương pháp học sâu mở ra cơ hội nâng cao tính tự động hóa, giảm thiểu thao tác thủ công và cho phép tạo ra các mô hình 3D có độ chính xác cao hơn. Từ định hướng đó, trong khuôn khổ đồ án tốt nghiệp này, tôi đề xuất và xây dựng một hệ thống tái tạo vật thể 3D từ ảnh 2D đơn, với kiến trúc pipeline được chia thành nhiều giai đoạn rõ ràng nhằm nâng cao khả năng kiểm soát và tối ưu hóa chất lượng đầu ra.

Cụ thể, hệ thống được thiết kế theo ba thành phần chính:

- Tiền xử lý và trích xuất đặc trưng: Xử lý ảnh đầu vào và trích xuất bản đồ độ sâu (depth map) thông qua mô hình MiDaS – một mô hình học sâu pretrained có khả năng trích độ sâu từ ảnh đơn một cách hiệu quả.
- Point Cloud Inference: Sử dụng kiến trúc CNN được dựa trên mô hình gốc Pixel2Point để dự đoán đám mây điểm (point cloud) của vật thể.
- Hậu xử lý dữ liệu đầu ra: Giai đoạn này bao gồm các bước như nội suy (upsampling) đám mây điểm, loại bỏ điểm nhiễu, tái tạo bề mặt và áp dụng các kỹ thuật làm mượt (smoothing) để hoàn thiện mô hình 3D. Đây là thành phần có vai trò quan trọng, ảnh hưởng trực tiếp đến độ chi tiết và tính thẩm mỹ của mô hình cuối cùng.

Báo cáo được triển khai như sau:

Chương 1: Tổng quan vấn đề – Chương này trình bày về vấn đề trong việc tái tạo vật thể 3D, những nghiên cứu liên quan, thách thức và mục tiêu nghiên cứu.

Chương 2: Cơ sở lý thuyết – Chương này trình bày về kiến trúc của mô hình dự đoán point cloud, những thuật toán và mô hình upsampling, tái tạo bề mặt vật thể, và các kỹ thuật làm mượt bề mặt vật thể.

Chương 3: Giải pháp đề xuất – Chương này mô tả chi tiết giải pháp và quy trình triển khai hệ thống, bao gồm chuẩn bị và xử lý dữ liệu, thiết kế và huấn luyện mô hình, triển khai mô hình, thiết lập cấu hình trên nền tảng cloud và xây dựng ứng dụng web.

Chương 4: Đánh giá kết quả – Chương này trình bày kết quả huấn luyện, đánh giá hiệu suất mô hình trên tập kiểm thử cùng với kết quả dự đoán, đánh giá hiệu năng hoạt động của hệ thống khi triển khai thực tế.

Kết luận và hướng phát triển – Phần này tổng kết các đóng góp chính của đề tài, tóm lược kết quả đạt được, chỉ ra các giới hạn còn tồn tại, đồng thời đề xuất một số hướng nghiên cứu và cải tiến trong tương lai nhằm nâng cao hiệu quả và khả năng ứng dụng thực tế của hệ thống.

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1. Khái niệm về tái tạo vật thể 3D

Theo Wikipedia, trong thị giác máy tính và đồ họa máy tính, **tái tạo vật thể 3D** là quy trình tái tạo lại hình dạng và ngoại hình của vật thể thật. Quá trình này được thực hiện bằng cách tạo ra mô hình ba chiều hoàn chỉnh từ các dữ liệu đầu vào còn thiếu như ảnh 2D, ảnh RGB-D hoặc đám mây điểm (point cloud).

Trong quá trình tái tạo, hệ thống phải suy luận và khôi phục các thông tin không gian quan trọng như vị trí, kích thước, hình dạng, cấu trúc bề mặt, và trong nhiều trường hợp còn bao gồm cả màu sắc hoặc đặc trưng vật lý khác. Kết quả của quá trình tái tạo thường là các mô hình 3D dưới dạng mesh, point cloud, hoặc là voxel – tùy thuộc và mục tiêu sử dụng và định dạng dữ liệu đầu ra mong muốn.

Một trong những thách thức lớn nhất của bài toán này nằm ở tính chất không đầy đủ và mơ hồ của dữ liệu đầu vào. Một hình ảnh 2D có thể tương ứng với vô số hình học 3D khả thi, do vậy việc tái tạo đòi hỏi phải áp dụng những kiến thức tiên nghiệm hoặc các kỹ thuật học máy/học sâu hiện đại để lựa chọn mô hình 3D phù hợp nhất với dữ liệu quan sát. Ngoài ra dữ liệu đầu vào cũng có thể bị nhiễu, thiếu điểm, hoặc sai lệch do hạn chế của thiết bị thu thập và điều kiện môi trường, khiến việc xử lý càng trở nên phức tạp.

Tái tạo vật thể 3D không chỉ là một bài toán kỹ thuật mà còn là nền tảng quan trọng cho nhiều ứng dụng thực tiễn như thiết kế kỹ thuật, in 3D, bảo tồn di sản văn hóa, thực tế ảo và tăng cường (VR/AR), robot, xe tự lái, bản đồ 3D, và nhiều lĩnh vực nghiên cứu khoa học khác. Đặc biệt trong kỷ nguyên số, việc xây dựng các mô hình 3D chính xác từ ảnh chụp đã trở thành một bước đi quan trọng để số hóa thế giới vật lý và phục vụ các hệ thống thông minh như mô hình song sinh số (digital twins). Trong khuôn khổ đề tài này, việc tái tạo 3D được tập trung vào bài toán sinh mô hình đám mây điểm từ ảnh RGB-D. Đây là một bước nền quan trọng, cho phép máy tính không chỉ nhận diện đối tượng qua ảnh tĩnh, mà còn hiểu được cấu trúc hình học không gian của vật thể - từ đó mở đường cho các ứng dụng nâng cao như hoàn thiện mô hình vật thể, nâng độ chi tiết (point cloud upsampling), hay dựng mesh (mesh reconstruction).

1.2. Tổng quan về những ứng dụng liên quan hiện nay

Trong thực tiễn, đã có nhiều công cụ phần mềm và dịch vụ hỗ trợ việc chuyển ảnh 2D thành các mô hình 3D – từ giải pháp trực tuyến đơn giản đến các phần mềm chuyên nghiệp với độ phức tạp và chi phí khác nhau.

Các giải pháp trực tuyến (web-based) gồm các công cụ phổ biến như:

- **Hyper3D Rodin** là một nền tảng AI miễn phí chuyên tạo mô hình 3D chất lượng cao từ ảnh hoặc mô tả văn bản (text prompt). Công cụ này ngoài chức năng chuyển đổi image-to-mesh thì còn tích hợp các tính năng nâng cao như AI Texture Generation – tự động tạo bảo đồ texture và vật liệu được tối ưu cho đồ họa kỹ thuật số chất lượng cao, AI Remesh & Re-Texture: tối ưu hóa topology, làm mượt bề mặt và tái cấu trúc lưới để sẵn sàng sử dụng trong game, VR hoặc các visual pipeline. Ngoài ra nền tảng cũng hỗ trợ những tính năng quan trọng khác như Text-to-3D & Image-to-3D, hỗ trợ xuất file đa định dạng như OBJ, FBX, GLB, STL…
- **Meshi, Tripo AI** là các nền tảng AI tạo mô hình 3D từ ảnh đơn hoặc nhiều ảnh đa góc nhìn; cho kết quả rất tốt nhưng thường bị giới hạn về số lần sử dụng miễn phí.

Đối với nền tảng di động thì có các ứng dụng phổ biển như:

- **KIRI Engine** là ứng dụng cho iOS/Android dùng kỹ thuật photogrammetry (chuỗi ảnh chụp đa góc nhìn), quét đa góc nhìn để tạo mô hình 3D (OBJ/FBX/STL) nhanh chóng và miễn phí giới hạn theo tuần.
- **Qlone** – ứng dụng photogrammetry trên smartphone, hỗ trợ quét 3D dễ dùng, phù hợp trong ngành in hoặc model học.

Đối với các phần mềm đồ họa truyền thống, nhiều công cụ mạnh mẽ đã hỗ trợ người dùng dựng vật thể 3D từ ảnh, tuy nhiên phần lớn vẫn dựa vào thao tác thủ công với các plugin hoặc tính năng phụ trợ chứ chưa đạt đến mức tự động hóa toàn bộ quy trình:

- **Blender** là phần mềm đồ họa 3D mã nguồn mở mạnh mẽ, được sử dụng để làm phim hoạt hình, kỹ xảo, ảnh nghệ thuật, mẫu in 3D, phần mềm tương tác 3D và video game. Hiện nay Blender hỗ trợ dựng hình thể 3D từ ảnh thông qua các add-on như Photogrammetry Importer – add-on cho phép nhập dữ liệu từ các công cụ SfM/MVS như COLMAP, Meshroom, VisualSFM, rồi trực tiếp tái tạo point cloud, mesh và camera poses thông qua Geometry Nodes hoặc OpenGL. Các add-on như Image-to-Mesh (I2M) và Sketch N'Trace được phát triển cho phép chuyển ảnh 2D thành mô hình mesh hoặc curve sơ khai; hỗ trợ batch processing, UV projection, rất phù hợp cho concept art hoặc FX. Ngoài ra add-on như Motion-tracking, 3D Marker to Mesh theo dõi các điểm đặc trưng từ nhiều khung hình, Blender có thể tái tạo point cloud rồi

chuyển thành lưới 3D, từ đó người dùng có thể remesh và gắn texture thủ công.

- **Adobe Substance 3D** là bộ công cụ mạnh mẽ hỗ trợ quy trình tạo và hoàn thiện mô hình 3D từ ảnh thực tế hoặc dữ liệu quét. Substance 3D Sampler cho phép tái tạo point cloud, mesh và texture chất lượng cao từ ảnh chụp đa góc nhìn bằng AI, giúp tối ưu mô hình trước khi xuất các phần mềm khác như Substance Painter hay Blender. Substance 3D Modeler cung cấp môi trường điêu khắc 3D linh hoạt, hỗ trợ sculpting chi tiết trên cả VR và desktop. Substance 3D Stager giúp dựng cảnh và render photorealistic, tích hợp AI tạo nền từ text prompt và tự động điều chỉnh ánh sáng, phối cảnh, hỗ trợ animation quay 360 độ. Hệ sinh thái plugin của Adobe Substance 3D tích hợp sâu với các phần mềm dựng hình và game engine như Unreal Engine, giúp áp dụng và tùy chỉnh vật thể nhanh chóng, nâng cao hiệu quả quy trình làm việc.

Mặc dù các phần mềm đồ họa truyền thống rất mạnh mẽ, chúng vẫn đòi hỏi người dùng có kỹ năng chuyên sâu và thực hiện nhiều thao tác thủ công trong quá trình dựng hình. Quy trình chuyển đổi từ ảnh sang mô hình 3D thường không hoàn toàn tự động, đặc biệt khi cần tinh chỉnh mesh, UV, hoặc texture. Ngược lại, các nền tảng ứng dụng AI như Hyper3D Rodin, Meshi, Tripo AI,... mang lại quy trình gần như tự động hoàn toàn – từ phân tích ảnh, tái tạo hình khối, đến sinh texture và tối ưu mô hình. Chúng hỗ trợ xuất đa định dạng, sẵn sàng cho các pipeline đồ họa hoặc game engine. Chính khả năng tự động hóa cao và thân thiện với người dùng không chuyên của những nền tảng này là nguồn cảm hứng quan trọng cho hệ thống mà tôi xây dựng trong đồ án này.

1.3. Tổng quan về hệ thống tái tạo vật thể 3D từ ảnh 2D

Trong khuôn khổ đề tài này, tôi đã nghiên cứu xây dựng một hệ thống có khả năng tái tạo vật thể 3D dưới dạng biểu diễn point cloud và mesh từ ảnh RGB đầu vào. Hệ thống bao gồm các thành phần chính: pipeline xử lý dữ liệu, mô hình học sâu dự đoán point cloud, bước hậu xử lý tạo mesh, cùng giao diện web trực quan giúp người dùng tải ảnh lên và trực quan hóa kết quả tái tạo 3D. Mô hình được triển khai trên hạ tầng cloud (Google Cloud Run) để thực hiện inference và quản lý model hiệu quả.

1.3.1. Mục tiêu đề tài

Đề tài hướng đến việc xây dựng một hệ thống tái tạo vật thể 3D hoàn chỉnh, với mục tiêu cụ thể như sau:

- *Tái hiện hình học 3D từ ảnh 2D*: Phát triển một pipeline hoàn chỉnh có khả năng suy luận cấu trúc không gian của vật thể từ ảnh chụp 2D, thông qua mô

hình học sâu, và biểu diễn kết quả dưới hai dạng phổ biến là point cloud và mesh.

- *Thiết kế mô hình học sâu hiệu quả*: Áp dụng và tinh chỉnh mô hình dự đoán point cloud (Pixel2Point) để tối ưu khả năng học đặc trưng hình học từ ảnh RGB-D. Đảm bảo chất lượng tái tạo cao, đặc biệt với các vật thể có hình dạng phức tạp.
- *Xây dựng giao diện ứng dụng trực quan*: Phát triển một hệ thống web cho phép người dùng tải ảnh lên, khởi chạy quá trình tái tạo và xem trực tiếp kết quả 3D. Việc trực quan hóa kết quả dưới dạng tương tác 3D giúp tăng khả năng kiểm chứng đầu ra của mô hình.
- *Tích hợp công nghệ điện toán đám mây*: Sử dụng nền tảng Google Cloud để triển khai mô hình inference trên Cloud Run, lưu trữ và quản lý mô hình học sâu bằng Artifact Registry, đồng thời đảm bảo khả năng mở rộng, dễ cập nhật và tích hợp vào các hệ thống lớn hơn trong tương lai.
- *Đảm bảo tính khả thi và ứng dụng thực tế*: Hướng đến việc tạo ra một hệ thống có thể hoạt động với dữ liệu đầu vào từ môi trường thực, qua đó mở ra tiềm năng ứng dụng trong các lĩnh vực như thiết kế đồ họa, mô phỏng kỹ thuật, số hóa sản phẩm và hỗ trợ sáng tạo nội dung 3D.

1.3.2. Công nghệ sử dụng

Trong quá trình phát triển hệ thống, tôi đã sử dụng các công cụ và công nghệ chính sau:

- Ngôn ngữ lập trình: **Python** cho pipeline huấn luyện mô hình và backend, Javascript cho ứng dụng web frontend.
- Môi trường huấn luyện: **Google Colab** để tận dụng tài nguyên GPU, **Google Drive** để lưu trữ dữ liệu.
- Framework và thư viện: **PyTorch** cho xây dựng kiến trúc và huấn luyện mô hình học sâu. Các thư viện bao gồm **OpenCV, NumPy và torchvision** cho xử lý ảnh và dữ liệu số, **Matplotlib và Plotly** để trực quan hóa dữ liệu và kết quả huấn luyện.
- Hạ tầng triển khai: **Docker** để đóng gói mô hình và triển khai trên **Google Cloud Run**, sử dụng **Artifact Registry** để quản lý phiên bản mô hình.
- Frontend: Kế thừa giao diện web từ Three.js Editor.
- Backend API: sử dụng **FastAPI** để xây dựng các endpoint inference và kết nối giữa frontend và mô hình triển khai trên cloud.

1.3.3. Thách thức khi thực hiện

Mặc dù hệ thống đã đạt được những kết quả nhất định, quá trình thực hiện vẫn gặp nhiều thách thức kỹ thuật và một số hạn chế chưa thể khắc phục triệt để.

- **Chất lượng dữ liệu đầu vào**: Việc suy luận hình học 3D từ ảnh RGB đơn thuần là một bài toán bất định, đặc biệt khi ảnh đầu vào thiếu thông tin về chiều sâu, góc chụp bị khuất, hoặc vật thể có hình dạng phức tạp. Điều này ảnh hưởng lớn đến độ chính xác của point cloud được tạo ra.
- **Hiệu năng và độ chính xác của mô hình**: Mặc dù mô hình cho kết quả tốt trên dữ liệu tổng hợp, việc tái tạo các chi tiết nhỏ hoặc phần khuất vẫn còn hạn chế.
- **Hậu xử lý hình học**: Việc chuyển từ point cloud sang mesh đòi hỏi các bước xử lý hình học như surface reconstruction (tái tạo bề mặt) và mesh smoothing. Đây là những bước phức tạp, dễ sinh lỗi hình học như mesh không khép kín, lỗ hổng hoặc mặt tam giác lệch.
- **Tối ưu hóa triển khai trên cloud**: Việc đóng gói mô hình và triển khai inference trên hạ tầng cloud như Google Cloud Run gặp khó khăn ban đầu liên quan đến giới hạn tài nguyên, thời gian phản hồi, và quản lý môi trường phụ thuộc.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

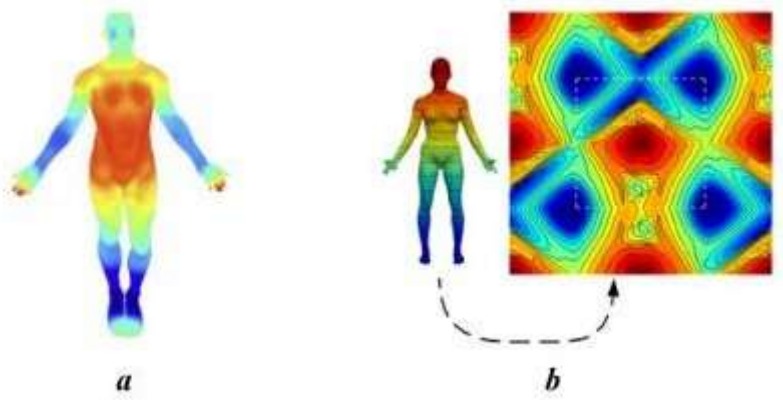
2.1. Các hình thức biểu diễn dữ liệu 3D

Trong lĩnh vực thị giác máy tính và đồ họa máy tính, dữ liệu 3D có thể được biểu diễn dưới nhiều hình thức khác nhau, tùy thuộc vào mục đích sử dụng và đặc điểm kỹ thuật của từng ứng dụng. Các hình thức biểu diễn này không chỉ khác nhau về cách lưu trữ và xử lý thông tin mà còn có thể được phân loại dựa trên cấu trúc hình học thành hai nhóm chính: dữ liệu Euclidean và dữ liệu phi Euclidean.

2.1.1. Dữ liệu Euclidean

Dữ liệu Euclidean là dữ liệu có cấu trúc lưới đều đặn, tuân theo hình học Euclidean, cho phép áp dụng trực tiếp các phép toán học truyền thống như tích chập (convolution). Những dạng dữ liệu này phù hợp với các mô hình học sâu tiêu chuẩn do tính chất tổ chức đều và dễ xử lý. Những dạng biểu diễn dữ liệu 3D chính thuộc dữ liệu Euclidean bao gồm: descriptors, projections, dữ liệu RGB-D, volumetric data, multi-view data. [1]

- **Descriptors** trong không gian 3 chiều là một dạng biểu diễn đặc trưng được trích xuất từ hình dạng, bề mặt, hoặc vùng lân cận của một điểm trong dữ liệu 3D. Loại dữ liệu này nhằm mô tả các đặc điểm hình học hoặc topo học của nó. Descriptor 3D cung cấp một cách biểu diễn dữ liệu gọn nhẹ nhưng vẫn giữ được các đặc tính quan trọng của đối tượng. Chúng thường được kết hợp với các mô hình học máy/học sâu để trích xuất thêm các đặc trưng phân biệt có đặc tính phân cấp cao hơn, giúp mô hình hiểu hình dạng tốt hơn.
- **Projections** là một phương pháp biểu diễn dữ liệu 3D bằng cách chiếu đối tượng 3 chiều sang không gian 2 chiều. Quá trình chiếu này giữ lại một số đặc trưng quan trọng của hình dạng gốc, tùy thuộc vào loại phép chiếu được sử dụng. Phép chiếu giúp chuyển đổi đối tượng 3D thành một lưới 2D có cấu trúc Euclidean, cho phép áp dụng trực tiếp các mô hình học sâu truyền thống vốn đã được nghiên cứu kỹ lưỡng cho ảnh 2D. Tuy nhiên, hạn chế lớn của phương pháp này là mất mát thông tin trong quá trình chiếu, khiến chúng chưa phù hợp cho các bài toán yêu cầu độ chính xác cao, như nhận diện điểm tương ứng dày đặc (dense correspondence) trong thị giác máy tính 3D.

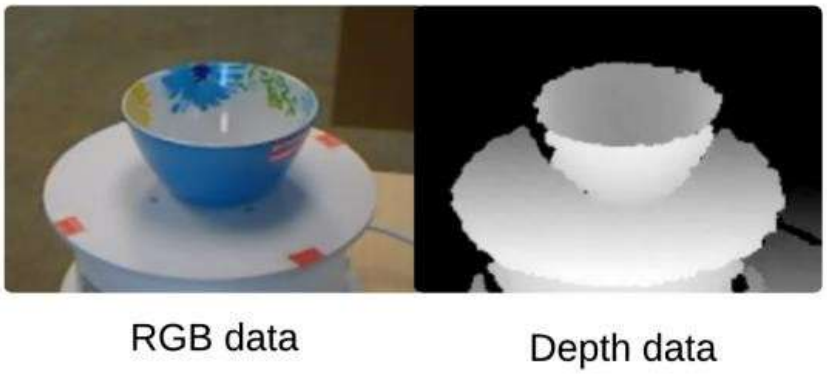


Hình 2.1: (a) Dữ liệu descriptors. (b) Dữ liệu projections

- **Dữ liệu RGB-D** là một dạng biểu diễn 3D khá phổ biến trong những năm gần đây, nhờ vào sự phát triển và phổ biến của các cảm biến RGB-D như Microsoft Kinect. Dữ liệu RGB-D cung cấp thông tin "2.5D" về đối tượng 3D, bằng cách kết hợp hình ảnh màu (RGB) với bản đồ độ sâu (Depth map – D). Loại dữ liệu này dễ thu thập, đồng thời lại hiệu quả trong việc mô tả hình dạng không gian của các đối tượng, nên thường được sử dụng trong nhiều bài toán như nhận dạng danh tính (identity recognition), ước lượng tư thế (pose regression) và tìm điểm tương ứng (correspondence).

  Một lợi thế lớn của dữ liệu RGB-D là sự phong phú về dữ liệu, số lượng tập dữ liệu RGB-D hiện có vượt trội hơn so với các loại dữ liệu 3D khác như point cloud hay mesh, góp phần hỗ trợ mạnh mẽ cho việc huấn luyện và đánh giá các mô hình học sâu trên không gian 3D.

  Trong hệ thống tôi xây dựng cho đồ án này, RGB-D được sử dụng làm đầu vào cho mô hình dự đoán point cloud thay vì ảnh RGB thông thường, nhằm giữ lại những thông tin không gian quan trọng, từ đó giúp mô hình tái tạo hình dạng vật thể 3D một cách chính xác hơn.



