

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN



HUỖNH KHÁNH TRẦN
MSSV: 219979

**HỆ THỔNG XÂY DỰNG TỰ ĐỘNG VẬT THỂ 3D
TỪ VIDEO SỬ DỤNG TRÍ TUỆ NHÂN TẠO (AI)**

KHỎA LUẬN TỐT NGHIỆP ĐẠI HỌC
Ngành: Công Nghệ Thông Tin
Mã số ngành: 7480201

Cần Thơ, tháng 6 năm 2025

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN

HUỲNH KHÁNH TRẦN
MSSV: 219979

HỆ THỐNG XÂY DỰNG TỰ ĐỘNG VẬT THỂ 3D
TỪ VIDEO SỬ DỤNG TRÍ TUỆ NHÂN TẠO (AI)

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC
Ngành: Công Nghệ Thông Tin
Mã số ngành: 7480201

GIẢNG VIÊN HƯỚNG DẪN
TS. NGÔ HỒ ANH KHÔI

Cần Thơ, tháng 6 năm 2025

CHẤP THUẬN HỘI ĐỒNG

Khóa luận “*Hệ thống xây dựng tự động vật thể 3D từ video sử dụng trí tuệ nhân tạo (ai)*”, do sinh viên **Huỳnh Khánh Trân** thực hiện dưới sự hướng dẫn của Giảng viên **Ngô Hồ Anh Khôi**. Khóa luận tốt nghiệp đã báo cáo và được Hội đồng chấm khóa luận tốt nghiệp thông qua ngày.... tháng năm 20....

Ủy Viên

Thư ký

.....

.....

Phản biện 1

Phản biện 2

.....

.....

Cán bộ hướng dẫn

Chủ tịch Hội đồng

.....

.....

LỜI CẢM TẠ

Mỗi sự thành công đều gắn liền với sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp từ những người xung quanh. Trong suốt chặng đường học tập, từ những bước chân đầu tiên đến giảng đường đại học cho đến ngày hôm nay, em luôn nhận được sự quan tâm, động viên và hỗ trợ quý báu từ thầy cô, gia đình và bạn bè.

Trước hết, em xin gửi lời cảm ơn trân trọng và sâu sắc đến quý thầy cô Khoa Công nghệ Thông tin, Trường Đại học Nam Cần Thơ đã tận tình giảng dạy, định hướng và truyền đạt cho em những kiến thức quý báu trong suốt quá trình học tập. Chính nhờ sự dìu dắt, chỉ bảo tận tâm của thầy cô mà em đã có nền tảng vững chắc để thực hiện và hoàn thành khóa luận tốt nghiệp với đề tài "Hệ thống xây dựng tự động vật thể 3D từ video sử dụng trí tuệ nhân tạo (AI)".

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến TS. Ngô Hồ Anh Khôi – người đã trực tiếp hướng dẫn, hỗ trợ và đồng hành cùng em trong suốt quá trình thực hiện khóa luận. Sự tận tình chỉ bảo, những góp ý quý báu của thầy là yếu tố quan trọng giúp em hoàn thiện đề tài này một cách hiệu quả nhất.

Em cũng xin chân thành cảm ơn Ban giám hiệu Trường Đại học Nam Cần Thơ, cùng toàn thể thầy cô trong khoa đã luôn tạo điều kiện thuận lợi cho em trong quá trình học tập và nghiên cứu.

Do giới hạn về thời gian cũng như kinh nghiệm thực tiễn còn hạn chế, khóa luận không thể tránh khỏi những thiếu sót. Em kính mong nhận được sự góp ý, chỉ dẫn từ quý thầy cô để em có thể tiếp tục hoàn thiện, nâng cao kiến thức và kỹ năng của bản thân, phục vụ tốt hơn cho công việc trong tương lai.

Em xin chân thành cảm ơn!

Cần thơ, ngày 17 tháng 06 năm 2025

Sinh viên thực hiện

LỜI CAM ĐOAN

Em xin cam kết nội dung khóa luận này được hoàn thành dựa trên các kết quả nghiên cứu của cá nhân và các kết quả này chưa được dùng cho bất cứ khóa luận cùng cấp nào khác. Và những kết quả đạt được hoàn thành dựa trên các kết quả nghiên cứu của em trong khuôn khổ của của đề tài "Hệ thống xây dựng tự động vật thể 3D từ video sử dụng trí tuệ nhân tạo (AI)" được trình bày trong quyển báo cáo nghiên cứu này.