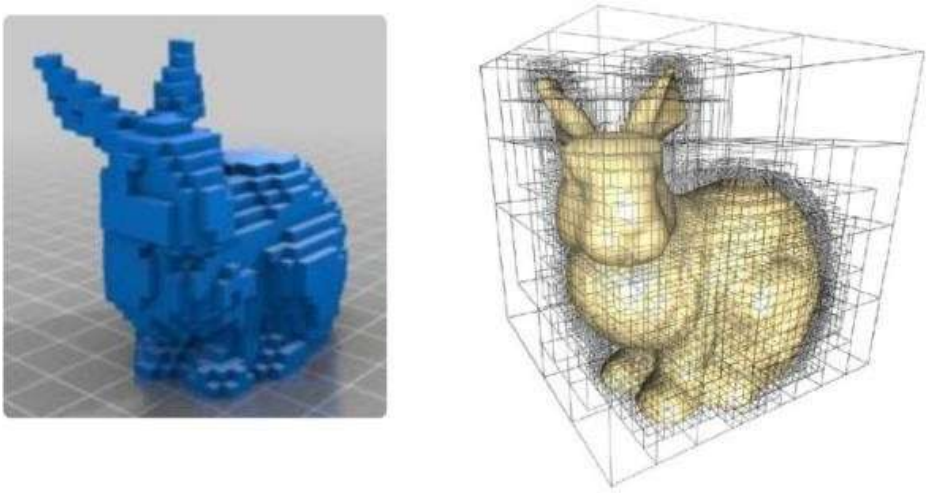
Hình 2.2: Dữ liệu ảnh RGB và bản đồ độ sâu của nó

- **Volumetric data** là một phương pháp biểu diễn bằng cách ánh xạ đối tượng vào một lưới đều trong không gian 3 chiều, trong đó các phần tử nhỏ nhất được gọi là voxel. Mỗi voxel mô tả sự phân bố của đối tượng trong không gian, có thể mang thông tin về hình dạng cũng như góc nhìn bằng cách phân loại chúng thành các voxel nhìn thấy, bị che khuất (occluded) hoặc tự che khuất (self-occluded).

  Biểu diễn voxel-based có ưu điểm về tính đơn giản và khả năng mã hóa hình dạng 3D một cách rõ ràng. Tuy nhiên, nó cũng tồn tại nhiều hạn chế, đặc biệt là hiệu quả về lưu trữ. Phương pháp này lưu trữ cả các phần không gian bị chiếm dụng lẫn không bị chiếm dụng, dẫn đến dung lượng bộ nhớ tăng cao, gây khó khăn khi xử lý các đối tượng có độ phân giải cao.

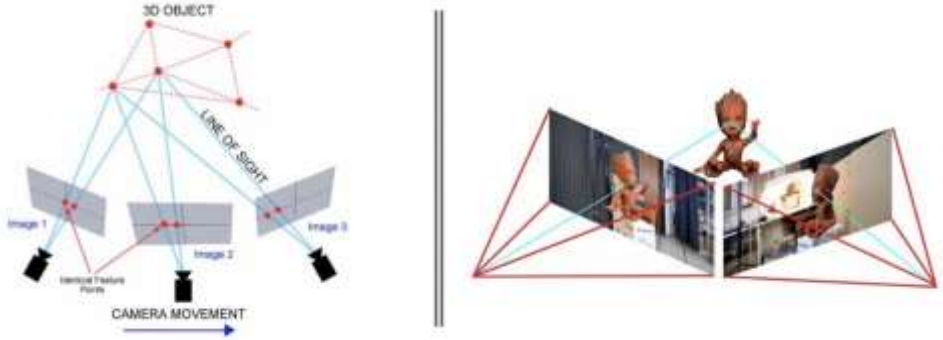
  Một hướng cải tiến là sử dụng biểu diễn octree-based, trong đó không gian 3D được phân chia theo cấu trúc phân cấp, tương tự như quadtree trong không gian 2D. Thay vì các voxel có kích thước cố định, octree sử dụng các voxel có kích thước thay đổi, giúp đại diện các vùng đồng nhất bằng các khối lớn hơn, từ đó tiết kiệm bộ nhớ và giảm chi phí tính toán.

  Tuy vậy, cả voxel-based và octree-based đều không bảo toàn được đầy đủ hình học nội tại và tính mượt của bề mặt đối tượng, nên chưa phải là lựa chọn tối ưu cho các tác vụ đòi hỏi độ chính xác hình học cao.



Hình 2.3: Dữ liệu volumetric gồm voxels (trái) và octree (phải)

- **Multi-view data** là một phương pháp biểu diễn đối tượng 3D thông qua tập hợp các hình ảnh 2D được chụp từ nhiều góc nhìn khác nhau. Cách tiếp cận này cho phép mô hình học được nhiều tập đặc trưng từ các quan sát khác nhau, từ đó giảm thiểu ảnh hưởng của nhiễu, thiếu dữ liệu, hiện tượng che khuất và thay đổi ánh sáng trong quá trình thu nhận dữ liệu. Tuy nhiên, một thách thức lớn của phương pháp này là xác định số lượng góc nhìn tối ưu. Số lượng quá ít có thể không đủ để bao quát đầy đủ các đặc trưng của đối tượng, dẫn đến hiện tượng overfitting. Ngược lại, sử dụng quá nhiều góc nhìn sẽ làm tăng chi phí tính toán không cần thiết.



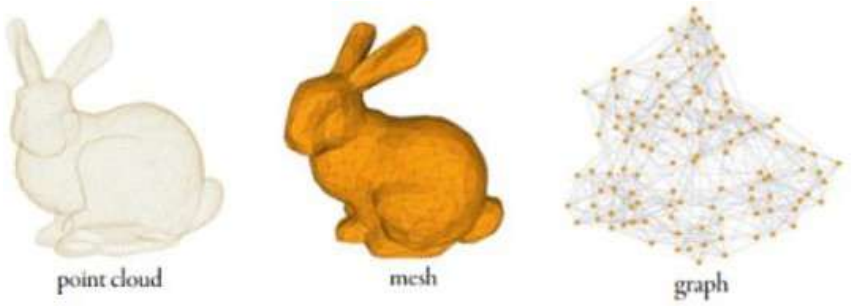
Hình 2.4: Dữ liệu đa góc nhìn của một vật thể

Mặc dù tồn tại những hạn chế như vậy, dữ liệu đa góc nhìn (multi-view) vẫn chứng minh vẫn được chứng minh là hiệu quả hơn so với dữ liệu thể tích (volumetric) trong nhiều tác vụ xử lý và nhận dạng đối tượng 3D. Cả hai loại dữ liệu multi-view và volumetric đều tỏ ra phù hợp hơn với các đối tượng có hình dạng cứng (rigid) và ít biến dạng, chẳng hạn như các mô hình CAD trong tập dữ liệu ModelNet (bao gồm các đối tượng như bàn, ghế, tủ…). Ngược lại,

với các đối tượng biến dạng mạnh như dữ liệu biểu cảm khuôn mặt trong bộ BU4DFE, các phương pháp này thường không đạt hiệu quả cao.

2.1.2. Dữ liệu phi Euclidean

Loại biểu diễn thứ hai trong biểu diễn dữ liệu 3D là dữ liệu phi Euclidean. Loại dữ liệu này không có hệ tọa đồ toàn cục hoặc cấu trúc không gian vector chung, khiến cho việc xử lý trở nên phức tạp hơn. Các dạng dữ liệu tiêu biểu thuộc loại này gồm point cloud, mesh và đồ thị (graph). Những cấu trúc này thường có đặc điểm hình học phức tạp như độ cong vô hạn, khả năng tự giao nhau và thay đổi chiều không gian tùy theo vị trí hoặc mức độ phân tích. Mặc dù một số dạng dữ liệu như point cloud và mesh có thể được coi là Euclidean ở mức độ cục bộ, nhưng trên toàn cục, chúng vẫn mang tính phi Euclidean. Do đó, chúng thường được xử lý ở quy mô toàn bộ để phục vụ cho các tác vụ phức tạp như nhận dạng (recognition) và tìm tương ứng (correspondence), hoặc tái tạo hình học (3D reconstruction).



Hình 2.5: Ba loại dữ liệu phi Euclidean phổ biến nhất

- **Mesh** (lưới tam giác) là một trong những dạng biểu diễn phổ biến nhất cho hình dạng 3D, được cấu tạo từ các đa giác (faces) tạo thành bởi tập các đỉnh (vertices) có thông tin kết nối với nhau trong không gian 3D. Ở mức cục bộ, mesh có thể được coi là dữ liệu Euclidean do thiếu các đặc tính như bất biến tịnh tiến, phép toán vector và hệ tọa độ toàn cục.

  Việc xử lý mesh bằng các phương pháp học sâu gặp nhiều khó khăn do dữ liệu có cấu trúc không đều, dễ bị nhiễu, thiếu dữ liệu và khác biệt về độ phân giải. Một hướng tiếp cận hiệu quả là biểu diễn mesh dưới dạng đồ thị (graph), trong đó các đỉnh của đồ thị tương ứng với các đỉnh trong mesh và các cạnh biểu diễn quan hệ kết nối. Dạng biểu diễn này cho phép ứng dụng các kỹ thuật học sâu trên đồ thị (Graph Neural Networks – GNNs), đặc biệt thông qua phân tích phổ đồ thị (spectral analysis) như sử dụng phân rã trị riêng của ma trận Laplacian để định nghĩa các phép tích chập tương tự như trên ảnh. Cách

tiếp cận này đã mở ra nhiều hướng nghiên cứu mới đầy tiềm năng trong việc xử lý dữ liệu hình học.

- **3D point cloud** là tập hợp các điểm 3D rời rạc trong không gian, dùng để mô tả hình học của bề mặt đối tượng. Mặc dù ở quy mô cục bộ có thể coi là dữ liệu Euclidean, nhưng ở quy mô toàn cục, point cloud được xem là dữ liệu phi Euclidean do thiếu thông tin về kết nối hình học giữa các điểm. Dữ liệu point cloud có thể thu thập dễ dàng từ các thiết bị như LiDAR, Kinect, hoặc camera độ sâu hiện đại, và ngày càng được sử dụng rộng rãi trong nhiều lĩnh vực như: tự lái, robotics, bản đồ 3D, thực tế ảo (AR/VR), và đặc biệt là trong các tác vụ như nhận dạng, tái tạo 3D, và dự đoán hình dạng.

  Tuy nhiên việc xử lý chúng gặp nhiều thách thức do không có cấu trúc lưới cố định (khác với mesh) gây khó khăn trong việc xác định bề mặt. Một thách thức nữa là việc xử lý bị ảnh hưởng bởi môi trường thu thập như nhiễu, thiếu dữ liệu, mật độ điểm không đồng đều hoặc có lỗ hổng do giới hạn của cảm biến và điều kiện môi trường.

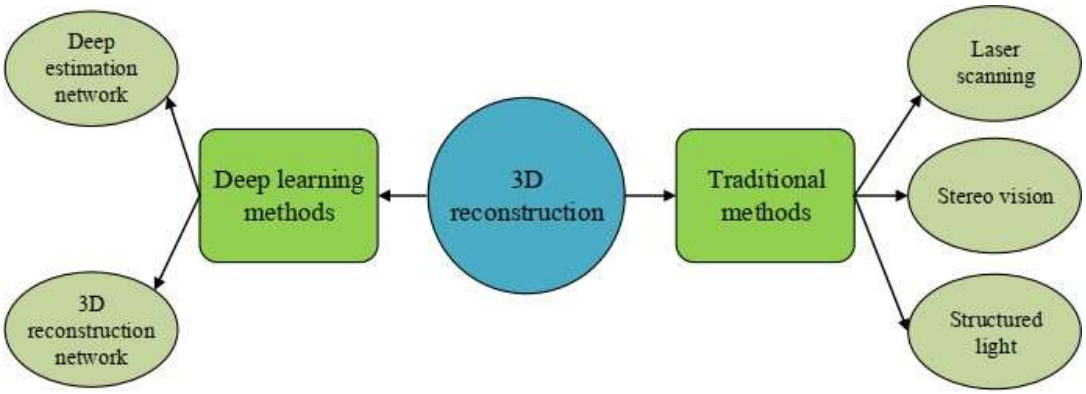
  Gần đây, với sự phát triển của học sâu, nhiều mô hình tiên tiến như PointNet/PointNet++, DGCNN, PointNeXt (CVPR 2023), Pixel2Point,… đã được đề xuất để khai thác hiệu quả đặc trưng từ dữ liệu point cloud mà không cần chuyển đổi sang dạng mesh hoặc voxel.

Trong bối cảnh các phương pháp tái tạo hình học 3D từ ảnh 2D ngày càng phát triển, việc lựa chọn dạng dữ liệu đầu ra phù hợp đóng vai trò quan trọng trong hiệu quả huấn luyện và chất lượng kết quả. Trong số các dạng biểu diễn hình học phổ biến như voxel, mesh hay point cloud, trong đồ án này dữ liệu point cloud được lựa chọn làm đầu ra cho mô hình dự đoán với những lý do sau:

- Point cloud không yêu cầu cấu trúc kết nối ổn định như mesh, cũng không đòi hỏi bộ nhớ lớn như voxel. Điều này giúp mô hình dễ dàng học được đặc trưng hình học của đối tượng mà không bị ràng buộc bởi các yếu tố hình học phức tạp như liên kết cạnh hay độ phân giải lưới.
- Point cloud nhẹ về mặt tính toán và bộ nhớ, phù hợp với các mô hình học sâu khi xử lý dữ liệu 3D ở quy mô lớn hoặc thời gian thực.
- Với sự phát triển của các mô hình như PointNet, DGCNN, PointNeXt, Pixel2Point…, dữ liệu point cloud cho phép trích xuất đặc trưng hiệu quả và học ánh xạ trực tiếp từ ảnh 2D sang không gian 3D mà không cần bước trung gian chuyển đổi sang mesh hay voxel.

2.1.3. Phân loại các phương pháp trong tái tạo 3D

Theo M. Jia và M. Zhang [2], các phương pháp tái tạo 3D hiện này có thể chia thành hai nhóm chính: phương pháp truyền thống và phương pháp dựa trên học sâu. Phương pháp truyền thống thường có độ chính xác cao, kỹ thuật đã được hoàn thiện và kiểm chứng qua thời gian. Trong khi đó, các phương pháp học sâu nổi bật với tốc độ xử lý nhanh, khả năng hoạt động thời gian thực tốt, và ít phụ thuộc vào phần cứng chuyên dung.



Hình 2.6: Phân loại phương pháp trong bài toán tái tạo vật thể 3D

a) Phương pháp hình học truyền thống

Các phương pháp truyền thống chủ yếu dựa trên việc phân tích trực tiếp các manh mối hình học có trong ảnh, sau đó sử dụng các thuật toán đặc thù để khôi phục thông tin không gian ba chiều như độ sâu và hình dạng bề mặt.

Một trong những nghiên cứu sớm nhất được thực hiện bởi Robert et al. [3], ông phân tích khả năng thu được thông tin ba chiều từ ảnh hai chiều thông qua các phương pháp thị giác máy tính. Horn et al. [4] đã giới thiệu phương pháp tái tạo độ sâu từ bóng đổ (Shape From Shading – SFS). Còn Kiyasu et al. [5] từ Đại học Tokyo thì sử dụng hình ảnh phản chiếu ánh sáng trên vật thể để khôi phục hình dạng bề mặt 3D.

Sự phát triển tiếp theo là sự ra đời của phương pháp Structure From Motion (SfM) do Snavely et al. [6] đề xuất. SfM khai thác ràng buộn hình học giữa các ảnh liên tiếp để phát hiện, đối sánh các điểm đặc trưng và từ đó tính toán độ sâu, tái tạo mô hình 3D của đối tượng hoặc cảnh. Dự án Kinect Fusion [7] của Microsoft là một ví dụ nổi bật, sử dụng cảm biến Kinect để quét vật thể liên tục và tái tạo hình học 3D trong thời gian thực với độ chính xác cao hơn.

Tổng quan lại, các phương pháp truyền thống có thể chia thành hai nhóm chính: cảm biến chủ động (active sensing) và chiến lược thị giác thụ động (vision-based passive).

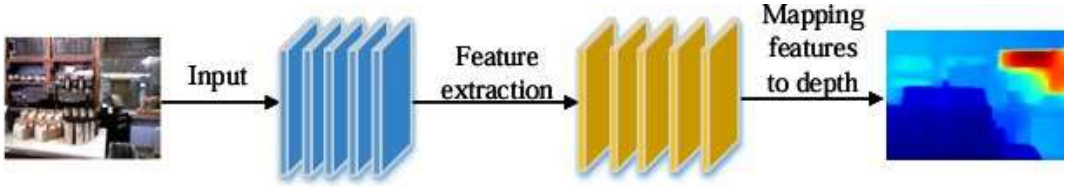
- Các kỹ thuật chủ động sử dụng nguồn phát (như laser, ánh sáng cấu trúc, hoặc cảm biến Kinect) để phát tín hiệu đến bề mặt vật thể và phân tích tín hiệu phản hồi nhằm thu thập dữ liệu 3D.
- Ngược lại, các phương pháp thị giác thụ động tận dụng thông tin từ ảnh RGB thông thường để phân tích đặc trưng và tái tạo mô hình 3D, nổi bật với ưu điểm về chi phí thấp, thiết bị đơn giản và hiệu suất cao.

b) Phương pháp học sâu

Các phương pháp học sâu trong tái tạo 3D từ ảnh có thể được chia thành hai hướng chính: mạng dự đoán độ sâu (Depth Prediction Networks) và mạng tái tạo 3D (3D Reconstruction Networks). Hai phương pháp này đều nhằm mục tiêu chuyển đổi thông tin 2D từ hình ảnh sang cấu trúc không gian 3D.

Mạng dự đoán độ sâu tập trung vào việc ước lượng thông tin độ sâu tại mỗi điểm ảnh dựa trên một hoặc nhiều ảnh đầu vào 2D. Các mạng này sử dụng các đặc trưng như kết cấu, biến đổi màu sắc, thị sai, chuyển động… để tính khoảng cách từ điểm ảnh tới gốc hệ tọa độ của camera. Thông tin độ sâu được tổ chức thành bản đồ độ sâu (depth map), đóng vai trò trung gian quan trọng trong quá trình tái tạo mô hình 3D.

Quy trình chung gồm ba bước chính: Trích xuất đặc trưng từ ảnh, ánh xạ đặc trưng sang thông tin độ sâu, và từ đó tái dựng mô hình 3D thông qua các phép biến đổi hình học và nguyên lý quang học.

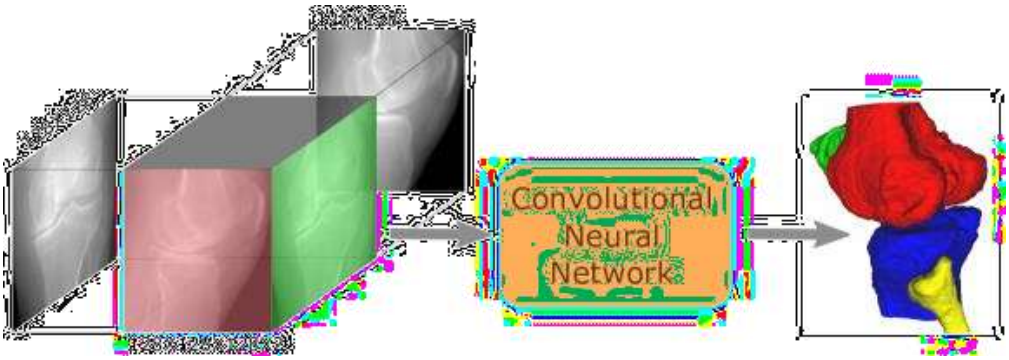


Hình 2.7: Quy trình hoạt động của mạng dự đoán độ sâu

Tóm lại, mạng dự đoán độ sâu đóng vai trò nền tảng trong mô hình hóa 3D, hỗ trợ hiệu quả việc chuyển thông tin từ ảnh sang không gian ba chiều, đồng thời mở rộng tiềm năng ứng dụng trong nhiều lĩnh vực.

Khác với mạng dự đoán độ sâu, **mạng tái tạo 3D** trực tiếp học ánh xạ từ ảnh 2D sang cấu trúc không gian 3D hoàn chỉnh. Mạng này sử dụng các kỹ thuật học sâu để trích xuất và phân tích các đặc trưng phức tạp như hình dạng, kết cấu, mối quan hệ giữa các đối tượng trong ảnh, từ đó học được mối liên hệ giữa đặc trưng ảnh và mô hình hình học 3D.

Mạng tái tạo 3D đặc biệt hiệu quả trong việc xử lý các hình dạng phức tạp và có khả năng tạo ra các mô hình 3D chính xác, liền mạch hơn. Tuy nhiên, các phương pháp này yêu cầu tài nguyên tính toán cao hơn và cần lượng dữ liệu được gán nhãn lớn, đồng thời thường phức tạp hơn về mặt huấn luyện.



Hình 2.8: Minh họa quy trình hoạt động của mạng tái tạo 3D

Tổng kết lại cả hai phương pháp truyền thống và phương pháp học sâu đều có những ưu – nhược điểm riêng biệt. Để làm rõ hơn sự khác biệt cũng như lý do tại sao các phương pháp học sâu ngày càng được ưa chuộng trong nghiên cứu và ứng dụng hiện đại, tôi đã tổng hợp một cách trực quan các đặc điểm nổi bật của hai hướng tiếp cận qua bảng dưới đây:

Bảng 2.1: So sánh phương pháp truyền thống và học sâu

	Phương pháp truyền thống	**Phương pháp học sâu**
Ưu điểm	Không yêu cầu lượng dữ liệu huấn luyện lớn như phương pháp học sâu. Có thể dễ dàng giải thích được kết quả dựa trên nguyên lý vật lý và hình học. Phù hợp trong các ứng dụng cần suy luận dựa trên quy tắc cứng (rule-based systems), đặc biệt khi điều kiện ánh sáng và góc chụp được kiểm soát tốt.	Có khả năng học đặc trưng phức tạp và trích xuất thông tin hiệu quả từ dữ liệu đầu vào mà không cần thiết kế thủ công các đặc trưng. Đạt độ chính xác cao hơn trong nhiều tác vụ, đặc biệt khi có sẵn tập dữ liệu lớn và đa dạng. Có thể tái tạo tốt cả trong môi trường không lý tưởng (nhiễu, ánh sáng yếu, vật thể biến dạng).
Nhược điểm	Độ chính xác thấp trong các môi trường phức tạp hoặc có nhiều nhiễu. Phụ thuộc nhiều vào chất lượng ảnh đầu vào, điều kiện ánh sáng,	Cần lượng lớn dữ liệu để huấn luyện mô hình hiệu quả. Mô hình khó giải thích (black-box), khó kiểm soát khi xảy ra lỗi.

	góc chụp, và các đặc trưng có thể phát hiện được. Khó áp dụng trong các bối cảnh có tính chất thay đổi cao, như ảnh ngoài trời, ảnh có vật thể đa dạng.	

2.2. Các giải pháp học sâu trong tái tạo 3D

Khi áp dụng các mô hình học sâu để sinh vật thể 3D hoặc để giải các bài toán khác như phân đoạn, nhận diện, phân loại vật thể, hình thức biểu diễn của vật thể đóng một vai trò quan trọng trong việc thiết kế mạng học sâu. Cách biểu diễn được sử dụng nhiều nhất trong phương pháp học sâu là volumetric data, mesh và point cloud.

2.2.1. Volumetric data

Ưu điểm lớn trong việc sử dụng volumetric data là sự đơn giản trong cài đặt và khả năng tương thích cao với các mạng neuron tích chập 3D (3D-CNN). Nhờ đó, nhiều mô hình học sâu đã được xây dựng trên dạng dữ liệu này. Ví dụ tiêu biểu là 3D-GAN [8], một mạng đối nghịch tạo sinh (GAN) được đề xuất để sinh ra các vật thể 3D từ không gian xác suất bằng cách sử dụng mạng CNN 3D. Mô hình này cho thấy khả năng học biểu diễn không giám sát hiệu quả hơn các phương pháp trước đó bằng cách ánh xạ từ không gian ẩn chiều thấp sang không gian đối tượng 3D.

Bên cạnh đó, 3D-R2N2 [9] là một mô hình nổi bật khác sử dụng mạng neuron hồi quy (RNN) với kiến trúc LSTM để ước lượng hình học 3D của vật thể từ nhiều ảnh chụp ở các góc nhìn khác nhau. Gần đây, 3D-FHNet được giới thiệu như một phương pháp tái tạo phân cấp có khả năng tổng hợp thông tin từ bất kỳ số lượng ảnh nào để tái dựng hình dạng 3D một cách hiệu quả hơn [10].

Tuy nhiên, một hạn chế lớn của phương pháp biểu diễn bằng voxel là chi phí tính toán và bộ nhớ rất cao, đặc biệt khi độ phân giải tăng. Điều này dẫn đến việc các chi tiết hình học nhỏ và tinh vi có thể bị mất do voxel chỉ biểu diễn dưới dạng không gian bị chiếm dụng lẫn không chiếm dụng, khiến khả năng tái tạo các phần phức tạp của vật thể bị hạn chế.

2.2.2. Mesh

Nhằm khắc phục các hạn chế của phương pháp biểu diễn thể tích, biểu diễn dạng mesh được xem là lựa chọn hấp dẫn hơn trong các ứng dụng thực tế do khả năng thể hiện chi tiết hình dạng vật thể một cách chính xác hơn. 3D mesh là cách biểu diễn phổ biến trong mô hình hóa hình dạng 3D.

Tuy nhiên, do cấu trúc của mesh không đều đặn và không tuân theo một lưới hình học chuẩn như voxel, việc áp dụng trực tiếp các mô hình học sâu để sinh ra mesh vẫn là một thách thức lớn. Một số nghiên cứu đã đề xuất cách tiếp cận thông qua tham số hóa hình học bề mặt. Ví dụ, A. Sinha et al. [11] đã đề xuất phương pháp xây dựng một hệ thống tạo hình ảnh hình học (geometry images) mã hóa tọa độ bề mặt (x, y, z) của vật thể. Ba mạng encoder-decoder riêng biệt được huấn luyện để sinh ra các ảnh hình học tương ứng với từng tọa độ x, y, z. Mỗi mạng có thể nhận đầu vào là ảnh RGB hoặc ảnh RGB-D.

Một hướng đi khác là ước lượng trường biến dạng từ ảnh đầu vào, sau đó áp dụng trường này lên một hình mẫu (template shape) để tạo ra vật thể 3D đã tái dựng. Ví dụ, DeformNet [12] là một mô hình được đề xuất để thực hiện quá trình này. Mạng nhận đầu vào là ảnh và hình mẫu 3D gần nhất từ một tập dữ liệu, sau đó sử dụng lớp Free Form Deformation (FFD) để biến dạng hình mẫu sao cho khớp với ảnh đầu vào.

Ngoài ra, một phương pháp đáng chú ý khác là Pixel2Mesh [13], cho phép tái tạo mesh trực tiếp từ một ảnh đơn. Mạng sử dụng biểu diễn mesh dưới dạng đồ thị và áp dụng graph-based CNN (GCNN) để biến dạng một hình elip ban đầu, khai thác các d dặc trưng nhận thức từ ảnh đầu vào. Phương pháp sử dụng chiến lược từ thô đến tinh (coarse-to-fine) nhằm đảm bảo quá trình biến dạng được ổn định và chính xác.

Tuy vậy, một hạn chế chung của các phương pháp dựa trên mesh là hình dạng đầu ra thường bị giới hạn bởi hình mẫu hoặc hình elip ban đầu được sử dụng trong quá trình biến dạng. Điều này khiến mô hình có thể gặp khó khăn khi tái tạo các vật thể có hình học quá phức tạp hoặc khác biệt nhiều so với hình mẫu.

2.2.3. 3D Point cloud

Để vượt qua những hạn chế đã nêu trên của các phương pháp biểu diễn volumetric và mesh, biểu diễn point cloud được sử dụng nhằm mô tả dữ liệu 3D một cách hiệu quả hơn. Dữ liệu point cloud có thể được biểu diễn dưới dạng ma trận kích thước $N \times 3$, trong đó mỗi hàng tương ứng với tọa độ x, y, z của một điểm trong không gian.

Mạng Point Set Generation Network (PSGN) [14] là mô hình đầu tiên được đề xuất để sinh point cloud từ một ảnh đầu vào duy nhất, và đã cho kết quả vượt trội hơn so với các phương pháp volumetric. Trong mô hình RealPoint3D [15], mạng được thiết kế với hai encoder: một encoder trích xuất đặc trưng 2D từ ảnh đầu vào, encoder còn lại trích xuất đặc trưng 3D từ vật thể tương tự nhất về hình dạng được truy xuất từ bộ dữ liệu ShapeNet. Các đặc trưng này sau đó được kết hợp và đưa vào một decoder để tạo

ra point cloud chi tiết. Việc sử dụng point cloud từ vật thể truy xuất giúp cải thiện độ chính xác và chi tiết của kết quả tái dựng.

Một hướng tiếp cận khác là sử dụng mạng học sâu để sinh trực tiếp point cloud từ ảnh RGB mà không cần truy xuất hình mẫu tương đồng. Pixel2Point [16] là một ví dụ điển hình cho hướng tiếp cận này. Mô hình có thiết kế đơn giản, chỉ sử dụng một đầu vào duy nhất là ảnh RGB và một point cloud khởi tạo ban đầu (initial point cloud), sau đó dự đoán trực tiếp point cloud đầu ra. Khác với RealPoint3D, Pixel2point không sử dụng giám sát 2D như silhouettes hay thông tin từ các mô hình 3D tương đồng. Thay vào đó, nó tận dụng đặc trưng hình ảnh để cải thiện độ chính xác trong việc tái dựng cấu trúc 3D của vật thể.

2.3. Các bộ dữ liệu 3D phổ biến

Các bộ dữ liệu 3D đóng vai trò rất quan trọng trong việc phát triển và đánh giá các phương pháp học sâu trong tái tạo hình học ba chiều. Tùy vào mục tiêu cụ thể như phân loại, phân đoạn, hoặc tái tạo 3D, mỗi bộ dữ liệu cung cấp những dạng biểu diễn và thông tin khác nhau. Trong phần này, tôi sẽ trình bày và so sánh một số bộ dữ liệu phổ biến, đồng thời đi sâu hơn vào bộ dữ liệu ShapeNet.

2.3.1. ShapeNet

ShapeNet [17] là một trong những bộ dữ liệu 3D đa dạng và phong phú nhất, ShapeNet chứa hơn 3 triệu mô hình CAD với khoảng 220.000 mô hình được phân loại theo hơn 3100 danh mục (WordNet synsets). Subset phổ biến nhất của bộ dữ liệu này là ShapeNetCore với khoảng 51.300 mô hình và 55 danh mục chủ yếu dành cho nghiên cứu học sâu. Ngoài ra còn có ShapeNetSem với 12.000 mô hình và 270 danh mục dành cho nghiên cứu ngữ nghĩa. Các mô hình trong ShapeNet được chuẩn hóa về tọa độ, hướng và kích thước, đồng thời được gán nhãn theo phân cấp WordNet giúp thuận tiện cho các bài toán phân loại hoặc tái tạo theo từng lớp.

Hình 2.9: Các mô hình 3D trong tập dữ liệu ShapeNet

Trong khuôn khổ đồ án này, ShapeNetCore được sử dụng làm bộ dữ liệu huấn luyện cho mô hình trong hệ thống, nhờ lượng dữ liệu lớn và phân loại rõ ràng. Đây là bộ dữ liệu phổ biến trong các công trình học sâu khác như PointNet, Pixel2Point, DGCNN,… giúp mô hình học ánh xạ hình ảnh sang biểu diễn point cloud 3D.

2.3.2. ModelNet

ModelNet [18] là bộ dữ liệu CAD 3D do Princeton phát triển, là một trong những bộ dữ liệu 3D nền tảng được sử dụng rộng rãi, đặc biệt trong các nghiên cứu ban đầu về phân loại và nhận dạng hình học. ModelNet40 và ModelNet10 là hai phiên bản phổ biến nhất, lần lượt chứa 12.311 và 4.899 mô hình 3D, được chia thành các lớp như ghế, giường, tủ lạnh, và đồ gia dụng khác. Mặc dù ít đa dạng hơn ShapeNet, ModelNet vẫn giữ vai trò quan trọng trong việc so sánh và đánh giá các mô hình học sâu trên những tập dữ liệu chuẩn hóa.

2.3.3. ScanNet

ScanNet [19] là một bộ dữ liệu thu thập từ các cảnh trong nhà thực tế bằng cảm biến RGB-D. Bộ dữ liệu bao gồm hơn 1.500 cảnh với thông tin hình học, màu sắc, bản đồ độ sâu, và nhãn phân đoạn ngữ nghĩa. ScanNet thường được sử dụng trong các bài toán tái dựng môi trường thực tế, phân đoạn điểm, hoặc SLAM (Simultaneous Localization and Mapping). Do trọng tâm nghiên cứu của đề tài này là tái dựng các đối tượng riêng lẻ từ ảnh RGB, nên ScanNet chỉ mang tính chất tham khảo như một ví dụ tiêu biểu cho các bộ dữ liệu 3D trong môi trường thực.

2.3.4. Pix3D

Pix3D [20] là một bộ dữ liệu kết hợp giữa ảnh RGB thực tế và mô hình 3D tương ứng, nhằm mục đích nghiên cứu các bài toán tái dựng 3D từ ảnh đơn. Mỗi mẫu dữ liệu bao gồm ảnh RGB, bản đồ độ sâu, và mô hình mesh tương ứng, cùng với thông tin căn chỉnh pose chính xác. Bộ dữ liệu này đặc biệt hữu ích cho các mô hình cần học ánh xạ từ không gian ảnh sang không gian 3D, đóng vai trò như một benchmark thực tế hơn so với các tập dữ liệu chỉ chứa mô hình 3D.

Bảng 2.2: Khảo sát các bộ dữ liệu 3D

Bộ dữ liệu	**Năm**	**Số lượng mẫu dữ liệu**	**Định dạng dữ liệu**	**Khả năng kết hợp với ảnh 2D**	**Ứng dụng**
ShapeNetCore	2015	Hơn 50.000	Mesh, point cloud	Có hỗ trợ với ShapeNetRender	Tái tạo vật thể 3D
ModelNet	2015	12.311 (ModelNet40) 4.899 (ModelNet10)	Mesh	Không hỗ trợ	Benchmark phân loại 3D
Pix3D	2018	Khoảng 10.000	Ảnh RGB, depth map, mesh	Có sẵn (ảnh RGB, bản đồ độ sâu, mesh)	Tái tạo vật thể 3D, benchmark thực tế
ScanNet	2017	Hơn 1.500 cảnh quét thực tế	RGB-D video, mesh, point cloud, annotation	Có hỗ trợ (RGB-D sequence, frame-level annotation)	Phân đoạn cảnh indoor, SLAM, AR/VR

Từ bảng so sánh trên có thể thấy, ShapeNet mà cụ thể là ShapeNetCore là bộ dữ liệu phù hợp nhất cho bài toán tái tạo vật thể 3D từ ảnh đơn nhờ số lượng mẫu phong phú, đa dạng lớp danh mục, dữ liệu đã được chuẩn hóa và đặc biệt là khả năng kết hợp với ảnh 2D thông qua bộ ShapeNetRenderer, giúp mô hình học được mối liên kết giữa ảnh và biểu diễn hình học 3D một cách hiệu quả.

2.4. Mô hình Pixel2Point

Pixel2Point (Afifi et al., 2021) [16] là một mạng CNN được huấn luyện end-to-end giám sát trên tập dữ liệu ShapeNet. Mô hình nhận một ảnh đầu vào của một vật thể và sinh ra point cloud của vật thể đó với số điểm cố định. Điểm nổi bật trong kiến trúc mạng Pixel2Point là tác giả đề xuất khởi tạo một point cloud hình cầu ban đầu (initial point cloud) nhằm giúp cải thiện chất lượng của point cloud đầu ra. Tác giả cho rằng với phương pháp đề xuất của mình, mô hình có hiệu suất vượt trội hơn so với các phương pháp state-of-the-art (SOTA) trong việc giải quyết bài toán cả về mặt định lượng lẫn định tính trên dữ liệu tổng hợp và dữ liệu ảnh thật.

Kiến trúc mạng bao gồm hai thành phần chính là bộ mã hóa (**encoder**) và bộ tạo điểm (**generator**). Ngoài ra intial point cloud cố định dạng hình cầu với vai trò là dữ liệu trung gian, được đưa vào như một đầu vào phụ nhằm hỗ trợ việc tái tạo hình học ba chiều. Ý tưởng then chốt của phương pháp này là kết hợp giữa đặc trưng hình ảnh và cấu trúc không gian 3D sơ khai để sinh ra một point cloud mô tả đầy đủ và chi tiết vật thể.

Thành phần đầu tiên của mô hình là một bộ mã hóa tích chập (**CNN encoder**), có vai trò trích xuất đặc trưng hình ảnh cấp cao từ ảnh RGB đầu vào. Kiến trúc của encoder bao gồm 7 lớp tích chập 2D liên tiếp, mỗi lớp đều được theo sau bởi hàm kích hoạt ReLU nhằm tăng tính phi tuyến và khả năng biểu diễn của mạng.

- Ba lớp đầu tiên sử dụng số lượng kênh lần lượt là 32, 64 và 128, giúp mạng học được các đặc trưng cơ bản như biên cạnh, hình dạng khối và màu sắc.
- Bốn lớp tiếp theo sử dụng 256 kênh, giúp mã hóa những đặc trưng trừu tượng hơn như hình học ba chiều tiềm ẩn và quan hệ không gian.
- Tất cả các lớp đều sử dụng kernel size 3×3 và stride = 2, điều này không những giúp giảm dần kích thước không gian (giống như pooling) mà còn giữ lại các tham số có thể học được, giúp quá trình trích xuất đặc trưng hiệu quả và tối ưu hơn.

Kết quả đầu ra của encoder là một tensor kích thước 1×1×256, tương ứng với một vector đặc trưng duy nhất kích thước 256 chiều mô tả nội dung toàn ảnh. Tensor này được reshape về dạng vector 1×256 và được sử dụng để kết hợp với initial point cloud trong giai đoạn tiếp theo.

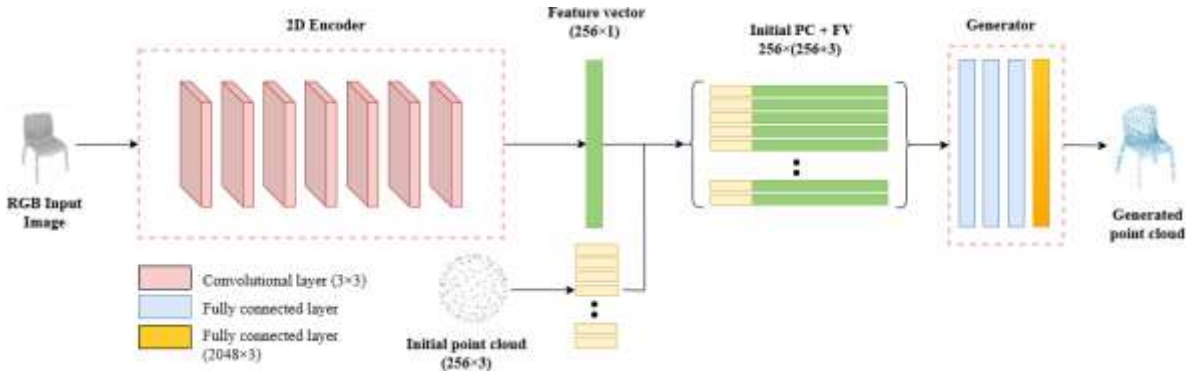
Một điểm nổi bật trong thiết kế của Pixel2Point là việc sử dụng một tập điểm ba chiều ban đầu cố định (**initial point cloud**) được xây dựng trước và đưa vào mạng như một dạng hình học khởi tạo. Dữ liệu này có dạng một hình cầu đều (uniform sphere)

gồm 256 điểm 3D. Đặc biệt, hình cầu này không được chọn ngẫu nhiên mà được sinh ra bằng thuật toán Fibonacci sphere – một thuật toán phổ biến giúp phân bố đều các điểm trên bề mặt cầu. Việc này giúp đảm bảo point cloud sinh ra có phân bố điểm đồng đều, tránh tụ điểm ở cực. Ngoài ra initial point cloud giúp cung cấp thông tin cấu trúc 3D sơ khai giúp mô hình dễ học hơn và tăng tính tổng quát của mô hình vì không phụ thuộc vào một template hình học cụ thể.

Tập điểm 256×3 này sau đó được nối (concatenate) với vector đặc trưng 1×256 được trích xuất từ encoder, mỗi điểm trong initial point cloud sẽ được nối với cùng một vector đặc trưng này, tạo thành một tensor đầu vào mới cho generator có kích thước 256×(3 + 256).

Bộ sinh (**generator**) là một mạng đơn giản gồm bốn lớp fully connected (FC) có chức năng ánh xạ từ không gian đặc trưng kết hợp sang không gian tọa độ 3D thực. Ba lớp đầu tiên sử dụng hàm kích hoạt ReLU, giúp mô hình học các phép biến đổi phi tuyến. Lớp cuối cùng không sử dụng hàm kích hoạt, nhằm đảm bảo đầu ra là các giá trị tọa độ thực.

Đầu ra cuối cùng của generator là một point cloud gồm 2048 điểm, mỗi điểm có tọa độ không gian 3 chiều (x, y, z) được biểu diễn dưới dạng ma trận kích thước 2048×3. Việc sinh nhiều điểm hơn so với initial point cloud cho phép mô hình nội suy và tái tạo hình học chi tiết hơn, bao gồm cả những phần không nhìn thấy trong ảnh đầu vào.



Hình 2.10: Kiến trúc của mô hình Pixel2Point

Tác giả lựa chọn số điểm của point cloud đầu ra từ mạng CNN là $N = 2048$. Từ nhiều thử nghiệm trước đó, đây là số điểm vừa đủ để phủ kín toàn bộ bề mặt của vật thể và bảo toàn được các cấu trúc chính.

2.5. Tiêu chí đánh giá mô hình dự đoán

Trong các bài toán học máy truyền thống, đặc biệt là các bài toán phân loại (classification) hoặc phát hiện đối tượng (object detection), các chỉ số như accuracy, precision, recall hay F1-score thường được sử dụng rộng rãi để đánh giá chất lượng mô

hình. Các chỉ số này có ý nghĩa rõ ràng vì đầu ra của mô hình là các nhãn rời rạc hoặc bounding box của đối tượng, và có thể dễ dàng so sánh với ground truth dưới dạng "đúng" hoặc "sai".

Tuy nhiên, trong bài toán tái tạo vật thể dưới dạng point cloud từ ảnh 2D, bản chất đầu ra của mô hình là một tập hợp điểm ba chiều không có thứ tự cố định (unordered 3D points). Do đó, việc đánh giá chất lượng đầu ra trở nên phức tạp hơn nhiều vì:

- Không tồn tại sự tương ứng một-một rõ ràng giữa các điểm trong point cloud dự đoán và ground truth. Hai tập điểm có thể mô tả hình dạng giống nhau nhưng không trùng khớp từng điểm cụ thể.
- Các điểm trong point cloud không có thứ tự như chuỗi dữ liệu, nên không thể áp dụng các metric như accuracy vốn đòi hỏi phải xác định từng phần tử có đúng vị trí hay không.

Chính vì vậy, để đánh giá hiệu quả mô hình trong bối cảnh này, cần phải chuyển hướng sang các tiêu chí đo mức độ tương đồng về hình học giữa hai tập điểm. Thay vì xác định một điểm có chính xác hay không, các metric mới sẽ đo khoảng cách trung bình hoặc toàn cục giữa các điểm trong hai tập point cloud, phản ánh mức độ "gần giống" giữa vật thể được dự đoán và ground truth.

Việc lựa chọn hàm mất mát phù hợp đóng vai trò then chốt trong quá trình huấn luyện mô hình học sâu, đặc biệt là trong các kiến trúc CNN áp dụng cho bài toán tái tạo point cloud. Hàm mất mát phải thỏa mãn các tiêu chí sau:

- Tính khả vi (differentiable) và hiệu quả tính toán (computationally cheap) để đảm bảo quá trình lan truyền ngược diễn ra mượt mà và khả thi trên các mô hình lớn.
- Tính ổn định đối với nhiễu hoặc điểm ngoại lai, vì trong quá trình tái tạo, mô hình có thể dự đoán ra những điểm lệch lớn do nhiễu từ ảnh đầu vào hoặc lỗi hội tụ.

Để đáp ứng các yêu cầu trên, các nghiên cứu hiện đại thường định nghĩa hàm mất mát L giữa hai point cloud, ký hiệu là $S_{pred}, S_{gt} \subset \mathbb{R}^3$ như sau:

$$L(S_{pred}, S_{gt}) = X_d(S_{pred}, S_{gt})$$

Trong đó:

- S_{pred}: point cloud được mô hình sinh ra
- S_{gt}: point cloud ground truth

- X_d: hàm đo khoảng cách giữa hai tập hợp điểm

Hiện nay, hai metric được sử dụng phổ biến và đã trở thành chuẩn mực trong cộng đồng nghiên cứu là Chamfer Distance (CD) và Earth Mover's Distance (EMD).