Cần thơ, ngày 17 tháng 06 năm 2025

Sinh viên thực hiện

MỤC LỤC

CHƯƠNG 1 GIỚI THIỆU	1
1.1 BỐI CẢNH PHÁT TRIỂN	1
1.2 MỤC TIÊU NGHIÊN CỨU	2
1.3 ĐỐI TƯỢNG NGHIÊN CỨU	2
1.4 PHẠM VI NGHIÊN CỨU.....	3
1.5 PHƯƠNG PHÁP LUẬN	3
CHƯƠNG 2 TỔNG QUAN CÁC CÔNG TRÌNH LIÊN QUAN VÀ CƠ SỞ LÝ THUYẾT	4
2.1 TỔNG QUAN CÁC CÔNG TRÌNH LIÊN QUAN	4
2.1.1 Structure from Motion kết hợp với Multi-View Stereo	4
2.1.2 Cách Mạng Thần Kinh trong Biểu Diễn Cảnh 3D	5
2.2 CƠ SỞ LÝ THUYẾT	6
2.2.1 Trích xuất khung hình từ video	6
2.2.2 Loại bỏ nền vật thể	7
2.2.3 Trích xuất và mô tả đặc trưng	8
2.2.4 Tái dựng cấu trúc 3D từ nhiều ảnh	9
2.2.5 Tái dựng bề mặt chi tiết và tạo lưới 3D.....	9
CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ GIẢI THUẬT	11
3.1 KIẾN TRÚC TỔNG THỂ CỦA HỆ THỐNG	11
3.2 TRÍCH XUẤT KHUNG HÌNH TỪ VIDEO VỚI FFMPEG	13
3.3 TÁCH NỀN VỚI REMBG	13
3.4 TRÍCH XUẤT ĐẶC TRƯNG VỚI SUPERPOINT	14
3.4.1 Tổng quan SuperPoint.....	14
3.4.2 Sự khác biệt giữa đầu ra SuperPoint và đầu vào OpenMVG.....	18
3.5 XÂY DỰNG SFM VỚI OPENMVG	19
3.5.1 Khởi tạo dữ liệu.....	20
3.5.2 Tính toán đặc trưng	21
3.5.3 Tính toán điểm khớp	22
3.5.4 Giải bài toán SfM	22

3.6 TÁI TẠO BỀ MẶT VỚI OPENMVS	22
3.6.1 Chuyển đổi dữ liệu từ OpenMVG sang OpenMVS	22
3.6.2 Tái tạo đám mây điểm dày đặc.....	23
3.6.3 Tái dựng lưới 3D	24
3.6.4 Tinh chỉnh lưới	25
3.6.5 Áp kết cấu cho mô hình.....	26
3.6.6 Kết xuất mô hình 3D	27
CHƯƠNG 4 XÂY DỰNG MÔ HÌNH VÀ ĐÁNH GIÁ KẾT QUẢ.....	28
4.1 THIẾT LẬP MÔI TRƯỜNG THỬ NGHIỆM ĐỀ XUẤT.....	28
4.2 SO SÁNH KẾT QUẢ TRONG ĐIỀU KIỆN CÓ VÀ KHÔNG CÓ GPU	28
4.3 KẾT QUẢ HỆ THỐNG ĐỀ XUẤT SO VỚI GAUSSIAN SPLATTING.....	30
CHƯƠNG 5 SẢN PHẨM NGƯỜI DÙNG.....	34
5.1 CHỨC NĂNG SẢN PHẨM.....	34
5.2 CÁC THÀNH PHẦN CỦA SẢN PHẨM	37
5.3 THỰC THỂ.....	40
5.3.1 Thực thể models3d	40
5.3.2 Thực thể users.....	40
5.3.3 Thực thể payments	41
5.3.4 Thực thể messages.....	41
5.3.5 Thực thể settings	42
5.4 SƠ ĐỒ ERD.....	42
5.5 GIAO DIỆN	43
CHƯƠNG 6 TỔNG KẾT	47
6.1 KẾT LUẬN.....	47
6.2 HƯỚNG PHÁT TRIỂN TƯƠNG LAI.....	47
TÀI LIỆU THAM KHẢO	50

DANH SÁCH BẢNG

Bảng 3.1 Quy trình và Công nghệ sử dụng của hệ thống đề xuất.....	12
Bảng 3.2 Đánh giá theo loại biến đổi.....	14
Bảng 3.3 Đánh giá theo nhiều ngưỡng sai số homography	15
Bảng 3.4 Đầu ra của SuperPoint và đầu vào của OpenMVG	18
Bảng 3.5 Tóm tắt cách khắc phục	19
Bảng 4.1 Kết quả đánh giá có GPU và không có GPU.....	30
Bảng 4.2 So sánh tổng quát Hệ thống nghiên cứu và Gaussian Splatting	32
Bảng 5.1 Thực thể models3d	40
Bảng 5.2 Thực thể users.....	40
Bảng 5.3 Thực thể payments.....	41
Bảng 5.4 Thực thể messages	41
Bảng 5.5 Thực thể settings.....	42

DANH SÁCH HÌNH

Hình 3.1 Quy trình tổng thể của hệ thống.....	11
Hình 3.2 Ảnh trước khi xóa nền.....	13
Hình 3.3 Ảnh sau khi xóa nền.....	13
Hình 3.4 Khớp đặc trưng giữa ảnh 1 và ảnh 2	17
Hình 3.5 Khớp đặc trưng giữa ảnh 1 và ảnh 19	17
Hình 3.6 Vị trí camera được trích xuất từ video	20
Hình 3.7 Đám mây điểm dày	24
Hình 3.8 Lưới 3D	25
Hình 3.9 Lưới 3D đã tinh chỉnh	25
Hình 3.10 Mô hình hoàn thiện	26
Hình 3.11 Ảnh kết cấu	26
Hình 4.1 Minh họa mesh hệ thống đề xuất	31
Hình 4.2 Minh họa điểm Gaussian 3D.....	31
Hình 4.3 Mô hình từ hệ thống đề xuất	31
Hình 4.4 Mô hình từ Gaussian Splatting.....	31
Hình 5.1 Sơ đồ chức năng.....	34
Hình 5.2 Sơ đồ Usecase	35
Hình 5.3 Sơ đồ ERD	42
Hình 5.4 Giao diện trang chủ	43
Hình 5.5 Giao diện trang đăng nhập, đăng ký	44
Hình 5.6 Giao diện trang liên hệ	44
Hình 5.7 Giao diện trang hướng dẫn.....	45
Hình 5.8 Giao diện trang tài khoản người dùng.....	45
Hình 5.9 Giao diện trang xem mô hình.....	46
Hình 5.10 Giao diện các trang admin	46