2.5.1. Chamfer Distance

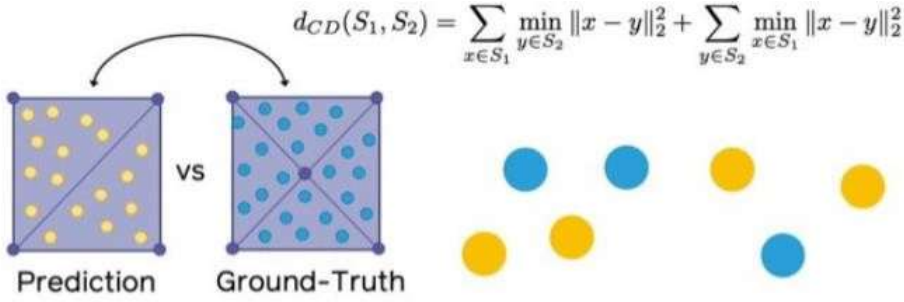
Chamfer Distance (CD) là một hàm đo khoảng cách phổ biến được sử dụng rộng rãi trong các bài toán học sâu liên quan đến tái tạo vật thể 3D dưới dạng point cloud. Khác với các metric truyền thông như accuracy hay F1-score, CD đo lường mức độ tương đồng hình học giữa hai tập điểm không có thứ tự cố định – cụ thể là giữa tập điểm dự đoán và ground truth.

Cho hai tập điểm trong không gian 3 chiều $S_1, S_2 \subset \mathbb{R}^3$, Chamfer Distance được định nghĩa như sau:

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2$$

Trong đó:

- Thành phần đầu tiên đại diện cho trung bình khoảng cách từ mỗi điểm trong tập dự đoán S_1 đến điểm gần nhất trong tập ground truth S_2.
- Thành phần thứ hai thực hiện điều tương tự nhưng theo chiều ngược lại, từ ground truth đến tập dự đoán.



Hình 2.11: Chamfer Distance

Một trong những ưu điểm nổi bật của CD là tính khả vi và liên tục. Cụ thể, đây là một hàm liên tục và khả vi theo từng phần, cho phép gradient có thể được lan truyền ngược trong quá trình huấn luyện mô hình học sâu. Ngoài ra, CD cũng hiệu quả về mặt tính toán. Việc tìm điểm gần nhất giữa hai tập point cloud là một quá trình độc lập cho từng điểm, không phụ thuộc vào thứ tự hay cấu trúc toàn cục. Do đó quá trình này có thể được song song hóa dễ dàng trên GPU, giúp tăng tốc độ huấn luyện đáng kể và tối ưu hóa mô hình trên tập dữ liệu lớn. Một điểm mạnh khác là nó không yêu cầu ánh xạ

một-một giữa các điểm trong hai tập. Điều này đặc biệt quan trọng vì dữ liệu point cloud là dạng unordered, và việc xác định tương ứng chính xác từng điểm giữa đầu ra và ground truth thường không khả thi hoặc không cần thiết.

Tuy nhiên, CD vẫn tồn tại một số hạn chế như thiếu cơ chế kiểm soát sự phân bố đều (uniformity) của các điểm trong tập point cloud được sinh ra. Trong quá trình tối ưu, mô hình có thể học cách tái tạo một phần hình dạng với mật độ điểm rất cao, trong khi các vùng còn lại thưa thớt hoặc thậm chí bị bỏ trống. Điều này làm giảm chất lượng hình học tổng thể của vật thể được tái tạo.

Bên cạnh đó, CD cũng có xu hướng dẫn đến hiện tượng hội tụ tại các cực tiểu cục bộ. Do sự tự do trong việc chọn vị trí điểm gần nhất, mô hình có thể đạt được giá trị loss thấp mà không cần tái hiện đầy đủ hình dạng thực tế, dẫn đến các kết quả thiếu chi tiết hoặc không đồng đều. Hiện tượng này khá phổ biến khi mô hình không được khởi tạo tốt hoặc không có các cơ chế bổ sung để duy trì tính đa dạng hình học trong tập điểm.

2.5.2. Earth Mover's Distance

Earth Mover's Distance (EMD) là một metric đo lường mức độ tương đồng hình học giữa hai tập point cloud bằng cách xác định phép ánh xạ tối ưu giữa chúng. Khác với CD, EMD nhắm tới việc xây dựng mối quan hệ một-một (bijection) giữa hai tập điểm, đảm bảo cấu trúc được giữ lại một cách toàn diện.

Cho hai tập điểm bất kỳ $S_1, S_2 \subset \mathbb{R}^3$, trong đó $|S_1| = |S_2| = N$, EMD được định nghĩa như sau:

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \|x - \phi(x)\|_2$$

Trong đó, ϕ là ánh xạ một-một giữa hai tập điểm, đảm bảo mỗi điểm trong S_1 được ghép với một điểm duy nhất trong S_2 và ngược lại. EMD đo tổng bình phương khoảng cách Euclidean giữa từng cặp điểm tương ứng theo ánh xạ này. Nhờ đó, EMD phản ánh chính xác mức độ khác biệt về hình học tổng thể giữa hai hình dạng 3D được biểu diễn dưới dạng point cloud.

Ưu điểm lớn nhất của EMD là khả năng duy trì cấu trúc hình học toàn diện của đối tượng. Do EMD yêu cầu ánh xạ một-một giữa hai tập điểm, nó đảm bảo mỗi điểm trong tập điểm được đối sánh với một điểm tương ứng duy nhất, giúp tránh được các hiện tượng như bỏ sót một phần hình dạng hoặc phân bổ quá dày điểm ở một vùng cụ thể. EMD còn được xem là một metric đúng nghĩa trong không gian toán học. Khoảng cách này được xây dựng dựa trên lý thuyết Wasserstein-1, vốn có các tính chất hình thức

của một metric, bao gồm tính đối xứng, bất đẳng thức tam giác và nhận giá trị bằng 0 khi hai tập điểm trùng khớp hoàn toàn.. Điều này khiến EMD trở thành metric lý tưởng để đánh giá độ tương đồng giữa các cấu trúc hình học một cách chính xác và nhạy cảm với các biến đổi không gian.

Tuy nhiên, EMD tồn tại hạn chế lớn là chi phí tính toán. Việc xác định ánh xạ tối ưu giữa hai tập điểm đòi hỏi phải giải một bài toán tối ưu kiết hợp đôi (optical matching), thường được biểu diễn dưới dạng bài toán lập trình tuyến tính. Trong thực tế, bài toán này thường được giải bằng thuật toán Hungarian với độ phức tạp thời gian là $O(N^3)$, với N là số điểm trong mỗi tập. Điều này dẫn đến việc tiêu tốn đáng kể tài nguyên bộ nhớ và thời gian xử lý, đặc biệt khi làm việc với các point cloud có độ phân giải cao.

Do đó, việc sử dụng EMD làm hàm mất mát trong các mô hình học sâu quy mô lớn thường không khả thi. Mặc dù EMD cung cấp đánh giá chính xác hơn về mặt hình học nhưng chi phí tính toán cao khiến nó ít được áp dụng trực tiếp trong quá trình huấn luyện so với lựa chọn như CD.

2.6. Các thuật toán upsampling point cloud

Mặc dù nhiệm vụ chính của hệ thống là tái tạo vật thể 3D từ ảnh 2D, đầu ra dưới dạng point cloud thường có mật độ khá thưa do giới hạn của mô hình dự đoán hoặc dữ liệu huấn luyện. Do đó, upsampling point cloud trở thành công đoạn quan trọng để:

- Tăng độ dày và chi tiết hình học, giúp mô hình tái tạo mesh mượt và chính xác.
- Giảm các vùng thiếu điểm, cải thiện chất lượng tổng thể.
- Chuẩn hóa mật độ điểm, hỗ trợ tốt các bước tiếp theo như mesh reconstruction và render.

Các thuật toán upsampling point cloud thường được chia làm 2 nhóm chính: thuật toán truyền thống và phương pháp sử dụng học sâu.

2.6.1. Thuật toán truyền thống

Các phương pháp truyền thống thường dựa vào các nguyên lý nội suy và tối ưu hóa hình học mà không cần đến dữ liệu nhãn (ground truth) hoặc huấn luyện phức tạp. Mặc dù chúng không đạt được chất lượng như các phương pháp học sâu, nhưng vẫn là lựa chọn đơn giản, nhanh chóng và ít tiêu tốn tài nguyên tính toán.

a) Moving Least Squares (MLS)

MLS là phương pháp tái tạo bề mặt liên tục từ dữ liệu điểm rời rạc thông qua tối ưu hóa bình phương tối thiểu có trọng số, đặc biệt hiệu quả cho upsampling point cloud.

MLS xây dựng một hàm ẩn $I(p)$ xác định khoảng cách có dấu đến bề mặt mục tiêu thông qua quy trình tối ưu hóa trọng số cục bộ. Với mỗi điểm p trong không gian, MLS giải bài toán:

$$\min_{a} \sum_{i=1}^{N} \theta(\|p - p_i\|) \cdot (f_a(p_i) - z_i)^2$$

Trong đó:

- $f_a(p_i) = \boldsymbol{a^T} \cdot \boldsymbol{b}(p_i)$: Hàm xấp xỉ đa thức (thường bậc 1 hoặc 2)
- θ: Hàm trọng số Gaussian $\theta(d) = e^{-d^2/h^2}$, với h là bán kính ảnh hưởng
- z_i: Giá trị mục tiêu (thường là tọa độ z hoặc độ lệch từ mặt phẳng tiếp tuyến)

Quy trình xử lý point cloud của MLS gồm 4 bước:

- Xác định vùng lân cận động: Với mỗi điểm p, MLS tự động điều chỉnh bán kính lân cận h dựa trên mật độ điểm, đảm bảo số lượng điểm tham gia tối ưu (thường 15-30 điểm). Điều này khác biệt với các phương pháp lấy mẫu cố định như voxel grid.
- Ước lượng mặt phẳng tiếp tuyến: MLS sử dụng phân tích thành phần chính (PCA) trên các điểm lân cận để xác định mặt phẳng tiếp tuyến tối ưu, giảm nhiễu bằng cách gán trọng số nghịch đảo với khoảng cách.
- Projection và làm mịn: Các điểm được chiếu lên mặt phẳng tiếp tuyến và làm mịn bằng hàm đa thức cục bộ. Quá trình này lặp lại cho đến khi hội tụ, đảm bảo tính liên tục C^k (thường $k = 2$).
- Tái thiết bề mặt: Hàm ẩn $I(p)$ được sử dụng để sinh điểm mới thông qua upsampling và smoothing.

b) Poisson Disk Sampling

Poisson Disk Sampling là một kỹ thuật chọn mẫu điểm nhằm tạo ra các điểm phân bố đều, tránh hiện tượng tập trung hoặc quá thưa, rất phù hợp cho việc upsampling point cloud để tăng mật độ điểm mà vẫn giữ được tính đồng đều và chi tiết hình học. Thuật toán này đảm bảo rằng khoảng cách giữa bất kỳ hai điểm được chọn đều lớn hơn một giá trị tối thiểu gọi là bán kính Poisson (disk radius). Điều này giúp tránh việc các điểm mới bị chồng lấn hoặc quá gần nhau, tạo ra phân bố điểm đều và tự nhiên trên bề mặt. Thuật toán gồm các bước:

- Khởi tạo: Chọn ngẫu nhiên một điểm đầu tiên trong không gian mẫu.

- Sinh điểm mới: Từ mỗi điểm hiện có, sinh các điểm ứng viên trong vòng bán kính gấp đôi bán kính Poisson.
- Lặp lại: Tiếp tục quá trình cho đến khi không thể sinh thêm điểm mới thỏa mãn điều kiện khoảng cách.

Thuật toán này có thể được tối ưu bằng cách sử dụng cấu trúc dữ liệu như k-d tree hoặc voxel grid để tăng tốc việc kiểm tra khoảng cách và tìm kiếm điểm gần nhất.

c) Farthest Point Sampling (FPS)

Trong bài toán upsampling point cloud, thuật toán FPS được sử dụng để chọn ra các điểm phân bố đều trên bề mặt, nhằm tăng mật độ điểm một cách đồng đều và hiệu quả. FPS bắt đầu với một điểm ngẫu nhiên trong tập điểm gốc, sau đó lặp lại chọn điểm xa nhất so với tập điểm đã chọn, nhằm đảm bảo các điểm được trải đều trên toàn bộ không gian. Quá trình này giúp tránh việc các điểm mới bị tập trung quá mức hoặc thưa thớt. Quy trình hoạt động của thuật toán toán gồm:

- Chọn điểm hạt giống (seed points): FPS được dùng để lấy mẫu các điểm đại diện từ point cloud ban đầu, làm cơ sở cho việc sinh thêm điểm mới trong các vùng lân cận.
- Phân chia cục bộ: Từ các điểm hạt giống, các vùng lân cận được xác định để thực hiện upsampling cục bộ, giúp duy trì chi tiết hình học và cấu trúc bề mặt.
- Kết hợp điểm mới: Sau khi sinh thêm điểm trong các vùng này, FPS cũng có thể được áp dụng để điều chỉnh lại phân bố điểm nhằm đảm bảo tính đồng đều cho toàn bộ point cloud.

2.6.2. Phương pháp học sâu

Các kỹ thuật upsampling dựa trên học sâu đã có nhiều tiến bộ gần đây, từ các mô hình giám sát đến các giải pháp hiện đại như học tự giám sát và hỗ trợ upsampling ở bất kỳ tỉ lệ nào. Các mô hình học sâu phổ biến hiện nay gồm có PU-Net (Yu et al., 2018) [23], PU-GCN (Qian et al., 2021) [24],… Tuy nhiên hầu hết các phương pháp trên đều tồn tại một số hạn chế đáng chú ý:

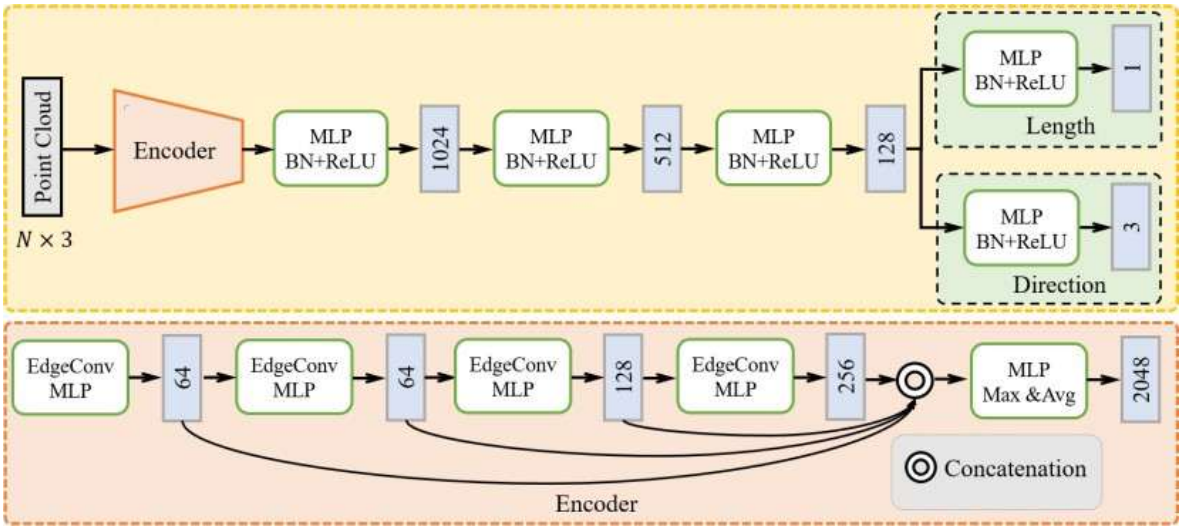
- Phụ thuộc vào dữ liệu ground truth dạng dense để huấn luyện, điều này không khả thi trong nhiều tình huống thực tế do việc thu thập dữ liệu dày đặc tốn kém và khó khăn.
- Nhiều mô hình chỉ hỗ trợ upsampling ở tỉ lệ cố định (ví dụ ×2, ×4), hạn chế tính linh hoạt khi áp dụng vào các hệ thống 3D có yêu cầu thay đổi mật độ điểm động.

Để khắc phục những điểm yếu này, mô hình SAPCU (Self-Supervised Arbitrary-Scale Point Cloud Upsampling) (Zhao et al., 2022) đã được đề xuất [25]. Đây là một kiến trúc học sâu không yêu cầu dữ liệu ground truth dense và có khả năng upsampling ở bất kỳ tỉ lệ nào nhờ cơ chế học tự giám sát và biểu diễn ngầm.

Ý tưởng cốt lõi của SAPCU là xem quá trình upsampling như một bài toán chiếu các seed points lên một bề mặt ngầm (implicit surface). Để thực hiện điều này, mô hình học 2 hàm ẩn chính:

- **Direction Estimator (f_n)**: Dự đoán hướng chiếu từ seed point đến bề mặt.
- **Distance Estimator (f_d)**: Dự đoán độ dài đoạn chiếu để tìm điểm gần nhất trên bề mặt mục tiêu.

Cả hai hàm này đều được huấn luyện theo phương pháp tự giám sát thông qua các tác vụ học tiền nhiệm (pretext learning tasks), hoàn toàn không cần cặp dữ liệu sparse-dense.



Hình 2.12: Kiến trúc của mô hình SAPCU

Kiến trúc mô hình gồm 2 thành phần chính:

- **Encoder:** Mạng trích xuất đặc trưng dựa trên DGCNN [26], có khả năng nắm bắt thông tin hình học cục bộ và toàn cục nhiều cấp độ. Dữ liệu đầu vào là một tập con điểm đã được chuẩn hóa, qua encoder được biểu diễn thành vector đặc trưng 2048 chiều.
- **Implicit Decoders:** Cả hai hàm f_n và f_d sử dụng kiến trúc giống nhau, gồm 4 lớp fully-connected (FC) liên tiếp. Mỗi lớp FC được kết hợp với batch normalization và hàm kích hoạt ReLU, với số chiều lần lượt là 1024, 512, 128. Lớp cuối có kích thước là 3 chiều với f_n và 1 chiều với f_d.

2.7. Mesh reconstruction

Mesh reconstruction là quá trình chuyển đổi dữ liệu điểm rời rạc là point cloud thành biểu diễn bề mặt dạng lưới tam giác (triangular mesh) liên tục, cho phép mô hình hóa chính xác hình dạng hình học của đối tượng 3D. Trong thực tế, nhiều thuật toán mesh reconstruction đã được đề xuất, phổ biến nhất là các phương pháp dựa trên hình học (geometry-based) và phương pháp dựa trên trường hàm (function-based).

2.7.1. Phương pháp hình học

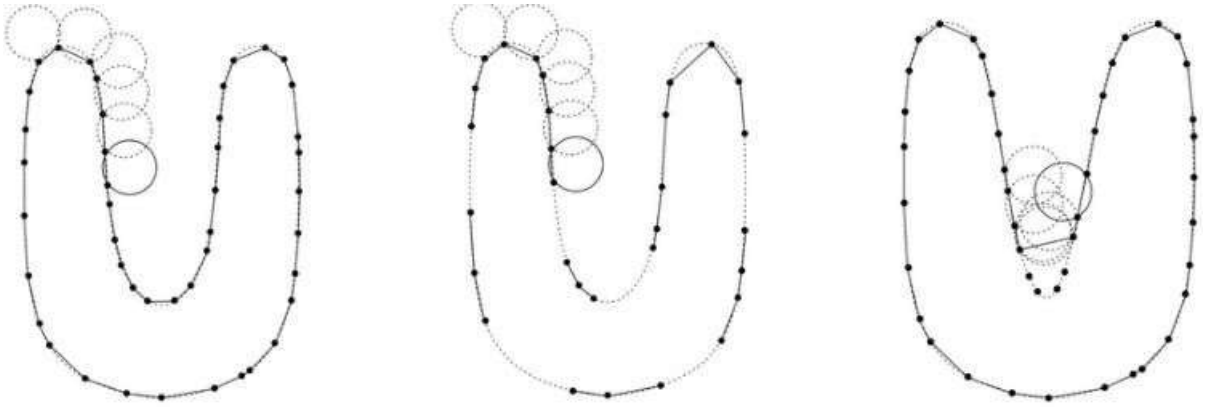
Các phương pháp này khai thác trực tiếp đặc trưng hình học của point cloud (vị trí, khoảng cách, vector pháp tuyến) để kết nối các điểm thành lưới tam giác. CHúng thường yêu cầu dữ liệu có mật đồ đều, ít nhiễu và đầy đủ.

a) Ball Pivoting Algorithm (BPA)

Thuật toán Ball Pivoting là một phương pháp hình học phổ biến để tái tạo lưới tam giác (mesh) từ point cloud có mật độ đều và đi kèm vector pháp tuyến. Ý tưởng cốt lõi là lăn một quả cầu ảo có bán kính cố định dọc theo bề mặt dữ liệu để xác định các tam giác hợp lệ.

Quá trình hoạt động của BPA diễn ra như sau:

- Một quả cầu được đặt sao cho nó tiếp xúc với ba điểm dữ liệu, tạo thành một tam giác nếu không có điểm nào khác nằm bên trong quả cầu đó.
- Sau khi tam giác đầu tiên được xác định, quả cầu lăn dọc theo cạnh của tam giác hiện tại để tìm điểm thứ ba kế tiếp, sao cho tiếp tục tạo thành tam giác mới.
- Quá trình này tiếp tục cho đến khi không còn tam giác mới nào có thể được tạo.



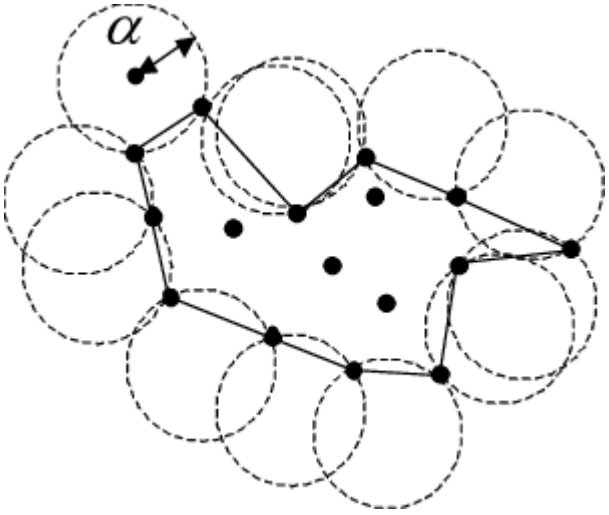
Hình 2.13: Quá trình hoạt động của Ball Pivoting Algorithm

Ưu điểm của BPA là khả năng tạo ra các mesh mịn, liên tục và có chất lượng cao nếu dữ liệu đầu vào đầy đủ và có vector pháp tuyến chính xác. Nhược điểm là nó phụ thuộc mạnh vào việc chọn đúng bán kính quả cầu, và dễ thất bại nếu dữ liệu bị thiếu, phân bố không đều hoặc chứa nhiễu.

b) Alpha Shapes

Thuật toán Alpha Shapes là phương pháp hình học tổng quát hóa khái niệm "convex hull" để mô tả biên của tập điểm theo nhiều mức độ chi tiết khác nhau. Phương pháp này sử dụng một tham số α để điều khiển mức độ "lỏng" của bề mặt bao phủ.