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Giải thích
AI	Artificial Intelligence
SFM	Structure from Motion
MVS	Multi-View Stereo
CNN	Convolutional Neural Network
NERF	Neural Radiance Fields
GPU	Graphics Processing Unit
CPU	Central Processing Unit

CHƯƠNG 1 GIỚI THIỆU

1.1 BỐI CẢNH PHÁT TRIỂN

Việc tái tạo và dựng hình vật thể 3D đóng vai trò quan trọng trong thị giác máy tính nhờ vào phạm vi ứng dụng rộng rãi. Có nhiều kỹ thuật để thực hiện điều này, chẳng hạn như máy quét laser và phép đo ảnh (photogrammetry). Trong những năm gần đây, các phương pháp dựa trên trí tuệ nhân tạo, như trường bức xạ thần kinh (Neural Radiance Field), đã được đề xuất và đạt được những kết quả ấn tượng (Van-Linh Nguyen và cộng sự, n.d).

Tái tạo mô hình vật thể 3D là quá trình chuyển đổi hình dạng và bề mặt của các đối tượng trong thế giới thực thành các mô hình 3D dưới dạng số hóa. Công nghệ này đóng vai trò ngày càng quan trọng trong nhiều lĩnh vực như thị giác máy, robot, thực tế ảo, bảo tồn di sản văn hóa, y học chẩn đoán hình ảnh và thiết kế công nghiệp. Việc xây dựng các mô hình 3D không chỉ giúp mô phỏng môi trường ảo một cách sinh động mà còn nâng cao khả năng tương tác giữa con người và máy móc, đồng thời cung cấp các mô hình có độ chính xác cao phục vụ cho nhiều công việc. Khả năng xác định hình học và tọa độ không gian 3D của các điểm trên bề mặt vật thể mang lại giá trị lớn cả về mặt nghiên cứu lẫn ứng dụng thực tiễn (Chuanzhi Xu và cộng sự, 2025).

Tái tạo mô hình 3D từ hình ảnh là một hướng nghiên cứu quan trọng trong thị giác máy tính, đã được quan tâm từ nhiều thập kỷ trước. Ban đầu, các phương pháp chủ yếu dựa vào nguyên lý hình học như tam giác hóa và quang trắc học, với quá trình trích xuất và ghép nối điểm đặc trưng thường thực hiện thủ công hoặc bán tự động, đòi hỏi điều kiện chụp lý tưởng và ảnh chất lượng cao nên khó ứng dụng rộng rãi. Từ những năm đầu thế kỷ XX, các kỹ thuật như Structure from Motion (SfM) và Multi-View Stereo (MVS) ra đời, giúp tự động hóa quá trình tái dựng mô hình từ nhiều ảnh chụp ở các góc nhìn khác nhau (Massimiliano Pepe và cộng sự, 2022). Bước tiến lớn xảy ra khi trí tuệ nhân tạo, đặc biệt là học sâu, được ứng dụng vào thị giác máy tính với các mô hình như mạng nơ-ron tích chập (CNN), Transformer, và gần đây là Neural Radiance Fields (NeRF), mang lại độ chính xác và mức độ chân thực cao hơn cho mô hình 3D. Cùng với đó, video ngày càng trở thành nguồn dữ liệu phổ biến nhờ cung cấp nhiều góc nhìn liên tục, cho phép kết hợp hiệu quả với các thuật toán SfM, MVS và AI trong các bước như trích xuất đặc trưng, tách nền, từ đó tạo nên một quy trình tái tạo mô hình tự động, chính xác và hiệu quả hơn. Hiện nay, với sự phát triển mạnh mẽ của phần cứng và các nền tảng hỗ trợ, công nghệ tái tạo mô hình 3D từ video đã trở nên dễ tiếp cận và sẵn sàng phục vụ cho nhiều lĩnh vực như giáo dục, y tế, công nghiệp và giải trí.

Trong những năm gần đây, việc ứng dụng trí tuệ nhân tạo (AI) vào quá trình tái tạo mô hình 3D từ hình ảnh, đặc biệt là từ video, đang thu hút sự quan tâm mạnh mẽ. Các phương pháp truyền thống thường dựa vào kỹ thuật hình học cơ bản và yêu cầu nhiều thao tác thủ công, do đó gặp những hạn chế khi xử lý các cảnh vật phức tạp hoặc môi trường có nhiều biến đổi. Sự phát triển nhanh chóng của các mô hình học sâu

(deep learning), cùng với khả năng tiếp cận ngày càng dễ dàng hơn với các tài nguyên tính toán, đã tạo điều kiện cho những bước tiến vượt bậc trong kỹ thuật tái thiết 3D dựa trên AI. Trí tuệ nhân tạo có thể tự động hóa nhiều công đoạn trong quy trình tái tạo mô hình, từ đó nâng cao đáng kể tốc độ, độ chính xác và tính hiệu quả. Đặc biệt, video là một nguồn dữ liệu hình ảnh dồi dào và giàu thông tin đang nổi lên như một lựa chọn hấp dẫn cho việc xây dựng hệ thống tái tạo mô hình 3D một cách tự động và thông minh (Liang Ma và cộng sự, 2023).