- Với mỗi giá trị α, thuật toán xây dựng một tập hợp các tam giác sao cho các hình tròn ngoại tiếp của chúng có bán kính nhỏ hơn hoặc bằng α.
- Khi α rất lớn, kết quả là một khối lồi (convex hull); khi α giảm, thuật toán sẽ loại bỏ dần các tam giác không phù hợp, làm lộ ra các chi tiết nhỏ hơn trong cấu trúc dữ liệu.
- Kết quả thu được là một lưới tam giác phản ánh hình dạng của dữ liệu với mức độ chi tiết tùy chỉnh.



Hình 2.14: Quá trình hoạt động của Alpha Shapes

Ưu điểm của Alpha Shapes là khả năng kiểm soát chi tiết bề mặt thông qua tham số α và tái dựng được các bề mặt có lỗ hoặc hình dạng phức tạp. Tuy nhiên, nhược điểm là việc chọn α phù hợp không dễ, và phương pháp nhạy cảm với nhiễu.

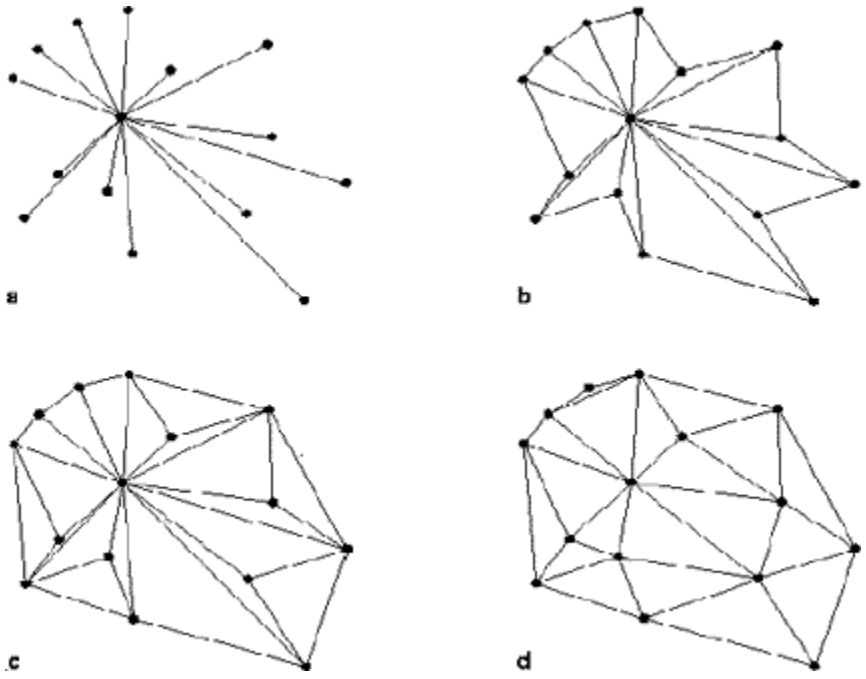
c) Delaunay Triangulation

Delaunay Triangulation là một thuật toán chia mặt phẳng thành các tam giác sao cho không có điểm nào nằm bên trong đường tròn ngoại tiếp của bất kỳ tam giác nào

trong phân hoạch. Phương pháp này được sử dụng rộng rãi để xây dựng lưới từ dữ liệu điểm nhờ các tính chất hình học tốt của tam giác Delaunay.

Quá trình gồm các bước:

- Dựng lưới tam giác từ tập điểm bằng thuật toán Delaunay trong không gian 2D hoặc 3D.
- Áp dụng các tiêu chí lọc để loại bỏ các tam giác không thuộc về bề mặt thực (dựa trên độ dài cạnh, diện tích hoặc hướng pháp tuyến).
- Kết nối các tam giác còn lại để tạo thành một bề mặt liên tục.



Hình 2.15: Quá trình hoạt động của Delaunay Triangulation

Ưu điểm là thuật toán đơn giản, dễ triển khai và tạo ra các lưới tam giác có chất lượng cao. Tuy nhiên cũng như các thuật toán hình học kể trên, khi gặp dữ liệu point cloud mật độ thấp, không đồng đều hoặc thiếu thông tin về pháp tuyến, thuật toán có thể cho ra kết quả không như mong muốn.

2.7.2. Phương pháp trường hàm

Nhóm phương pháp này không trực tiếp kết nối các điểm để tạo mesh, thay vào đó thì học hoặc xây dựng một hàm ngầm (implicit function) để mô tả hình dạng của đối tượng trong không gian 3D. Bề mặt của vật thể được xác định là tập hợp các điểm mà giá trị của hàm bằng 0 (zero-level set). Sau khi có trường hàm, thuật toán Marching Cubes thường được sử dụng để trích xuất bề mặt dưới dạng lưới tam giác từ thể tích.

a) Poisson Surface Reconstruction

Là một trong những phương pháp phổ biến nhất cho mesh reconstruction từ dữ liệu point cloud có thông tin vector pháp tuyến. Ý tưởng chính của thuật toán là chuyển việc tái tạo bề mặt thành một bài toán giải phương trình Poisson trong không gian 3 chiều. Cụ thể:

- Từ các điểm dữ liệu rời rạc và vector pháp tuyến tương ứng, thuật toán xây dựng một trường vector $\vec{V}$ sao cho mỗi vector hướng ra ngoài từ bề mặt thật.
- Giả định tồn tại một hàm vô hướng ẩn χ sao cho gradient của nó khớp với trường vector $\vec{V}$:

$$\nabla \chi \approx \vec{V}$$

- Bằng cách lấy divergence hai vế, ta có được bài toán:

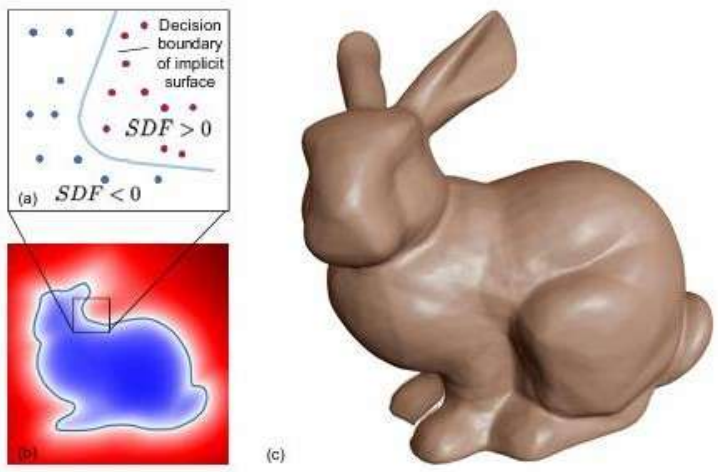
$$\Delta \chi = \nabla \cdot \vec{V}$$

Đây là phương trình Poisson – một bài toán elliptic có thể giải bằng các phương pháp số trên lưới voxel hóa. Thuật toán này cũng có hạn chế khi kết quả phụ thuộc nhiều vào độ chính xác của vector pháp tuyến đầu vào, nếu vector bị nhiễu thì hình dạng tái tạo có thể bị sai lệch đáng kể.

b) DeepSDF

DeepSDF (Park et al., 2019) [27] là một phương pháp dựa trên học sâu để mô hình hóa Signed Distance Function (SDF) – một dạng hàm ngầm mô tả bề mặt 3D. Trong DeepSDF, thay vì xây dựng SDF theo phương pháp truyền thống hoặc giải phương trình vi phân, mô hình sử dụng mạng neuron fully-connected để học trực tiếp một hàm ánh xạ từ tọa độ không gian $x \in \mathbb{R}^3$ sang giá trị khoảng cách có dấu $s \in \mathbb{R}$.

- Bề mặt của vật thể được biểu diễn bởi zero-level set của hàm SDF – tập hợp các điểm x sao cho $f(x) = 0$.
- Hàm SDF được học thông qua mạng neuron $f_0(x, z)$, trong đó z là vector đặc trưng đại diện cho một hình dạng cụ thể, và θ là tham số của mạng.



Hình 2.16: Minh họa hàm SDF

Phương pháp này cho phép biểu diễn bề mặt một cách mịn và liên tục, không yêu cầu mật độ điểm đều. Tuy nhiên độ chính xác phụ thuộc nhiều vào chất lượng dữ liệu huấn luyện, mô hình chỉ tổng quát tốt khi được học trên tập dữ liệu đủ đa dạng. Ngoài ra, việc đánh giá hàm SDF trên toàn bộ không gian 3D để trích xuất mesh có thể tiêu tốn thời gian tính toán.

c) GeoUDF

Khác với các phương pháp học SDF, GeoUDF (Xiang et al., 2023) [28] sử dụng Unsigned Distance Function (UDF) – hàm đo khoảng cách tuyệt đối đến bề mặt gần nhất, không phân biệt trong hay ngoài. Đây là ưu điểm quan trọng khi làm việc với dữ liệu không có nhãn hay thiếu vector pháp tuyến.

GeoUDF sử dụng một mạng neuron đơn giản Multi-Layer Perceptron (MLP) để học ánh xạ từ tọa độ không gian 3 chiều $x \in \mathbb{R}^3$ sang giá trị khoảng cách $d \in \mathbb{R}^+$, tức là:

$$f_\theta(x) \approx UDF(x)$$

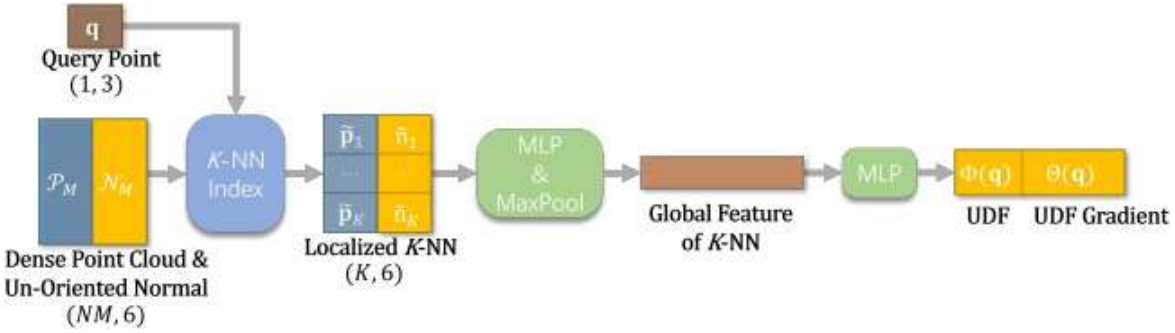
Trong đó θ là các tham số học được của mạng.

Điểm nổi bật của GeoUDF là thiết kế hàm mất mát với các thành phần hướng hình học, bao gồm:

- Surface consistency loss: đảm bảo giá trị UDF tiệm cận về 0 tại các điểm nằm trên bề mặt point cloud.
- Eikonal loss: khuyến khích gradient của UDF có độ lớn bằng 1, đảm bảo hàm khoảng cách được trơn và phù hợp về mặt hình học.

- Tangent constraint: sử dụng đặc trưng cục bộ của bề mặt từ các láng giềng gần nhất để đảm bảo bề mặt tái tạo có hướng tiếp tuyến phù hợp.

Sau khi học xong hàm UDF, GeoUDF sử dụng thuật toán Marching Cubes trên lưới voxel hóa để trích xuất zero-level set – tập hợp các điểm x sao cho $UDF(x) = 0$.



Hình 2.17: Quy trình hoạt động của GeoUDF

Ưu điểm lớn so với các phương pháp kể trên là GeoUDF không yêu cầu vector pháp tuyến hay thông tin bên trong-ngoài bề mặt, nhờ đó phù hợp với dữ liệu point cloud thực tế thường không đầy đủ hoặc bị nhiễu. Nhờ sử dụng UDF, mô hình có khả năng học hình dạng một cách ổn định ngay cả khi dữ liệu đầu vào thiếu nhãn hoặc thông tin hình học. GeoUDF sử dụng mạng MLP đơn giản với các ràng buộc hình học trong hàm mất mát, cho phép tái tạo bề mặt chi tiết, liên tục và mịn từ dữ liệu point cloud đầu vào.

2.8. Kỹ thuật làm mượt bề mặt mesh

Sau quá trình tái tạo mesh, đặc biệt từ dữ liệu point cloud nhiễu hoặc thưa thớt, bề mặt mesh có thể chứa nhiều đỉnh sắc nhọn, gợn sóng hoặc nhiễu hình học. Để cải thiện chất lượng hình dạng vật thể, các thuật toán làm mượt (mesh smoothing) được áp dụng nhằm giảm thiểu hình học và tái tạo bề mặt mịn hơn, đồng thời cố gắng bảo toàn các đặc trưng hình học quan trọng.

2.8.1. Laplacian Smoothing

Đây là phương pháp đơn giản và phổ biến nhất. Ý tưởng cơ bản là mỗi đỉnh v_i trên mesh sẽ được cập nhật bằng trung bình vị trí các đỉnh lân cận (1-ring neighbors):

$$v_i' = v_i + \lambda \cdot \sum_{j \in N(i)} (v_j - v_i)$$

Trong đó:

- λ là hệ số điều chỉnh (thường rất nhỏ, ví dụ 0.1)
- $N(i)$ là tập đỉnh kề với v_i

Phương pháp này dễ cài đặt và tính toán nhanh, nhưng nếu lặp nhiều lần mà không có cơ chế bảo toàn hình dạng, mô hình sẽ bị co rút, mất thể tích hoặc chi tiết gốc.

2.8.2. Taubin Smoothing

Taubin Smoothing được đề xuất bởi Gabriel Taubin (1995) nhằm khắc phục hiện tượng co rút của Laplacian smoothing. Phương pháp thực hiện hai bước làm mượt liên tiếp với hai hệ số trái dấu:

- Làm mượt với hệ số λ dương để di chuyển đỉnh.
- Làm mượt tiếp với hệ số μ âm để khôi phục lại thể tích đã mất.

Việc kết hợp hai bước này giúp duy trì hình dạng và thể tích tốt hơn, đặc biệt hữu ích trong các mô hình có chi tiết cong tròn hoặc cấu trúc phức tạp.

2.8.3. Bilateral Mesh Denoising

Đây là phương pháp làm mượt có trọng số, giúp bảo toàn các chi tiết hình học sắc cạnh hoặc góc nhọn. Thuật toán mở rộng từ bilateral filter trong xử lý ảnh sang mô hình 3D. Với mỗi đỉnh, trọng số các đỉnh lân cận được tính dựa trên:

- Khoảng cách Euclidean trong không gian 3D.
- Độ tương đồng giữa các vector pháp tuyến, giúp phát hiện biên hoặc bề mặt phẳng.

Hàm làm mượt kết hợp hai loại trọng số không gian và pháp tuyến, cho phép khử nhiễu hiệu quả mà vẫn giữ rõ các chi tiết đặc trưng, tránh hiện tượng mờ nhòe như ở các phương pháp đơn giản hơn.

2.8.4. HC Laplacian Smoothing

Phương pháp này được cải tiến từ Laplacian smoothing bằng cách thêm cơ chế hồi phục hình dạng gốc. Quy trình gồm hai bước:

- Làm mượt như thuật toán Laplacian, đưa đỉnh đến vị trí mới v'_i
- Phục hồi lại hình dạng bằng cách pha trộn giữa vị trí cũ và mới để giữ lại đặc trưng ban đầu.

Điều này giúp làm mượt các vùng phẳng hoặc cong mịn mà vẫn duy trì hình dáng tổng thể và hạn chế biến dạng. Phương pháp này thích hợp cho các ứng dụng cần bảo tồn chi tiết trong quá trình tái tạo mesh.

CHƯƠNG 3: GIẢI PHÁP ĐỀ XUẤT

3.1. Tổng quan giải pháp

3.1.1. Phát biểu bài toán

Tái tạo vật thể 3D từ dữ liệu 2D là một bài toán quan trọng lĩnh vực thị giác máy tính và đồ họa máy tính, với nhiều ứng dụng thực tiễn như thực tế ảo, thực tế tăng cường (VR/AR), in 3D, nhận diện đối tượng,… Bài toán đặt ra yêu cầu xây dựng một hệ thống có khả năng học và suy luận về cấu trúc không gian ba chiều của một vật thể, chỉ từ một hoặc vài hình ảnh đầu vào 2D.

Trong khuôn khổ phạm vi đồ án này, hệ thống sẽ nhận đầu vào là một ảnh đơn (single-view image), và đầu ra mong muốn là một biểu diễn 3D của vật thể tương ứng. Cụ thể, hệ thống hướng đến hai hình thức biểu diễn đầu ra chính là **point cloud** và **mesh**. Việc lựa chọn hai dạng biểu diễn này được đưa ra dựa trên cả yếu tố chủ quan về kỹ thuật lẫn yếu tố khách quan từ nhu cầu sử dụng thực tế.

Như đã đề cập ở chương 1, point cloud là tập hợp các điểm rời rạc trong không gian 3 chiều, đóng vai trò như một biểu diễn thô nhưng giàu thông tin về hình dạng tổng thể của vật thể. Đây là dạng biểu diễn đơn giản, dễ xử lý, và rất phù hợp với các mô hình học sâu hiện nay do tính chất rời rạc và linh hoạt. Nhiều nghiên cứu gần đây đã khai thác trực tiếp không gian point cloud nhằm tận dụng khả năng tổng quát hóa và biểu diễn linh hoạt, giúp đánh giá hiệu quả mô hình trong giai đoạn đầu của pipeline.

Ngược lại, mesh là dạng biểu diễn có cấu trúc cao hơn, trong đó các điểm được kết nối lại bằng các mặt tam giác để tạo thành một bề mặt liền mạch, trực quan hơn. Dạng biểu diễn này thường được ưa chuộng trong các ứng dụng thực tế, nơi người dùng cuối mong muốn có được mô hình 3D hoàn chỉnh, dễ tích hợp vào phần mềm đồ họa hoặc các môi trường mô phỏng.

3.1.2. Tổng quan giải pháp hệ thống

Thay vì xây dựng một mô hình end-to-end dự đoán trực tiếp mesh từ ảnh, tôi đề xuất thiết kế hệ thống theo hướng hai giai đoạn: đầu tiên dự đoán point cloud từ ảnh đầu vào bằng mô hình học sâu, sau đó thực hiện hậu xử lý để chuyển đổi point cloud thành mesh.

Lý do chính cho thiết kế này là mesh là dạng biểu diễn có cấu trúc phức tạp, yêu cầu kiến trúc mạng chuyên biệt và dữ liệu huấn luyện phong phú, điều này gây khó khăn trong việc huấn luyện mô hình ổn định và chính xác. Ngược lại, point cloud là biểu diễn

đơn giản hơn, phù hợp với các kiến trúc học sâu hiện tại và dễ đánh giá, từ đó giúp quá trình huấn luyện và kiểm thử ban đầu hiệu quả hơn. Bên cạnh đó, việc tách riêng hai giai đoạn mang lại nhiều lợi ích:

- Cho phép tận dụng ưu điểm riêng của từng dạng biểu diễn.
- Tăng tính linh hoạt trong điều chỉnh, kiểm thử và thay thế từng thành phần.
- Dễ dàng tích hợp các thuật toán hậu xử lý tiên tiến mà không cần huấn luyện lại toàn bộ mô hình.
- Giảm độ phức tạp của mô hình, giúp hệ thống dễ bảo trì và mở rộng trong tương lai.

Hệ thống được chia làm 3 thành phần chính:

- **Tiền xử lý và trích xuất đặc trưng**: Ảnh RGB đầu vào được resize về kích thước 128×128. Sau đó, depth map được trích xuất và ghép nối (concat) với ảnh RGB để tạo ảnh RGB-D. Ảnh này được chuyển thành tensor làm đầu vào cho mô hình.
- **Mô hình dự đoán point cloud**: Sử dụng lại kiến trúc Pixel2Point, nhưng được điều chỉnh để tiếp nhận ảnh RGB-D thay vì chỉ ảnh RGB. Mô hình đầu ra là tập hợp các điểm trong không gian 3D đại diện cho hình dạng vật thể.Hậu xử lý
- **Hậu xử lý**: Gồm các bước upsampling để tăng mật độ point cloud, loại bỏ điểm nhiễu (outlier removal), và cuối cùng là tái tạo mesh từ point cloud thông qua các thuật toán dựng hình phù hợp.

3.2. Chuẩn bị dữ liệu

Để huấn luyện và đánh giá hệ thống, tôi lựa chọn sử dụng bộ dữ liệu ShapeNetCore, một bộ dữ liệu lớn và được sử dụng rộng rãi trong các nghiên cứu về tái tạo hình học và nhận diện vật thể 3D.

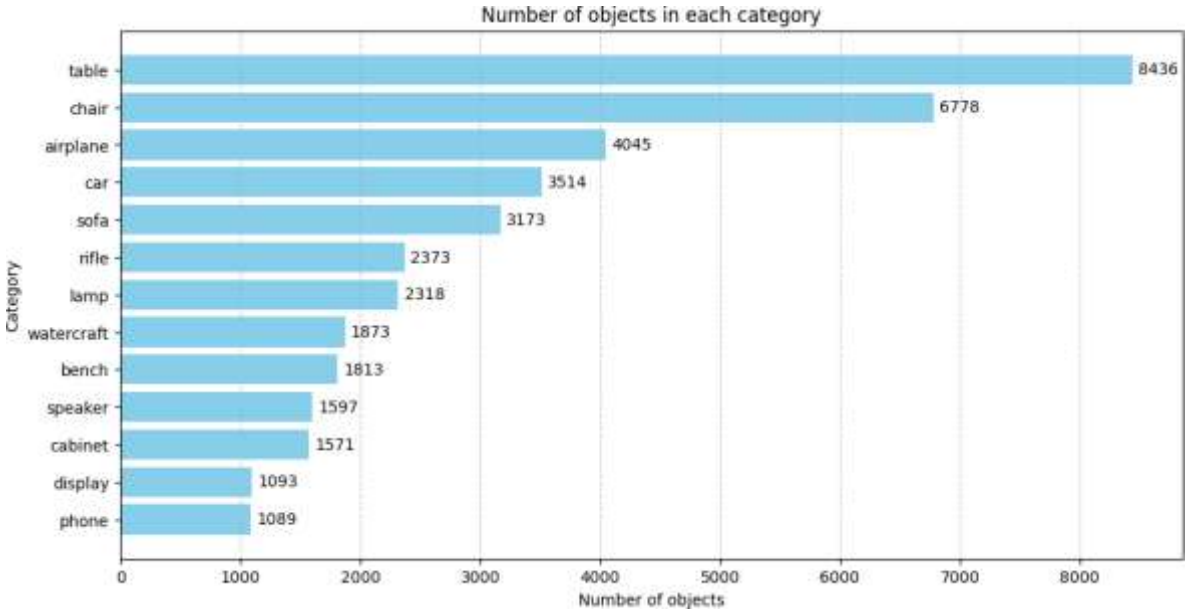
Tuy nhiên, một hạn chế đáng kể của ShapeNetCore là không cung cấp sẵn ảnh RGB 2D được render từ các mô hình này. Trong khi đó, hệ thống yêu cầu ảnh đầu vào ở định dạng RGB làm cơ sở để dự đoán mô hình vật thể 3D. Việc tự render ảnh từ các mô hình ShapeNet sẽ cần xây dựng pipeline riêng cho việc dựng hình, đặt camera, ánh sáng,… Điều này vừa tốn thời gian, vừa đòi hỏi tài nguyên tính toán lớn và dễ phát sinh sai lệch nếu không đảm bảo tính nhất quán giữa ảnh và mô hình gốc.