1.2 MỤC TIÊU NGHIÊN CỨU

Nghiên cứu này hướng đến việc phát triển một hệ thống tự động tái tạo mô hình 3D từ video bằng cách kết hợp trí tuệ nhân tạo (AI) với các phương pháp truyền thống nhằm nâng cao độ chính xác, hiệu quả và khả năng ứng dụng trong thực tiễn.

Trước hết, nghiên cứu tiến hành khảo sát và phân tích tài liệu kỹ thuật liên quan đến các phương pháp tái tạo mô hình 3D từ ảnh và video, bao gồm các thuật toán truyền thống như Structure from Motion (SfM), Multi-View Stereo (MVS) cũng như các kỹ thuật hiện đại sử dụng học sâu như SuperPoint, NeRF hay các mô hình tách nền dựa trên AI. Việc nghiên cứu này giúp xây dựng nền tảng lý thuyết vững chắc, từ đó định hướng cho các giải pháp tiếp theo. Tiếp theo, nghiên cứu tiến hành thực nghiệm hệ thống trên nhiều đoạn video khác nhau, nhằm đánh giá khả năng tái tạo mô hình trong điều kiện thực tế. Các bước xử lý được triển khai tự động hóa từ đầu vào video đến đầu ra là mô hình 3D. Cuối cùng, sản phẩm được đem đi so sánh trong nhiều điều kiện khác nhau đồng thời được so sánh với các phương pháp hiện có như Gaussian Splatting. Việc so sánh giúp đánh giá hiệu quả của hệ thống đề xuất cả về mặt chất lượng mô hình, thời gian xử lý và mức độ dễ sử dụng. Qua đó, nghiên cứu hướng tới mục tiêu xây dựng một giải pháp đơn giản, hiệu quả, và thân thiện với người dùng phổ thông trong việc tạo mô hình 3D từ video thông thường.

Mục tiêu cuối cùng là xây dựng một giải pháp đơn giản hóa quy trình tạo mô hình 3D, giảm sự phụ thuộc vào kỹ thuật đồ họa chuyên sâu, từ đó hỗ trợ người dùng phổ thông dễ dàng tiếp cận công nghệ mô hình hóa 3D từ video thông thường.

1.3 ĐỐI TƯỢNG NGHIÊN CỨU

Đề tài tập trung vào việc xây dựng hệ thống tái tạo mô hình 3D tự động từ video, thông qua sự kết hợp giữa các phương pháp truyền thống và trí tuệ nhân tạo (AI), nhằm nâng cao độ chính xác cũng như mức độ tự động hóa của quy trình.

Các thành phần chính được nghiên cứu và triển khai trong hệ thống bao gồm ba bước chính. Đầu tiên là tách nền khỏi khung hình bằng cách sử dụng mạng nơ-ron U-2-Net thông qua thư viện rembg, giúp loại bỏ các chi tiết không liên quan và tập trung vào đối tượng chính trong video. Tiếp theo là quá trình tái tạo mô hình 3D, trong đó hệ thống kết hợp phương pháp Structure from Motion và Multi-View Stereo với kỹ thuật trích xuất đặc trưng bằng mô hình học sâu SuperPoint. Việc tích hợp SuperPoint giúp nâng cao độ chính xác trong việc phát hiện và khớp đặc trưng giữa các khung hình, từ đó cải thiện chất lượng mô hình không gian và phục hồi bề mặt chi tiết tốt hơn. Cuối

cùng, kết quả tái tạo từ hệ thống được đánh giá và so sánh với phương pháp Gaussian Splatting là một kỹ thuật hiện đại hoàn toàn dựa trên AI, nhằm phân tích hiệu quả, độ chính xác và tính khả thi của từng cách tiếp cận.

1.4 PHẠM VI NGHIÊN CỨU

Đề tài tập trung vào việc nghiên cứu và ứng dụng trong phạm vi nước Việt Nam, với trọng tâm là khai thác những tiến bộ công nghệ trong những năm trở lại đây, đặc biệt trong các lĩnh vực thị giác máy tính (computer vision) và học sâu (deep learning). Đây là giai đoạn chứng kiến sự phát triển mạnh mẽ của các kỹ thuật tái dựng mô hình 3D từ dữ liệu 2D, mang lại nhiều cơ hội đổi mới trong việc số hóa vật thể thực. Tuy nhiên, việc triển khai các giải pháp tái tạo 3D vẫn còn gặp nhiều rào cản, chủ yếu do chi phí cao của các thiết bị chuyên dụng như cảm biến chiều sâu (depth sensors) hoặc hệ thống quét 3D. Để khắc phục hạn chế này, đề tài hướng đến việc tận dụng dữ liệu video RGB thông thường, được quay từ điện thoại di động là một thiết bị phổ biến và dễ tiếp cận với nhiều người nhằm giảm thiểu yêu cầu về thiết bị quay chụp.