Để giải quyết vấn đề này, tôi quyết định sử dụng bộ dữ liệu đã được xử lý sẵn từ ShapeNetCore, được xây dựng và công bố bởi Xu et al. [21]. Bộ dữ liệu này cung cấp ảnh RGB được render từ nhiều góc nhìn khác nhau quanh vật thể.

Mỗi vật thể trong bộ dữ liệu được render dưới 72 góc nhìn khác nhau, trong đó bao gồm 36 góc chụp dễ (easy views) – thường là các góc phổ biến như mặt trước, bên hông, từ trên xuống, và 36 góc chụp khó (hard views) – thường là các góc khuất hoặc ít thông tin hình học. Mỗi ảnh có kích thước ban đầu 224×224.

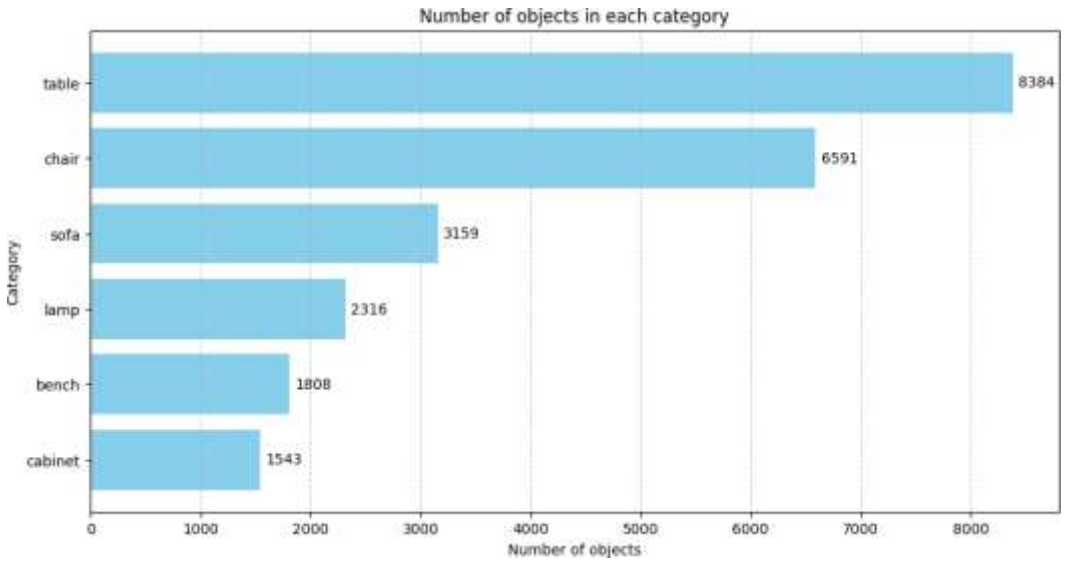
Hình 3.1: Các ảnh với góc chụp khác nhau của một vật thể

Bộ dữ liệu gồm tổng cộng 13 lớp với số lượng phân bố mẫu ban đầu như sau:



Hình 3.2: Thống kê số lượng mẫu ban đầu của các lớp

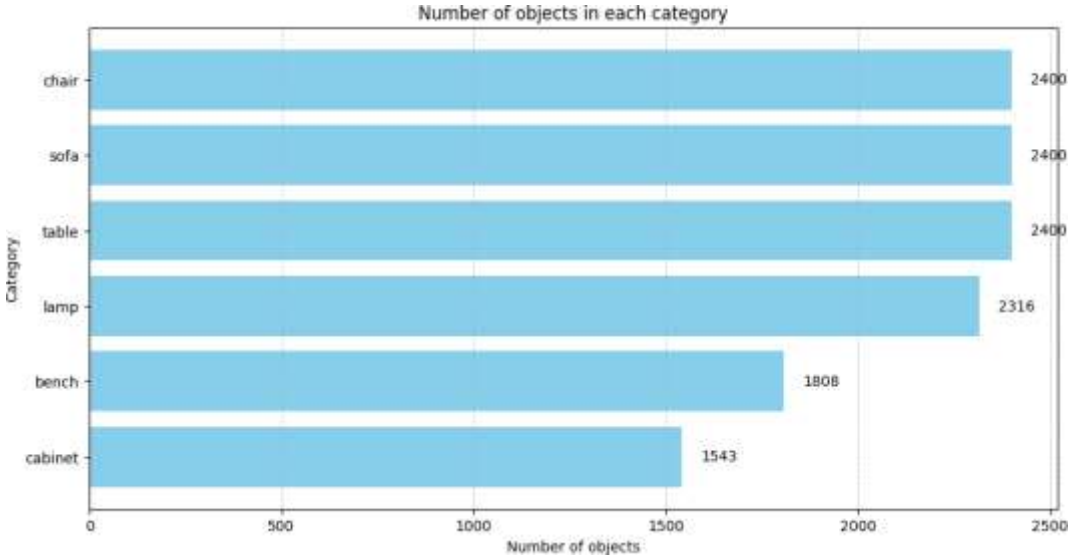
Trong khuôn khổ đồ án này, các đối tượng được lựa chọn tập trung vào nhóm vật thể nội thất (indoor), bao gồm 6 lớp: *table, sofa, bench, chair, cabinet, lamp*.



Hình 3.3: Thống kê số lượng mẫu ban đầu của 6 lớp đã chọn

Do sự chênh lệch đáng kể về số lượng mẫu giữa các lớp, việc giữ nguyên phân bố ban đầu có thể dẫn đến hiện tượng mất cân bằng dữ liệu. Điều này dễ khiến mô hình bị thiên lệch trong quá trình huấn luyện, tập trung quá mức vào các lớp phổ biến và bỏ qua các lớp có ít dữ liệu hơn. Hệ quả là khả năng tổng quát hóa bị suy giảm, đặc biệt trong bối cảnh bài toán yêu cầu mô hình nhận diện hiệu quả nhiều loại vật thể khác nhau.

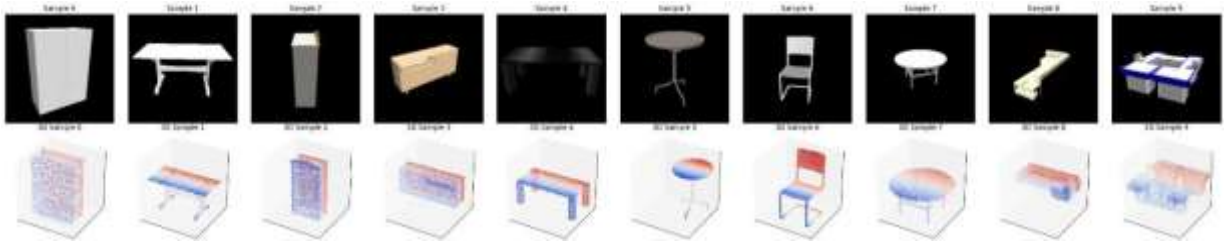
Để khắc phục vấn đề này, các lớp có số lượng mẫu vượt trội được tiến hành downsample, nhằm cân bằng lại phân bố dữ liệu giữa các lớp. Cách tiếp cận này không chỉ giúp mô hình học được đặc trưng hình học một cách đồng đều, giảm thiểu hiện tượng overfitting vào các lớp chiếm ưu thế, mà còn góp phần rút ngắn thời gian huấn luyện, phù hợp với điều kiện tài nguyên tính toán có hạn.



Hình 3.4: Thống kê số lượng mẫu của 6 lớp sau khi đã downsample

Sau khi downsample, bộ dữ liệu được chia thành 3 phần: tập huấn luyện (train set), tập kiểm định (validation set), tập kiểm thử (test set) theo tỉ lệ 80:10:10. Việc chia dữ liệu được thực hiện stratified sampling theo từng lớp nhằm đảm bảo tỉ lệ mẫu của mỗi lớp trong từng tập con phản ánh đúng phân bố của toàn bộ tập dữ liệu.

Tiếp theo là xây dựng bộ dữ liệu hoàn chỉnh bằng cách gắn mỗi ảnh RGB với point cloud tương ứng của vật thể, dựa trên object ID chung giữa hai nguồn dữ liệu.



Hình 3.5: Các cặp ảnh RGB với point cloud ground truth tương ứng

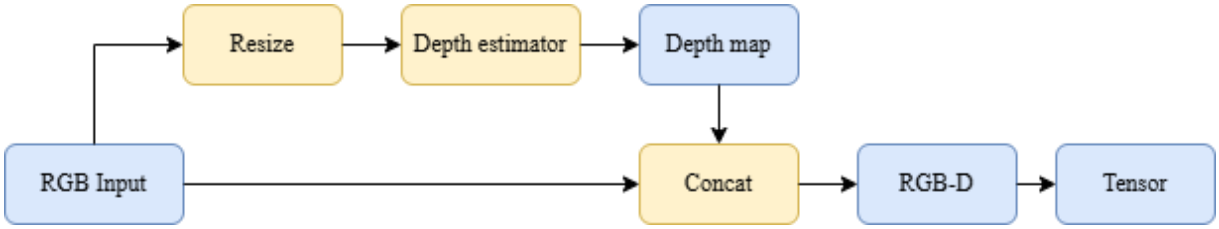
Mỗi obejct ban đầu được render với 36 ảnh từ easy views. Tuy nhiên trong thực tế, mô hình không cần đến toàn bộ số ảnh này để có thể trích xuất được các đặc trưng hình học cần thiết. Do đó, 24 ảnh được chọn ngẫu nhiên từ 36 ảnh easy views nhằm giảm thời gian xử lý khi xây dựng tập dữ liệu, đồng thời vẫn giữ được sự đa dạng góc nhìn cần thiết. Phương pháp này giúp tối ưu hiệu suất huấn luyện và khuyến khích mô hình học được các biểu diễn không gian tổng quát thay vì phụ thuộc vào số lượng lớn ảnh.

Cuối cùng tôi compile các sample đã xử lý (bao gồm ảnh RGB, point cloud tương ứng, object ID và nhãn lớp) vào các file .h5 riêng biệt cho từng lớp, được tổ chức theo ba tập train, validation và test.

Hình 3.6: Bộ dữ liệu sau khi đã compile thành file .h5

3.3. Tiền xử lý và trích xuất đặc trưng

Để chuẩn bị dữ liệu đầu vào cho mô hình học sâu, toàn bộ ảnh RGB được xử lý qua một chuỗi các bước tiền xử lý nhằm tích hợp thông tin chiều sâu vào biểu diễn ảnh. Quy trình hoạt động được mô tả trực quan qua sơ đồ dưới đây:



Hình 3.7: Sơ đồ hoạt động của bước tiền xử lý và trích xuất đặc trưng

Việc sử dụng bản đồ độ sâu (depth map) kết hợp với ảnh RGB để tạo ra RGB-D thay vì chỉ sử dụng ảnh RGB thuần túy làm đầu vào đã được chứng minh là mang lại hiệu quả cao trong các bài toán liên quan đến nhận diện và tái tạo hình học 3D. Giải pháp này được đề xuất bởi Yousra Shleibik [22], trong đó tác giả chỉ ra rằng ảnh RGB-D giúp mô hình khai thác tốt hơn các đặc trưng không gian của vật thể, đặc biệt trong các góc nhìn phức tạp hoặc ít thông tin thị giác.

Bản đồ độ sâu cung cấp thông tin về khoảng cách từ từng điểm ảnh đến camera, từ đó mô hình có thể hiểu rõ hơn về cấu trúc hình học ba chiều tiềm ẩn trong ảnh hai chiều. Khi kết hợp với đặc trưng màu sắc và kết cấu từ kênh RGB, ảnh RGB-D trở thành đầu vào giàu thông tin, hỗ trợ quá trình trích xuất đặc trưng một cách đầy đủ và chính xác hơn.

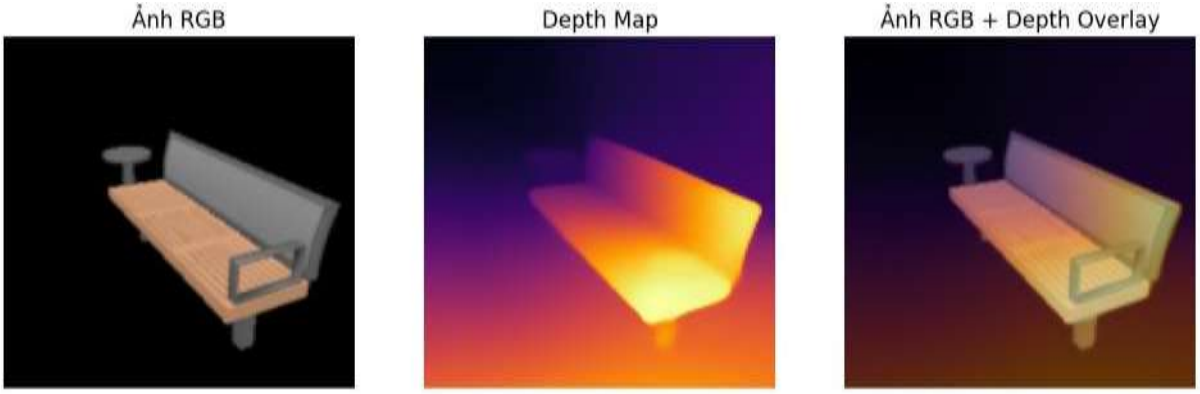
Depth estimator sử dụng pretrained model để dự đoán độ sâu cho các ảnh RGB. Mô hình được sử dụng là MiDaS, do Intel Intelligent Systems Lab phát triển. Đây là một trong những mô hình hàng đầu hiện nay trong lĩnh vực monocular depth estimation (ước lượng độ sâu từ ảnh đơn), với khả năng hoạt động tốt trên nhiều ngữ cảnh hình ảnh khác nhau mà không cần thêm thông tin từ camera hoặc các cảm biến bổ sung.

MiDaS hiện hỗ trợ 3 phiên bản chính: DPT_Large, DPT_Hybrid và MiDaS_small. Trong đó:

- **DPT_Large** là phiên bản có kiến trúc lớn nhất, cho ra bản đồ độ sâu với độ chi tiết và độ chính xác cao nhất thích hợp với các tác vụ yêu cầu độ phân giải không gian cao. Tuy nhiên, tốc độ inference chậm hơn, do độ phức tạp của mô hình.
- **DPT_Hybrid** là lựa chọn cân bằng giữa tốc độ và chất lượng, kết hợp giữa kiến trúc của DPT_Large và các đặc trưng và các đặc trưng nhẹ hơn từ các backbone khác như ResNet.
- **MiDaS_small** là phiên bản gọn nhẹ, phù hợp với các ứng dụng real-time hoặc trên thiết bị tài nguyên hạn chế nhưng có độ chính xác thấp nhất so với các phiên bản trên.

Do đó phiên bản DPT_Hybrid được lựa chọn nhằm đảm bảo sự cân bằng giữa hiệu suất tính toán và chất lượng độ sâu đầu ra.

Ảnh đầu vào RGB đầu tiên sẽ được resize về kích thước 128×128, sau đó sẽ đi qua Depth Estimator để trích xuất được bản đồ độ sâu tương ứng cùng kích thước. Bản đồ độ sâu được chuẩn hóa về khoảng giá trị [0, 1] và được ghép nối (concat) cùng với ảnh RGB theo chiều kênh, tạo thành một tensor 4 kênh (R, G, B, D).



Hình 3.8: Ảnh RGB, bản đồ độ sâu, và ảnh RGB-D

Tensor RGB-D thu được sau bước này là dạng đầu vào hoàn chỉnh, kết hợp giữa thông tin màu sắc và hình học không gian, sẵn sàng để đưa vào huấn luyện mô hình học sâu.

3.4. Huấn luyện mô hình

Kiến trúc mô hình huấn luyện được xây dựng dựa trên nền tảng của mô hình Pixel2Point, một mạng CNN được thiết kế cho tác vụ ánh xạ từ ảnh 2D sang point cloud 3D. Tuy nhiên kiến trúc gốc của Pixel2Point được thiết kế để xử lý ảnh RGB (3 kênh) làm đầu vào. Trong hệ thống hiện tại, ảnh đầu vào đã mở rộng thành dạng RGB-D (4 kênh), vì vậy kiến trúc ban đầu đã được điều chỉnh để tương thích với đầu vào mới. Phần còn lại của kiến trúc CNN vẫn được giữ nguyên, kế thừa từ thiết kế gốc của Pixel2Point, với nhiều tầng tích chập và các tầng fully connected.

Quá trình huấn luyện mô hình được thực hiện trên Google Colab với lựa chọn GPU NVIDIA L4. So với các GPU phổ biến như T4 hay P4, GPU L4 có bộ nhớ VRAM lớn hơn (24 GB) và khả năng xử lý nhanh hơn, từ đó hỗ trợ quá trình huấn luyện mô hình hiệu quả hơn.

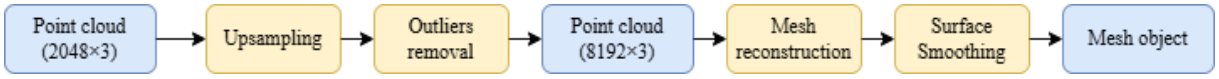
Để đảm bảo mô hình huấn luyện hiệu quả, các tham số huấn luyện quan trọng như learning rate, batch size, số epoch và các giá trị liên quan khác được trình bày trong bảng thông số dưới đây.

Bảng 3.1: Các tham số huấn luyện mô hình

Thông tin	Giá trị
Input shape	(4, 128, 128)
Batch size	16
Số epoch	15
Learning rate	0.00005
Optimizer	Adam
Loss function	Chamfer Distance
Gradient Scaler	AMP (Automatic Mixed Precision)
GPU	NVIDIA L4
Visualization	TensorBoard + Plotly

3.5. Hậu xử lý point cloud

Để thu được vật thể 3D với bề mặt hoàn chỉnh, có tính thẩm mĩ và có tính ứng dụng thực tế được, dữ liệu point cloud đầu ra từ mô hình học sâu cần trải qua các bước hậu xử lý. Quy trình hậu xử lý được mô tả như trong sơ đồ dưới đây:



Hình 3.9: Sơ đồ hoạt động của bước hậu xử lý point cloud

3.5.1. Upsampling

Point cloud đầu ra từ mô hình học sâu với $N = 2048$ điểm mặc dù được cho là vừa đủ để che phủ được bề mặt của vật thể, nhưng về kết cấu chung thì vẫn khá thưa. Do đó bước Upsampling là một bước quan trọng trong pipeline hậu xử lý nhằm đảm bảo point cloud trước khi đi qua mô hình tái tạo mesh có đủ độ dày và độ mịn cần thiết, giúp vật thể mesh đầu cuối được hoàn chỉnh hơn.

Ở bước này tôi sử dụng mô hình pretrained sử dụng kiến trúc SAPCU (Self-Supervised Arbitrary-Scale Point Cloud Upsampling) được cung cấp bởi chính tác giả Zhao [25]. Như đã đề cập ở chương 2, kiến trúc này có khả năng thực hiện upsampling ở các tỉ lệ tùy ý (arbitrary scale) mà không cần dữ liệu ground truth dense. Thay vì học trực tiếp mapping từ dữ liệu mật độ thưa sang dữ liệu mật độ dày, SAPCU tiếp cần bài toán dưới góc nhìn hình học: mỗi điểm được upsample được xem là kết quả chiếu của một seed point lên một bề mặt ngầm đã học.

Tôi cấu hình thực hiện upsampling từ 2048 điểm lên 8192 điểm, cho kết quả điểm dày và phân bố đều hơn. Các tham số như tỷ lệ scale và phương thức chọn điểm seed đều được giữ nguyên như thiết lập mặc định của tác giả, nhằm đảm bảo kết quả nhất quán với công bố gốc.

3.5.2. Outliers removal

Sau khi thực hiện upsampling, kết quả point cloud có thể vẫn chứa một số điểm nhiễu, những điểm nằm rải rác hoặc cách xa khỏi bề mặt chính của vật thể. Để loại bỏ nhiễu, tôi áp dụng k-nearest neighbors (k-NN). Cụ thể, với mỗi điểm trong point cloud, tôi tính khoảng cách trung bình đến 30 điểm lân cận gần nhất. Sau đó tính khoảng cách trung bình toàn cục của toàn bộ point cloud. Những điểm nào có khoảng cách trung bình lớn hơn 1.5 lần khoảng cách trung bình toàn cục sẽ được xem là nhiễu và bị loại bỏ.

Tuy nhiên, sau bước loại bỏ nhiễu, số lượng điểm còn lại không còn cố định — có thể ít hơn hoặc nhiều hơn mục tiêu 8192 điểm. Để đưa dữ liệu về đúng kích thước đầu ra yêu cầu là 8192 điểm, tôi áp dụng thuật toán Farthest Point Sampling (FPS) lên tập point cloud sau khi đã được đưa ngược trở lại không gian ban đầu. Cụ thể, từng điểm được nhân lại với hệ số scale và cộng lại với vector dịch chuyển loc (đã được tính trong bước chuẩn hóa trước khi đưa vào mô hình), nhằm phục hồi đúng vị trí và tỷ lệ ban đầu của vật thể.

FPS sau đó được sử dụng để chọn ra đúng 8192 điểm sao cho các điểm được lấy mẫu cách xa nhau nhất trong không gian 3D. Điều này không chỉ đảm bảo phân bố đều khắp bề mặt vật thể, mà còn giúp giữ lại các đặc trưng hình học quan trọng. Nhờ đó, point cloud đầu ra vừa được làm sạch nhiễu, vừa có mật độ điểm ổn định và phân bố tốt, phù hợp cho bước tái tạo mesh kế tiếp.

3.5.3. Mesh reconstruction

Bước tiếp theo trong pipeline hậu xử lý là mesh reconstruction, tái tạo lưới tam giác từ dữ liệu điểm point cloud rời rạc. Tôi sử dụng mô hình GeoUDF, kèm theo trọng số pretrained được cung cấp bởi tác giả Xiang et al. [28].