Toàn bộ quá trình nghiên cứu được thực hiện trong điều kiện thiết bị có cấu hình cố định, không sử dụng dữ liệu đầu vào từ cảm biến LiDAR hay ảnh độ sâu. Điều này vừa đảm bảo tính khả thi trong môi trường ứng dụng thực tế tại Việt Nam và phù hợp cho quá trình nghiên cứu và phát triển hệ thống, vừa góp phần thúc đẩy việc phổ cập công nghệ 3D đến các lĩnh vực giáo dục, nghiên cứu và truyền thông số.

1.5 PHƯƠNG PHÁP LUẬN

Phương pháp tiếp cận tổng thể, đề tài bắt đầu bằng việc nghiên cứu và tìm hiểu các công nghệ hiện có liên quan đến tái tạo mô hình 3D từ video. Sau đó, tiến hành triển khai các công nghệ này để tạo ra sản phẩm ban đầu. Tiếp theo, nghiên cứu và tích hợp thêm các thành phần, công cụ mới nhằm nâng cao chất lượng và hiệu quả của sản phẩm cuối cùng.

Phương pháp triển khai và kiểm thử, sau khi lựa chọn và tích hợp các công nghệ, hệ thống được xây dựng và triển khai từng bước. Mỗi thành phần được kiểm thử độc lập và trong toàn bộ quy trình để đảm bảo hoạt động đúng và hiệu quả. Quá trình kiểm thử được thực hiện trên nhiều bộ dữ liệu khác nhau nhằm đánh giá chính xác khả năng vận hành và kết quả tái tạo mô hình 3D.

Phương pháp so sánh, phân tích và đánh giá kết quả, kết quả từ hệ thống được so sánh trên nhiều cấu hình phần cứng khác nhau cùng với các thuật toán đa dạng, trong đó có phương pháp Gaussian Splatting. Qua việc so sánh này, đề tài đánh giá được ưu điểm và hạn chế của từng phương pháp, từ đó rút ra các điểm mạnh cần phát huy và những điểm yếu cần cải tiến để nâng cao hiệu quả hệ thống.

CHƯƠNG 2 TỔNG QUAN CÁC CÔNG TRÌNH LIÊN QUAN VÀ CƠ SỞ LÝ THUYẾT

Tái tạo mô hình 3D từ hình ảnh và video là một lĩnh vực đã được nghiên cứu từ lâu, khởi nguồn từ các nguyên lý hình học thị giác và kỹ thuật chụp ảnh lập thể. Các phương pháp truyền thống như Structure-from-Motion (SfM) và Multi-View Stereo (MVS), cùng với các thư viện mã nguồn mở như OpenMVG và OpenMVS, đã góp phần phổ biến kỹ thuật này trong cả học thuật lẫn công nghiệp. Tuy nhiên, những phương pháp này còn nhiều hạn chế khi áp dụng vào môi trường thực tế do phụ thuộc vào ánh sáng, chuyển động và yêu cầu xử lý thủ công phức tạp. Sự phát triển mạnh mẽ của trí tuệ nhân tạo trong thập kỷ qua, đặc biệt là các mô hình học sâu như U-2-Net, SuperPoint, Neural Radiance Fields (NeRF) và Gaussian Splatting, đã mở ra hướng đi mới cho việc tái tạo 3D với độ chính xác và tự động hóa cao hơn. Tuy vậy, các phương pháp kết xuất thần kinh hiện tại vẫn chưa tạo ra được mô hình dạng lưới một yếu tố cần thiết trong in 3D, kỹ thuật và y học. Xuất phát từ thực tiễn đó, đề tài này xây dựng một hệ thống tái tạo mô hình 3D hoàn toàn tự động từ video RGB thông thường, kết hợp giữa thuật toán hình học truyền thống và công nghệ học sâu hiện đại, nhằm tạo ra mô hình lưới chất lượng cao, dễ ứng dụng trong giáo dục, kỹ thuật, bảo tồn và đào tạo số.

2.1 TỔNG QUAN CÁC CÔNG TRÌNH LIÊN QUAN

Structure-from-Motion và Multi-View Stereo là hai kỹ thuật cốt lõi trong quy trình tái dựng mô hình 3D từ dữ liệu ảnh 2D. SfM cho phép ước lượng vị trí camera và trích xuất đám mây điểm thưa từ nhiều góc nhìn, trong khi MVS tiếp tục bổ sung chi tiết để tạo ra mô hình 3D dạng lưới với mật độ cao. Tuy nhiên, trong những năm gần đây, sự bùng nổ của các mô hình học sâu đã tạo ra một “cuộc cách mạng thần kinh” trong lĩnh vực biểu diễn cảnh 3D. Các phương pháp như Neural Radiance Fields và Gaussian Splatting không chỉ tái hiện không gian với độ chân thực vượt trội mà còn mở rộng khả năng hiển thị 3D theo thời gian thực, mở ra một kỷ nguyên mới cho công nghệ số hóa thế giới thực.