Như đã đề cập ở chương 2, GeoUDF tiếp cận bài toán tái tạo bề mặt dưới dạng hồi quy một trường khoảng cách vô hướng (unsigned distance field - UDF), trong đó mỗi điểm trong không gian 3D được ánh xạ đến khoảng cách gần nhất đến bề mặt vật thể, không phân biệt bên trong hay bên ngoài. Đây là ưu điểm lớn trong trường hợp không có nhãn ground truth hoặc thiếu vector pháp tuyến – một đặc điểm phổ biến của dữ liệu point cloud thu từ ảnh RGB.

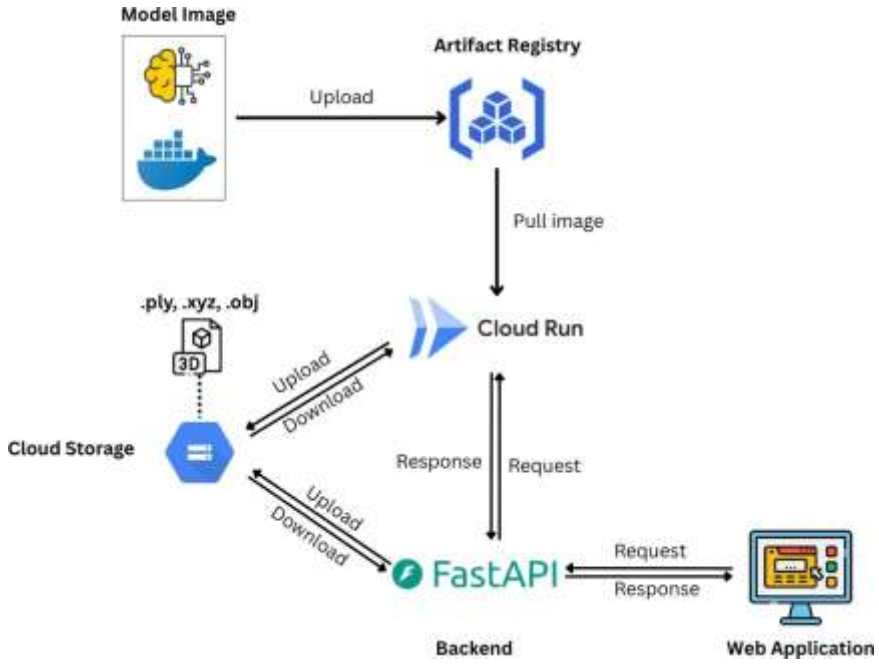
Cụ thể, point cloud (8192×3) sau khi đã được upsample và làm sạch được đưa qua mô hình GeoUDF để infer trường khoảng cách vô hướng UDF ẩn trong không gian 3D. Mô hình cho đầu ra là một trường giá trị liên tục, trong đó mỗi điểm trong không gian được ánh xạ đến khoảng cách gần nhất tới bề mặt vật thể.

Để chuyển đổi trường giá trị này thành mesh tam giác, thuật toán Marching Cubes – một phương pháp phổ biến nhằm trích xuất isosurface từ trường vô hướng được áp dụng. Kết quả thu được là một mô hình vật thể mesh được lưu định dạng .ply đại diện cho hình dạng 3D liền mạch của vật thể.

3.6. Xây dựng hệ thống

3.6.1. Tổng quan kiến trúc hệ thống

Hệ thống gồm các thành phần chính: Web Application, Backend FastAPI, Google Cloud Run, và các dịch vụ hỗ trợ khác của Google Cloud như Artifact Registry và Cloud Storage.



Hình 3.10: Sơ đồ kiến trúc của hệ thống tái tạo vật thể 3D

- **Web Application**: Giao diện frontend cho phép người dùng tải lên ảnh đầu vào và nhận kết quả 3D. Ứng dụng gửi yêu cầu tới backend và hiển thị kết quả trả về.
- **FastAPI Backend**: Đóng vai trò trung gian xử lý logic ứng dụng. Backend nhận request từ frontend, thực hiện các tác vụ trao đổi file với Cloud Storage, và gửi yêu cầu tới mô hình inference thông qua Cloud Run.

- **Cloud Run**: Triển khai các mô hình học sâu trong container. Khi được gọi, Cloud Run tự động pull image từ Artifact Registry và thực hiện xử lý inference với dữ liệu được lấy từ Cloud Storage, sau đó trả kết quả về server FastAPI.
- **Artifact Registry**: Lưu trữ các Docker image chứa mô hình. Cloud Run sẽ truy xuất image tại đây mỗi khi khởi chạy hoặc được cập nhật.
- **Cloud Storage**: Lưu trữ dữ liệu input (ảnh, point cloud) và output (.ply, .xyz, .obj). Cả Cloud Run và FastAPI đều tương tác hai chiều với Storage để upload/download dữ liệu phục vụ cho pipeline xử lý.

3.6.2. Thiết lập Google Cloud Platform

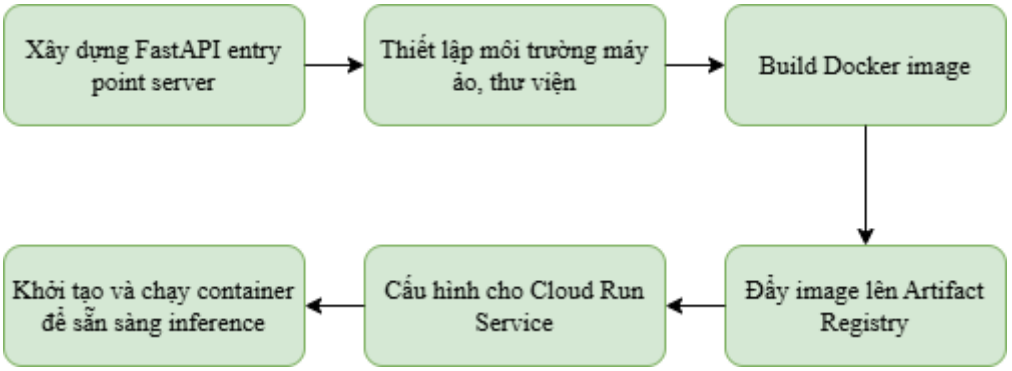
Với việc pipeline xử lý của hệ thống gồm nhiều mô hình học sâu trong các bước, việc lựa chọn một nền tảng dịch vụ có hỗ trợ GPU để host mô hình là một công việc quan trọng. Trong số các nền tảng hiện có, tôi lựa chọn Google Cloud Platform (GCP).

GCP là một nền tảng điện toán đám mây do Google cung cấp, với hệ sinh thái dịch vụ đa dạng, bao gồm cả hạ tầng phần cứng mạnh mẽ và các công cụ hỗ trợ triển khai các mô hình học máy quy mô lớn. Việc lựa chọn GCP giúp tận dụng được tài nguyên GPU để xử lý các bước nặng tính toán trong pipeline xử lý, đặc biệt là các mô hình học sâu như SAPCU, GeoUDF và mô hình dự đoán point cloud.

Với gói trial miễn phí được Google cung cấp, và $300 credit sử dụng trong vòng 90 ngày, đủ để triển khai thử nghiệm toàn bộ hệ thống. Ngoài khả năng tạo máy ảo GPU qua Compute Engine, GCP còn hỗ trợ nhiều dịch vụ hữu ích trong quá trình phát triển như:

- Vertex AI: quản lý, huấn luyện và triển khai mô hình.
- Cloud Storage Bucket: lưu trữ dữ liệu point cloud, mesh của vật thể và mô hình đã huấn luyện.
- Artifact Registry: quản lý và lưu trữ các Docker image chứa mô hình, thuận tiện cho việc triển khai container.
- Cloud Functions: được sử dụng để triển khai các container model. Trong hệ thống này, Cloud Functions đóng vai trò như entry point, sau khi khởi chạy container chứa mô hình học sâu, nhận request từ client để thực hiện inference và trả kết quả về.

Với mỗi mô hình học sâu, tôi thiết lập xây dựng và cấu hình cho Cloud Functions để tạo ra các entry point tương ứng với từng mô hình. Quy trình này được mô tả như trong sơ đồ dưới đây:



Hình 3.11: Quy trình thiết lập và cấu hình với Cloud Run Service

Cloud Run Service được cấu hình như sau:

- Region: asia-southeast1 (Singapore), khu vực này có hỗ trợ VM GPU.
- Authentication: yêu cầu xác thực đối với mỗi request đến endpoint, chỉ chấp nhận các request có credentials của tài khoản chính hoặc service account.
- Billing: theo instance-based, tức là sẽ tính phí trong suốt vòng đời hoạt động của instance này, khi sử dụng VM GPU thì yêu cầu này là bắt buộc.
- Service scaling: auto scale về 0 instance.
- Tài nguyên VM: đối với GPU thì yêu cầu memory 16GiB, 4v CPUs. GPU sử dụng NVIDIA L4 với số lượng là 1 GPU.
- Request timeout: mặc định 300 giây,
- Concurrency: cho phép tối đa 80 request đồng thời trên mỗi instance.

Ngoài ra Cloud Run Service có cơ chế health check theo hai hình thức là startup check và liveness check nhằm kiểm tra xem container đã được khởi chạy ổn định chưa. Tôi lựa chọn startup check vì nó chỉ kiểm tra lần đầu khi bắt đầu chạy service, còn liveness check thì liên tục gửi request để kiểm tra do đó gây tiêu hao tài nguyên không cần thiết.

Name ↑	Deployment type	Req/sec	Region	Authentication	Ingress
depth2point	Container	0	asia-southeast1	Require authentication	All
fastapi-mesh	Container	0	asia-southeast1	Require authentication	All
fastapi-upsampling	Container	0	asia-southeast1	Require authentication	All

Hình 3.12: Các entry point của mô hình sau khi triển khai trên Cloud Run Service

3.6.3. Xây dựng ứng dụng web

Để cung cấp một giao diện trực quan cho người dùng tương tác với hệ thống, tôi xây dựng một ứng dụng web tích hợp frontend và backend. Hệ thống web này không chỉ đảm nhận vai trò hiển thị mô hình 3D mà còn hỗ trợ toàn bộ pipeline xử lý – từ việc

nhận ảnh đầu vào, xử lý qua các mô hình học sâu, đến tái tạo mesh và hiển thị kết quả trực tiếp trên trình duyệt.

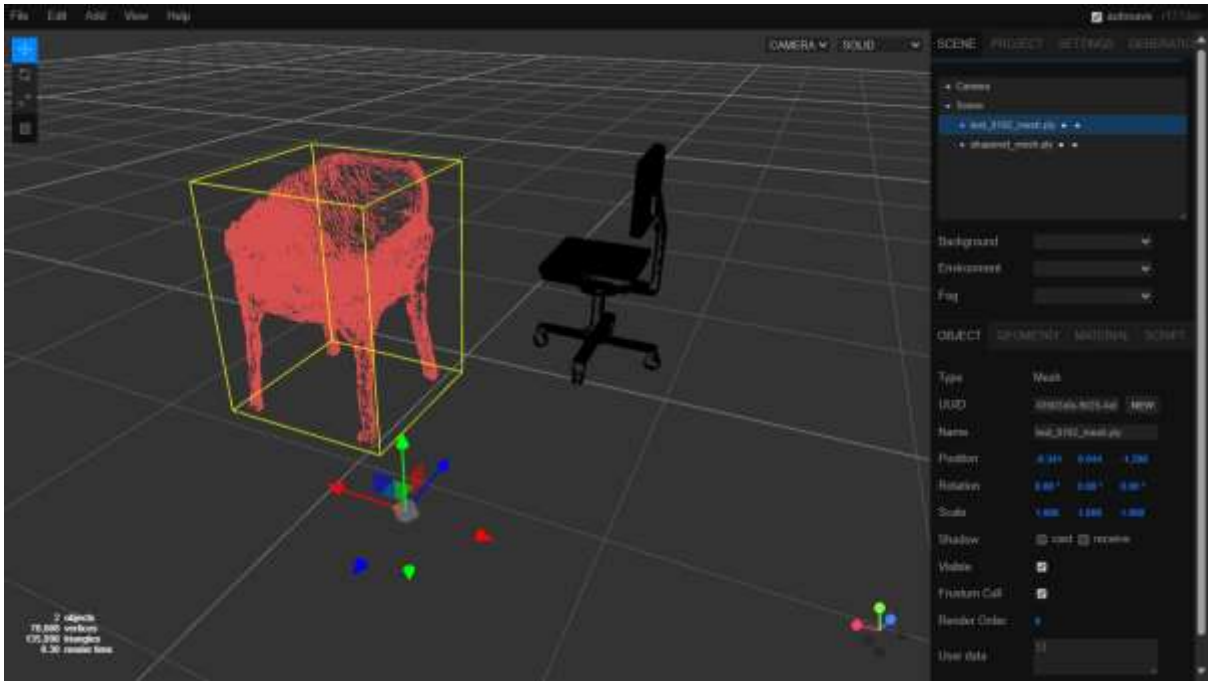
Frontend được xây dựng dựa trên mã nguồn mở của Three.js Editor, là một công cụ dựng sẵn được cung cấp trong bộ thư viện Three.js. Three.js Editor cung cấp các chức năng chỉnh sửa mô hình 3D cơ bản như: xoay, phóng to, thu nhỏ, thay đổi ánh sáng, hiển thị mesh hoặc point cloud, cũng như hỗ trợ import/export file các định dạng phổ biến như .ply, .obj, .glb,…

Tôi mở rộng Three.js Editor để tích hợp thêm các chức năng đặc thù của hệ thống:

- Giao diện upload ảnh 2D từ phía người dùng.
- Trình hiển thị kết quả point cloud và mesh tái tạo từ ảnh.
- Menu lựa chọn các thuật toán smoothing (như Laplacian Smoothing, Taubin Smoothing) để cải thiện độ mượt bề mặt.

Backend được xây dựng với FastAPI – một framework hiện đại, nhẹ và có hiệu năng cao dành cho các ứng dụng web nói chung và các xử lý liên quan đến học máy, học sâu. Backend gồm các nhiệm vụ chính sau:

- Nhận ảnh đầu vào từ phía người dùng.
- Tiền xử lý dữ liệu và trích xuất bản đồ độ sâu thành ảnh RGB-D.
- Gửi ảnh đến endpoint của Cloud Run tương ứng với mô hình học sâu.
- Xử lý kết quả inference trả về từ mô hình, bao gồm point cloud hoặc mesh của vật thể.
- Chuyển đổi dữ liệu thành các file .xyz, .ply để upload và lưu trên Bucket của Google Cloud Storage.
- Cung cấp API cho phép frontend export file theo các định dạng mong muốn (.xyz, .ply, .obj,…) để sử dụng ở các phần mềm đồ họa chuyên nghiệp hơn.

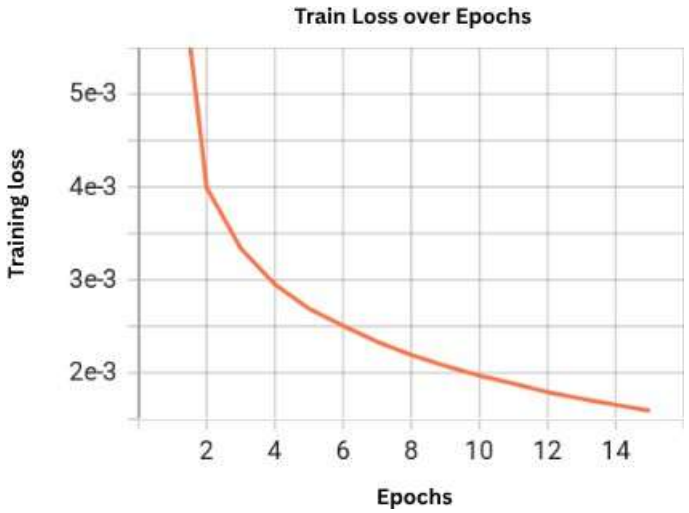


Hình 3.13: Giao diện của ứng dụng editor 3D

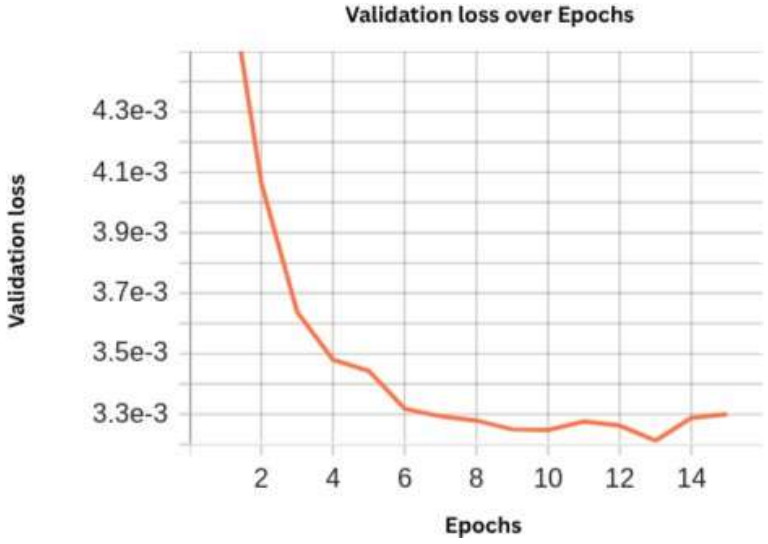
CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ

4.1. Kết quả huấn luyện

Quá trình huấn luyện mô hình được theo dõi thông qua chỉ số CD loss trên tập train và tập validation qua từng epoch. Sự thay đổi của các giá trị loss trong suốt quá trình huấn luyện được minh họa trong biểu đồ dưới đây:



Hình 4.1: Biểu đồ train loss



Hình 4.2: Biểu đồ validation loss

Như đã thấy, training loss giảm đều qua các epoch, cho thấy mô hình dần học được đặc trưng của dữ liệu đầu vào. Từ epoch 1 đến 6, validation loss giảm mạnh, phản ánh quá trình học diễn ra hiệu quả cả trên tập train và tập validation. Bắt đầu từ epoch 7, độ giảm của validation loss chậm lại và dao động nhẹ trong khoảng [3.21e-3, 3.30e-3] từ epoch 11 đến 15, cho thấy mô hình đã dần hội tụ và không có dấu hiệu overfitting.

Trong đó, epoch 13 đạt validation loss thấp nhất là 3.2124e-3, được chọn làm best checkpoint để lưu lại trọng số mô hình. Việc lựa chọn checkpoint theo tiêu chí validation loss nhỏ nhất nhằm đảm bảo mô hình được đánh giá tại thời điểm tổng quát hóa tốt nhất.

4.2. Kết quả kiểm thử

Sau khi hoàn tất huấn luyện và lựa chọn checkpoint tốt nhất ở epoch 13, tôi tiến hành đánh giá mô hình trên tập test để kiểm chứng khả năng tổng quát hóa của mô hình. Việc đánh giá được thực hiện trên từng category trong tập dữ liệu, sử dụng hai metric đo lường độ tương đồng hình học giữa point cloud dự đoán và ground truth là Chamfer Distance Loss (CD) và Earth's Mover Distance Loss (EMD). Kết quả đánh giá chi tiết cho từng category được trình bày trong bảng dưới đây:

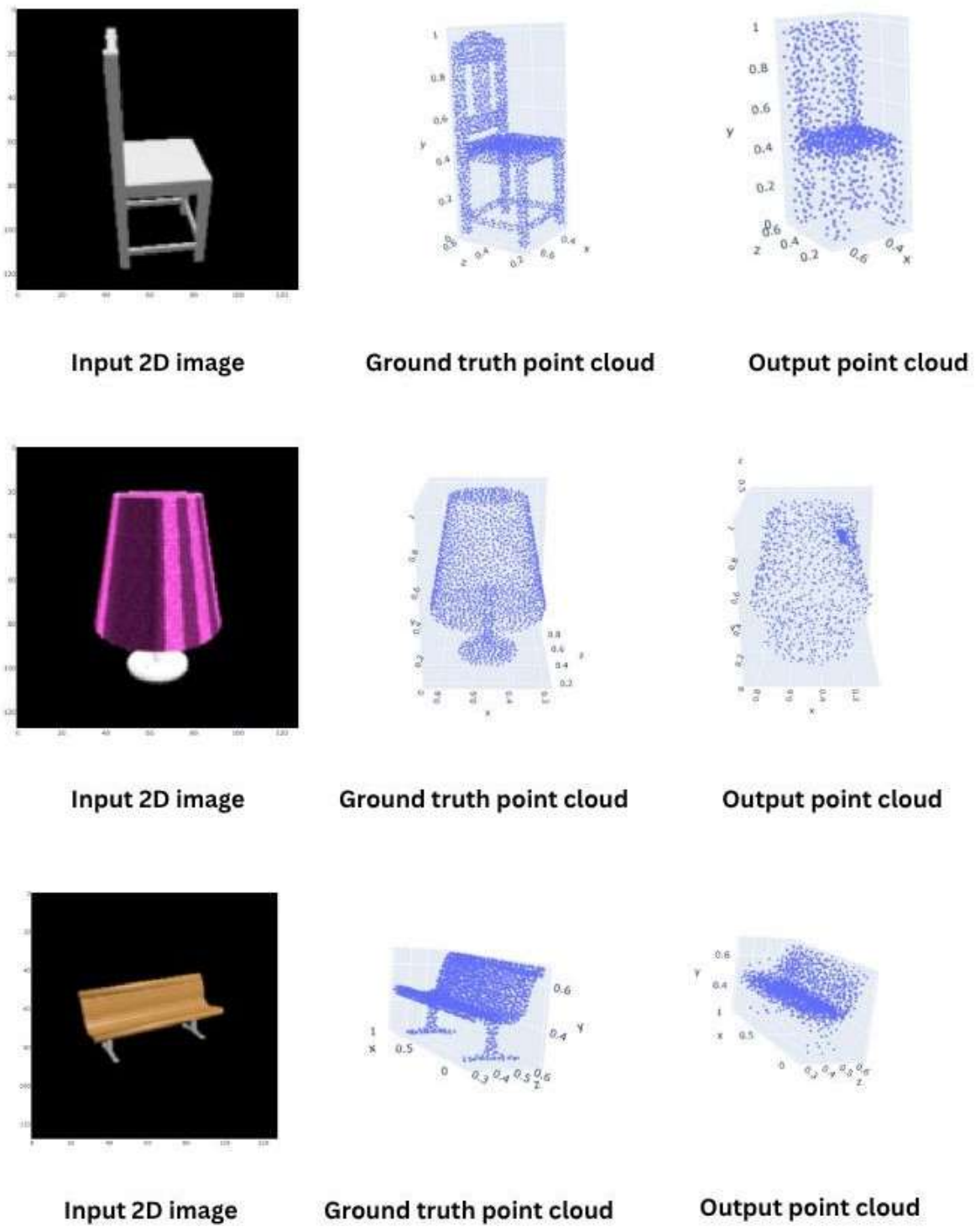
Bảng 4.1: Kết quả đánh giá trên tập test

Category	Chamfer Distance Loss	Earth's Mover Distance Loss
Chair	0.002969	0.119864
Sofa	0.002935	0.117158
Bench	**0.002864**	**0.115318**
Cabinet	0.002952	**0.120932**
Lamp	**0.003122**	0.119840
Table	0.002953	0.116849
Mean	0.002966	0.118327

Các kết quả cho thấy độ sai lệch giữa point cloud dự đoán và ground truth là rất nhỏ trên cả hai metric, phản ánh hiệu suất tái tạo hình dạng tốt và nhất quán của mô hình trên nhiều loại đối tượng khác nhau. Trong đó, mô hình đạt kết quả tốt nhất với Bench (CD thấp nhất 0.002864, EMD thấp nhất 0.115318), cho thấy khả năng học tốt các cấu trúc hình học đặc trưng của loại vật thể này.