2.1.1 Structure from Motion kết hợp với Multi-View Stereo

Structure-from-Motion là kỹ thuật thị giác máy tính dùng để phục hồi mô hình 3D từ tập ảnh 2D của một cảnh tĩnh, đồng thời ước lượng chuyển động tương đối của các máy ảnh. Quy trình SfM gồm ba bước chính bao gồm trích xuất và khớp đặc trưng ảnh, ước lượng tư thế máy ảnh và tái dựng cấu trúc 3D bằng cách tối thiểu hóa sai số tái chiếu. Các phương pháp cổ điển như của Tomasi và Kanade sử dụng phân tích nhân tố để xử lý toàn cục dữ liệu ảnh, thay vì xử lý từng cặp riêng lẻ. Hai hướng tiếp cận chính là SfM gia tăng (incremental) và SfM toàn cục (global). Gần đây, SfM đang chuyển sang các mô hình học sâu và xử lý song song nhằm khắc phục hạn chế về độ chính xác và khả năng mở rộng. Một ví dụ điển hình là Fast3R – mô hình sử dụng Transformer để xử lý đồng thời nhiều ảnh trong một lượt truyền, cho phép mỗi ảnh truy cập thông tin toàn cục từ các ảnh còn lại mà không cần lặp lại bước căn chỉnh. Cách tiếp cận mới này giúp giảm lỗi tích lũy, tăng hiệu quả suy luận và mở ra hướng phát triển cho các hệ

thống SfM nhanh, chính xác và có khả năng mở rộng tốt hơn (Ronen Basri và cộng sự, 2017).

Multi-View Stereo là bước then chốt trong quy trình tái tạo 3D, nhằm tạo ra mô hình 3D dày đặc và chi tiết từ nhiều ảnh chụp cùng cảnh ở các góc nhìn khác nhau. MVS hoạt động dựa trên thông tin về tư thế máy ảnh và đám mây điểm thưa do Structure-from-Motion (SfM) cung cấp. Các phương pháp MVS truyền thống gồm: thể tích hóa không gian (voxel), làm việc trực tiếp trên đám mây điểm, và ước lượng bản đồ độ sâu rồi hợp nhất. Tuy nhiên, các kỹ thuật này gặp nhiều thách thức trong môi trường thực tế như nhiễu, chiếu sáng thay đổi, hoặc bề mặt phản chiếu, đồng thời đòi hỏi tài nguyên tính toán lớn và khó mở rộng (Fangjinhua Wang và cộng sự, 2024). Hiện nay, MVS đang chuyển mạnh sang hướng học sâu, nhờ khả năng học đặc trưng mạnh mẽ và thích nghi tốt với các tình huống phức tạp. Các mô hình deep learning giúp vượt qua giới hạn của việc khớp ảnh thủ công, mang lại độ chính xác cao hơn trong việc suy luận hình học ngay cả khi dữ liệu ảnh không đạt yêu cầu quang trắc truyền thống. MVS học sâu giúp công nghệ tái tạo 3D trở nên linh hoạt, ổn định và dễ áp dụng trong thực tế (Liang Zhang và cộng sự, 2021).

2.1.2 Cách Mạng Thần Kinh trong Biểu Diễn Cảnh 3D

Neural Radiance Fields (NeRF), được giới thiệu bởi Ben Mildenhall vào năm 2020, đã tạo ra bước ngoặt trong việc tái tạo cảnh 3D từ hình ảnh 2D thông thường thông qua biểu diễn ngầm bằng mạng nơ-ron. NeRF ánh xạ vị trí 3D và hướng nhìn sang màu RGB và mật độ thể tích bằng một hàm thể tích liên tục được học. Quá trình kết xuất bao gồm chiếu tia (ray casting), lấy mẫu theo thể tích, mã hóa vị trí (positional encoding), ước lượng bức xạ bằng mạng MLP và kết xuất thể tích để tạo ra màu sắc cuối cùng cho ảnh. Các cải tiến sau đó như Mip-NeRF, PlenOctrees hay các kỹ thuật mã hóa vị trí hiện đại đã giúp tăng hiệu suất, giảm chi phí bộ nhớ và nâng cao chất lượng. Dù vậy, NeRF vẫn gặp khó khăn khi tái tạo cảnh động, ánh sáng thay đổi mạnh hoặc tư thế camera sai lệch. Ngoài ra, NeRF ưu tiên mật độ thể tích nên chưa tối ưu cho việc trích xuất hình học rõ ràng như lưới tam giác hoặc in 3D. Tuy còn hạn chế, NeRF đã đánh dấu sự chuyển dịch quan trọng từ các biểu diễn 3D truyền thống (point cloud, mesh, voxel) sang biểu diễn ngầm dựa trên deep learning, mở rộng khả năng tạo mô hình 3D chân thực, liên tục và tự động hơn cho các ứng dụng như nhận thức cảnh, nội dung ảo và robot học (Denning Lu và cộng sự, 2023).

3D Gaussian Splatting là phương pháp tái tạo trường bức xạ mới được công bố tại SIGGRAPH 2023, sử dụng biểu diễn Gaussian 3D thay vì mạng thần kinh như NeRF. Nhờ rasterization rõ ràng, 3DGS cho phép kết xuất thời gian thực với chất lượng hình ảnh cao khi tổng hợp các góc nhìn mới từ ảnh hoặc video đa khung hình. Cảnh được mô hình hóa bằng các hạt Gaussian 3D bắt đầu từ đám mây điểm thưa do SfM cung cấp. Mỗi Gaussian có thể điều chỉnh hình dạng, giúp mô hình linh hoạt và tiết kiệm tính toán ở không gian trống. Tối ưu hóa xen kẽ: Thuộc tính của Gaussian (vị trí, phương sai, hướng) được tối ưu luân phiên để cải thiện độ chính xác hình học. Kết xuất nhận biết hiển thị: Thuật toán chỉ kết xuất các Gaussian có khả năng hiển thị, hỗ trợ

splatting dị hướng, giúp tăng tốc huấn luyện và kết xuất. So với NeRF, 3D Gaussian Splatting vừa giữ được hiệu ứng phụ thuộc góc nhìn vừa đạt tốc độ kết xuất rất nhanh, phù hợp cho các ứng dụng thời gian thực như VR, game hoặc thiết bị di động. Cả NeRF và 3DGS đều dùng Structure from Motion (SfM) để xác định tư thế máy ảnh. Tuy nhiên, 3DGS tận dụng trực tiếp đám mây điểm từ SfM để khởi đầu quá trình tối ưu Gaussian, thay vì huấn luyện toàn bộ mạng từ đầu (Michael Rubloff, 2025).

2.2 CƠ SỞ LÝ THUYẾT

Xuất phát từ những hạn chế trong các nghiên cứu trước đây, đề tài này tập trung cải thiện các bước quan trọng trong quy trình tái dựng 3D, đặc biệt là Structure-from-Motion (SfM) và Multi-View Stereo (MVS). Trong khi phần lớn hệ thống hiện tại sử dụng ảnh tĩnh làm dữ liệu đầu vào vốn dễ bị thiếu góc nhìn, gây ảnh hưởng đến chất lượng mô hình. Nghiên cứu này đề xuất khai thác video để trích xuất khung hình liên tục, giúp thu được nhiều góc nhìn đa dạng hơn và tăng độ đầy đủ của dữ liệu. Đồng thời, việc tích hợp bước tách nền trước khi xử lý bằng mô hình U-2-Net (thông qua thư viện rembg) giúp loại bỏ các yếu tố nhiễu nền, làm nổi bật vật thể chính và hỗ trợ hiệu quả cho các bước xử lý sau đó. Bên cạnh đó, đề tài áp dụng mô hình SuperPoint hỗ trợ cho các thuật toán trích xuất đặc trưng truyền thống, giúp phát hiện và mô tả điểm đặc trưng một cách chính xác và ổn định hơn, đặc biệt trong các khung hình có chất lượng không lý tưởng. Sự kết hợp giữa cải tiến dữ liệu đầu vào và hiện đại hóa quy trình xử lý đặc trưng góp phần nâng cao rõ rệt độ chính xác, tính ổn định và chất lượng của mô hình 3D thu được từ video RGB thông thường.

2.2.1 Trích xuất khung hình từ video

FFmpeg là một bộ công cụ dòng lệnh mã nguồn mở mạnh mẽ, được phát triển chủ yếu bằng ngôn ngữ lập trình C++, chuyên dùng để xử lý video và các luồng đa phương tiện khác. Bộ công cụ này tích hợp nhiều thư viện nội bộ, cho phép xử lý đa dạng các tác vụ liên quan đến đa phương tiện. Trong đó, libavcodec được sử dụng để mã hóa và giải mã các định dạng video, libavformat sẽ hỗ trợ xử lý các định dạng container phổ biến như MP4 và libswscale dùng để chuyển đổi kích thước và định dạng ảnh, còn libavfilter cho phép áp dụng các hiệu ứng và bộ lọc thông qua hệ thống pipeline. Với khả năng linh hoạt và hiệu suất cao, FFmpeg trở thành công cụ không thể thiếu trong nhiều ứng dụng liên quan đến xử lý video hiện nay.

Trong hệ thống dựng mô hình 3D từ video, FFmpeg đóng vai trò là bước khởi đầu quan trọng với chức năng chính là tách video đầu vào thành chuỗi ảnh tĩnh (.jpg hoặc .png) theo tốc độ khung hình được chỉ định, ví dụ như 10 khung hình/giây (fps). Lợi ích của việc chuyển đổi đầu vào video thành định dạng ảnh là giúp chuẩn hóa dữ liệu, tạo điều kiện thuận lợi cho quá trình xử lý bằng các công cụ thị giác máy tính như rembg và SuperPoint. Đồng thời, việc này còn cho phép điều chỉnh tần suất ảnh (frame rate) phù hợp với mức độ chi tiết mong muốn cũng như khả năng xử lý của hệ thống, từ đó tối ưu hiệu suất và chất lượng của quá trình tái tạo mô hình.

Việc sử dụng FFmpeg giúp đảm bảo rằng dữ liệu đầu vào được chuẩn hóa và linh hoạt về mặt xử lý, tạo nền tảng ổn định cho các bước tiếp theo như tách nền, trích xuất đặc trưng và tái dựng mô hình 3D.