Ngược lại, mô hình có kết quả sai lệch cao nhất với Lamp (CD cao nhất 0.003122) và Cabinet (EMD cao nhất 0.120932). Việc sai lệch cao ở Lamp có thể được lý giải bởi

tính chất hình học phức tạp và nhiều chi tiết nhỏ của loại vật thể này. Bên cạnh đó, sự đa dạng lớn về kiểu dáng giữa các mẫu Lamp trong tập dữ liệu cũng khiến mô hình khó học được biểu diễn chung, từ đó ảnh hưởng đến độ chính xác của kết quả tái tạo.

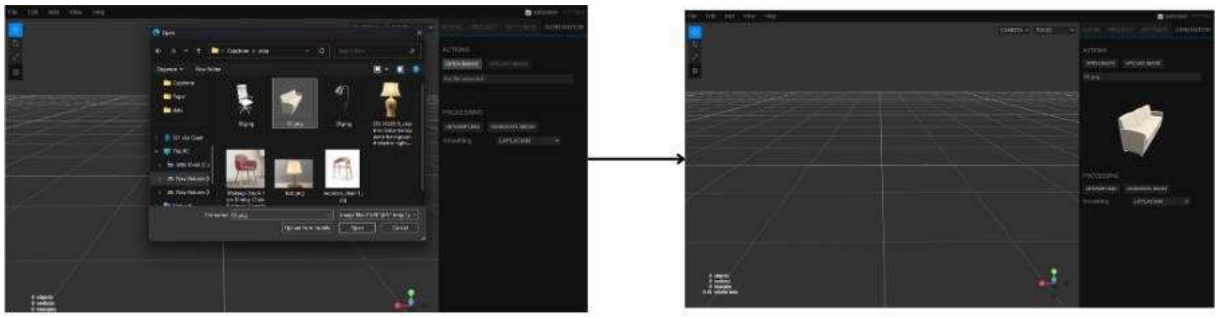


Hình 4.3: Kết quả định tính của mô hình trên các loại vật thể khác nhau. Từ trái qua phải: ảnh đầu vào, point cloud ground truth, point cloud dự đoán

4.3. Ứng dụng web editor 3D

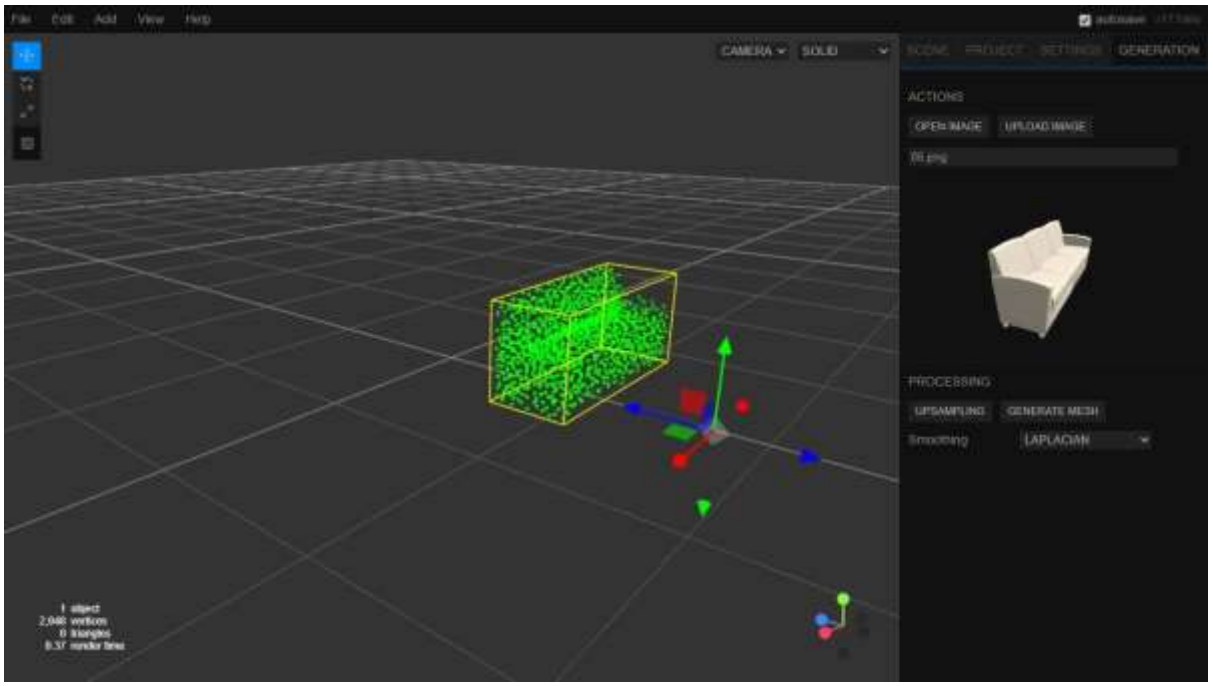
Kết quả đầu ra của hệ thống có thể được quan sát thông qua giao diện ứng dụng web được dựa trên mã nguồn mở Three.js Editor, được minh họa như các hình ảnh bên dưới.

Cụ thể, người dùng sẽ chọn và mở ảnh từ File System.



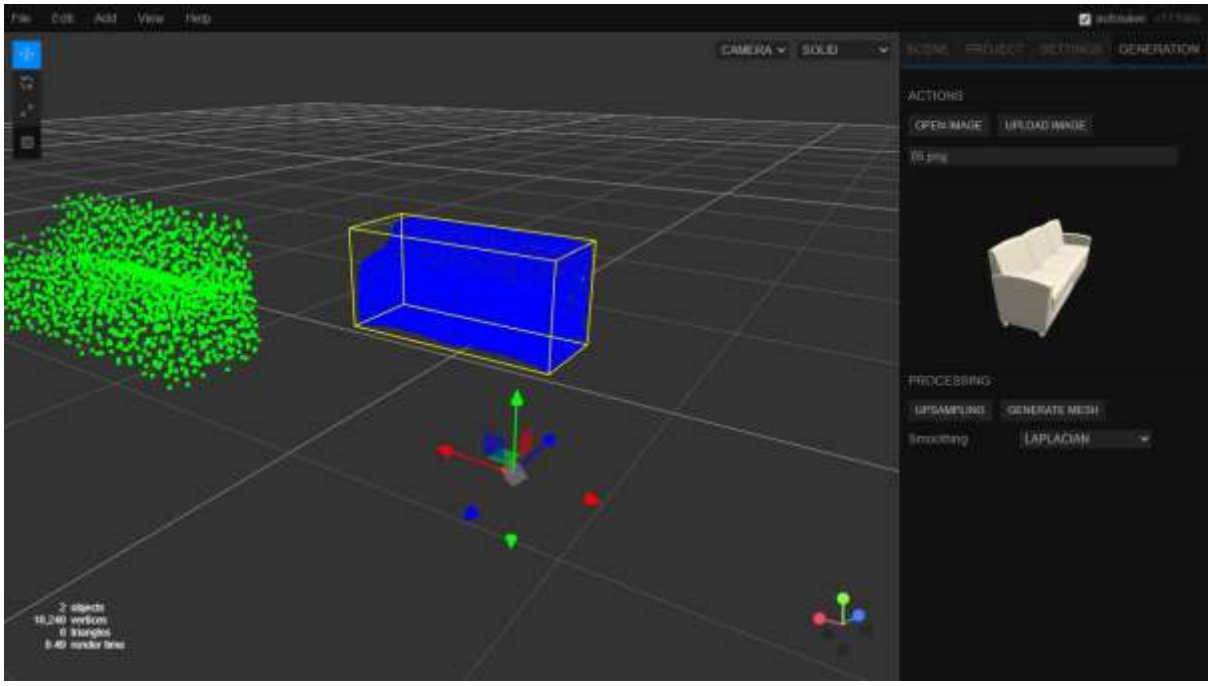
Hình 4.4: Chọn và upload ảnh

Sau khi upload ảnh lên thì nhận được kết quả mô hình vật thể dưới dạng biểu diễn point cloud:



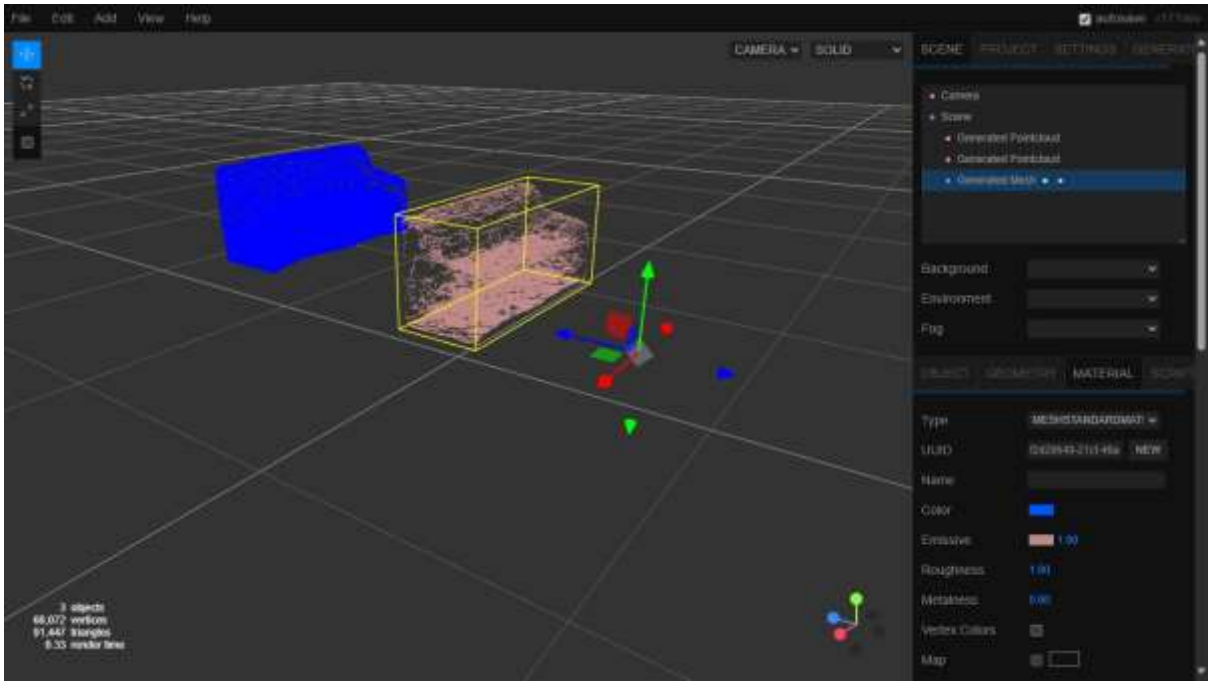
Hình 4.5: Kết quả vật thể 3D dưới dạng point cloud

Ứng dụng cung cấp chức năng cho phép upscale point cloud vừa sinh ra để tăng độ dày cho vật thể:



Hình 4.6: Point cloud sau khi đã được upscale

Sau khi có được point cloud sau khi upscale, tiến hành sinh mesh của vật thể và thu được kết quả:



Hình 4.7: Kết quả vật thể 3D dưới dạng mesh

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Thông qua quá trình nghiên cứu, tìm hiểu lý thuyết và triển khai thực nghiệm, dự án Xây dựng hệ thống tái tạo vật thể 3D từ ảnh 2D phục vụ cho thiết kế đồ họa đã hoàn thành phần lớn các mục tiêu đề ra. Tuy nhiên, cũng như nhiều công trình nghiên cứu thực nghiệm khác, hệ thống vẫn còn một số điểm cần cải thiện để đạt đến mức độ tối ưu và mở rộng tiềm năng ra ứng dụng thực tế.

Những kết quả đạt được:

- ***Về mặt học thuật***:
    - Đã khảo sát các phương pháp hiện đại trong bài toán tái tạo 3D từ ảnh 2D, bao gồm kỹ thuật phát sinh point cloud và khôi phục mesh, giúp xây dựng nền tảng lý thuyết cho hệ thống.
    - Lựa chọn và huấn luyện mô hình dự đoán point cloud dựa trên kiến trúc Pixel2Point, đồng thời đề xuất cải tiến input đầu vào từ ảnh RGB sang RGB-D, giúp mô hình khai thác được thông tin chiều sâu để tăng cường khả năng học đặc trưng không gian.
    - Phân tích hiệu quả của các mô hình hậu xử lý như SAPCU và GeoUDF trong việc tái tạo hình dạng mesh từ point cloud, qua đó tích hợp thành pipeline xử lý hoàn chỉnh.
    - Tìm hiểu các kỹ thuật đo lường hình học như Chamfer Distance và Earth's Mover Distance, nhằm đánh giá chính xác chất lượng tái tạo giữa dữ liệu đầu ra và ground truth.
    - Thiết kế quy trình huấn luyện hợp lý bao gồm lựa chọn kiến trúc, loss function, chiến lược chọn checkpoint và đánh giá mô hình để đảm bảo hiệu suất tốt nhất.
- ***Về mặt thực tiễn***:
    - Xây dựng hệ thống web ứng dụng với giao diện frontend kế thừa từ Three.js Editor, cho phép người dùng tải ảnh, nhận kết quả tái tạo và trực quan hóa mô hình dưới dạng point cloud và mesh.
    - Triển khai backend bằng FastAPI kết nối tới Cloud Run để thực hiện inference từ mô hình học sâu đã đóng gói dưới dạng container.
    - Tối ưu hóa quá trình triển khai bằng cách sử dụng Artifact Registry để quản lý container model và đảm bảo khả năng mở rộng, cập nhật linh hoạt.

- Xây dựng pipeline xử lý hoàn chỉnh từ ảnh → point cloud → mesh, cung cấp các chức năng xử lý nâng cao như smoothing và refinement mesh ngay trên trình duyệt.
- Hệ thống có khả năng hoạt động từ đầu vào ảnh thực tế với hiệu quả cao, tạo nền tảng để tích hợp vào các ứng dụng đồ họa, thiết kế sản phẩm, và công nghiệp sáng tạo.

Những hạn chế còn tồn tại:

- *Chất lượng point cloud chưa đồng đều giữa các category*: Mặc dù hệ thống hoạt động tốt với nhiều loại vật thể, nhưng với những đối tượng có hình dạng phức tạp hoặc cấu trúc mảnh như đèn (lamp), chất lượng point cloud còn chưa thực sự sắc nét, dẫn đến việc tái tạo mesh chưa đạt mức độ chi tiết mong muốn.
- *Phụ thuộc vào góc chụp và đặc điểm hình học của vật thể*: Các góc chụp khuất, che lấp nhiều phần bề mặt vật thể là thách thức lớn trong quá trình học và tái tạo. Dù đầu vào có chứa thông tin depth, mô hình vẫn gặp khó khăn trong việc dự đoán các phần bị ẩn, dẫn đến kết quả tái tạo chưa trọn vẹn ở một số trường hợp.
- *Chưa xử lý triệt để lỗi hình học sau khôi phục*: Ở một số trường hợp, mesh được tái tạo vẫn tồn tại các lỗi nhỏ như lỗ hổng, mặt tam giác bị lệch hoặc không khép kín hoàn toàn. Các bước smoothing hoặc refinement mới chỉ thực hiện ở mức cơ bản và chưa tối ưu về mặt hình học.
- *Thiếu đánh giá định lượng trên dữ liệu thực tế*: Hệ thống chủ yếu được huấn luyện và kiểm thử trên bộ dữ liệu tổng hợp. Việc thử nghiệm với ảnh thực tế trong điều kiện môi trường ngoài còn hạn chế, khiến tính tổng quát hóa chưa được đánh giá đầy đủ.

Định hướng phát triển:

- *Nâng cao chất lượng tái tạo point cloud*: Tiếp tục cải tiến mô hình dự đoán point cloud bằng cách thử nghiệm các kiến trúc học sâu hiện đại hơn hoặc kết hợp cơ chế attention giúp mô hình hiểu tốt hơn các chi tiết không gian. Có thể nghiên cứu các phương pháp kết hợp thông tin không gian hiệu quả hơn từ ảnh RGB-D để tăng độ chính xác trong các vùng hình học mảnh hoặc khuất tầm nhìn.
- *Cải tiến các bước hậu xử lý*: Tăng cường các thuật toán refinement và mesh repair để khắc phục lỗi hình học còn tồn tại như các lỗ mesh, mặt lệch hoặc

mesh không khép kín. Có thể tích hợp các công cụ xử lý hình học mạnh hơn như Poisson Surface Reconstruction hoặc các phương pháp học sâu tái tạo mesh trực tiếp từ point cloud.

- *Tối ưu hóa hệ thống cho ứng dụng thực tiễn*: Cải thiện tốc độ inference và giảm độ trễ xử lý bằng cách tối ưu kiến trúc mô hình, loại bỏ các tầng dư thừa, đồng thời tối ưu pipeline backend (FastAPI + Cloud Run). Bên cạnh đó, có thể triển khai thêm khả năng batch inference hoặc caching để phục vụ cho các tác vụ hàng loạt trong môi trường đồ họa chuyên nghiệp.

TÀI LIỆU THAM KHẢO

[1] E. Ahmed, A. Saint, A. E. R. Shabayek, K. Cherenkova, R. Das, G. Gusev, D. Aouada và B. E. Ottersten, “A survey on Deep Learning Advances on Different 3D Data Representations,” arXiv preprint arXiv: Computer Vision and Pattern Recognition, 2018.

[2] M. Jia và M. Zhang, "An Overview of Methods and Applications of 3D Reconstruction," International Journal of Computer Science and Information Technology, vol. 3, no. 1, 2024, doi: 10.62051/ijcsit.v3n1.03.

[3] J. L. Posdamer, “Computer Geometric Modeling For Machine Perception Of Three-Dimensional Solids,” trong Proc. SPIE, vol. 283, 1981, doi:10.1117/12.931983.

[4] B. K. P. Horn, “Shape From Shading: A Method for Obtaining the Shape of a Smooth Opaque Object From One View,” MIT Artificial Intelligence Laboratory, Tech. Rep. AITR-232, Nov. 1970.

[5] R. Muhr, G. Schutte và M. Vincze, “Measurement of the 3-D shape of specular polyhedrons using an M-array coded light source,” trong Proc. ISPRS Commission V Symposium, Newcastle upon Tyne, UK, 2010.

[6] N. Snavely, S. M. Seitz và R. Szeliski, “Photo Tourism: Exploring Photo Collections in 3D,” trong Proc. ACM SIGGRAPH, Los Angeles, CA, USA, Jul. 2006, tr. 835–846, doi:10.1145/1179352.1141964.

[7] J. Han, L. Shao, D. Xu và J. Shotton, “Enhanced Computer Vision With Microsoft Kinect Sensor: A Review,” IEEE Trans. Cybern., vol. 43, no. 5, tr. 1318–1334, Oct. 2013, doi:10.1109/TCYB.2013.2265376.

[8] J. Wu, C. Zhang, T. Xue, W. T. Freeman và J. B. Tenenbaum, “Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling,” NeurIPS, 2016.

[9] A. Choy, D. Xu, J. Gwak, K. Chen và S. Savarese, “3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction,” in ECCV, 2016, tr. 628–644, doi:10.1007/978-3-319-46478-7_38.

[10] Y. Liu, J. Wang, Z. Chen và J. Li, “3D-FHNet: Three-Dimensional Fusion Hierarchical Reconstruction Method for Any Number of Views,” arXiv preprint arXiv:2301.01234, 2024.

[11] A. Sinha, A. Unmesh, Q.-X. Huang, và K. Ramani, "SurfNet: Generating 3D shape surfaces using deep residual networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 791-800.

[12] X. Li, Z. Chen, H. Zhang, và L. Zhang, "DeformNet: Free-Form Deformation Network for 3D Shape Reconstruction from a Single Image," arXiv preprint arXiv:1708.01736, 2017.

[13] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, và Y. Jiang, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images," in Proc. European Conf. Computer Vision (ECCV), 2018, pp. 52-67.

[14] H. Fan, H. Su, và L. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2463-2471.

[15] Y. Liu, X. Li, và H. Zhang, "RealPoint3D: An Efficient Generation Network for 3D Object Reconstruction From a Single Image," arXiv preprint arXiv:1903.11716, 2019.

[16] J. Wang, Y. Zhang, và Z. Huang, "Pixel2point: 3D Object Reconstruction From a Single Image Using CNN and Initial Sphere," arXiv preprint arXiv:1805.07026, 2018.

[17] A. X. Chang et al., "ShapeNet: An Information-Rich 3D Model Repository," arXiv preprint arXiv:1512.03012, 2015.

[18] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, và J. Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912-1920.

[19] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, và M. Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5828-5839.

[20] H. Sun, S. Song, S. Liu, J. Guo, và J. Zhang, "Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2974-2983.

[21] Q. Xu, W. Wang, D. Ceylan, R. Mech, và U. Neumann, "DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019.

[22] Y. A. Shleibik, "3D Reconstruction of 2D Images Using Deep Leaning," M.S. thesis, Dept. Computer Science, University of Colorado Colorado Springs, Colorado Springs, CO, USA, 2023.

[23] L. Yu, X. Li, C. Fu, D. Cohen-Or, và P. Heng, "PU-Net: Point Cloud Upsampling Network," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2790-2799.

[24] Y. Wang, Q. Feng, và H. Zhang, "PU-GCN: Point Cloud Upsampling using Graph Convolutional Networks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2021, pp. 11888-11897.

[25] J. Huang, Y. Wang, S. Liu, và C. Theobalt, "Self-Supervised Arbitrary-Scale Point Clouds Upsampling via Implicit Neural Representation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2022.

[26] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, và J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," ACM Transactions on Graphics (TOG), vol. 38, no. 5, pp. 1-12, 2019.

[27] J. Park, P. Florence, J. Straub, R. Newcombe, và S. Lovegrove, "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 165-174.

[28] X. Guo, J. Huang, J. Yu, và H. Zhang, "GeoUDF: Surface Reconstruction from 3D Point Clouds via Geometry-guided Distance Representation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2023.