2.2.2 Loại bỏ nền vật thể

U-2-Net là một mô hình học sâu mạnh mẽ và đơn giản, được thiết kế chuyên biệt cho nhiệm vụ phát hiện và phân tách đối tượng nổi bật trong ảnh (salient object detection). Dựa trên kiến trúc U-Net truyền thống, U-2-Net được cải tiến để xử lý hiệu quả các ảnh có độ phân giải cao trong khi vẫn giữ được chi tiết và đường biên sắc nét. Nhờ khả năng phân đoạn chính xác, mô hình này đã trở thành công cụ quan trọng trong nhiều lĩnh vực như quảng cáo, điện ảnh, y học, nơi việc tách đối tượng chính ra khỏi nền là yêu cầu thiết yếu. Trong khuôn khổ hệ thống tái tạo mô hình 3D từ video RGB, U-2-Net được ứng dụng thông qua thư viện mã nguồn mở rembg để tự động loại bỏ nền khỏi từng khung hình trích xuất từ video. Việc tách bỏ các thành phần không liên quan trong nền giúp làm nổi bật vật thể chính, từ đó tăng độ chính xác cho các bước xử lý tiếp theo như trích xuất đặc trưng và tái dựng hình học ba chiều. Một điểm mạnh đáng chú ý là U-2-Net vẫn hoạt động tốt trên các máy tính không có GPU chuyên dụng, cho phép triển khai hệ thống trên cả các thiết bị phổ thông, dễ dàng áp dụng trong thực tế.

Rembg là một công cụ mã nguồn mở được xây dựng bằng ngôn ngữ Python, sử dụng công nghệ học sâu để thực hiện tách nền hoàn toàn tự động. Trọng tâm của rembg chính là mô hình U-2-Net, một mạng nơ-ron tích chập sâu (CNN) được huấn luyện cho bài toán phân đoạn vật thể nổi bật. Nhờ hoạt động không yêu cầu người dùng can thiệp thủ công, rembg rất phù hợp cho các quy trình xử lý hàng loạt ảnh từ video, tiết kiệm thời gian và công sức. Về mặt kỹ thuật, rembg sử dụng PyTorch làm framework chính để định nghĩa và chạy mô hình học sâu. PyTorch đảm nhận việc xử lý tensor đầu vào, thực hiện lan truyền tiến qua mạng nơ-ron trên CPU hoặc GPU, giúp quá trình tách nền diễn ra nhanh chóng và chính xác. Bên cạnh đó, thư viện NumPy hỗ trợ thao tác dữ liệu dạng mảng, giúp chuyển đổi ảnh RGB sang dạng số phù hợp với mô hình, đồng thời kết hợp mặt nạ đầu ra với ảnh gốc để tạo ra kết quả tách nền hoàn chỉnh.

Trong pipeline của hệ thống, ảnh đầu vào được trích xuất từ video bằng công cụ FFmpeg. Sau đó, rembg sử dụng mô hình U-2-Net để thực hiện tách nền, ảnh được chuyển thành tensor và đưa vào mạng nơ-ron, từ đó tạo ra mặt nạ foreground xác định vùng chứa đối tượng chính. Kết quả đầu ra là ảnh chỉ giữ lại vật thể cần thiết, với nền được làm trong suốt hoặc thay bằng nền trắng, tùy cấu hình. Quá trình tách nền mang lại nhiều lợi ích rõ rệt trong pipeline dựng hình 3D. Khi nền được loại bỏ, các thuật toán trích xuất và khớp đặc trưng hoạt động chính xác hơn vì không bị nhiễu bởi các chi tiết không liên quan. Điều này giúp giảm sai số trong bước tái dựng cấu trúc không gian và cải thiện chất lượng mô hình 3D đầu ra, đồng thời làm cho mô hình tập trung rõ nét vào đối tượng chính.

Việc tích hợp rembg làm bước tiền xử lý trong pipeline dựng mô hình 3D là một cải tiến đáng giá so với các quy trình truyền thống, giúp hệ thống hoạt động hiệu quả

hơn ngay cả trong điều kiện dữ liệu không lý tưởng như nền phức tạp hoặc ánh sáng không đồng đều. Đây là bước đệm quan trọng để nâng cao độ chính xác và độ nét của mô hình 3D được tạo ra từ video RGB thông thường.

2.2.3 Trích xuất và mô tả đặc trưng

Học sâu (Deep Learning) là một lĩnh vực then chốt của trí tuệ nhân tạo, đã và đang tạo ra những đột phá đáng kể trong xử lý ảnh và video. Trong đó, mạng nơ-ron tích chập là một trong những kiến trúc nổi bật nhất, với khả năng học trực tiếp từ dữ liệu và trích xuất các đặc trưng hình ảnh một cách tự động, thay thế cho các phương pháp thủ công truyền thống. CNN hoạt động bằng cách sử dụng các lớp tích chập để phát hiện đặc trưng cục bộ như cạnh, góc, hoa văn và hình dạng. Sau đó kết hợp với các lớp phi tuyến và lớp gộp để xây dựng biểu diễn không gian ngày càng trừu tượng và có tính phân biệt cao.

Mô hình SuperPoint là một mạng học sâu kết hợp giữa phát hiện điểm đặc trưng (keypoint detection) và mô tả đặc trưng (descriptor extraction). Không giống như các phương pháp cổ điển như SIFT hay ORB vốn dựa vào kỹ thuật xử lý ảnh truyền thống, SuperPoint tận dụng CNN để học biểu diễn đặc trưng mạnh mẽ hơn, cho phép phát hiện điểm đặc trưng ổn định ngay cả khi ảnh bị mờ, nhiễu hoặc ánh sáng không đồng đều. SuperPoint được huấn luyện theo phương pháp self-supervised learning, giúp mô hình không phụ thuộc vào dữ liệu gán nhãn thủ công mà vẫn đạt được hiệu quả cao và khả năng tổng quát tốt trên nhiều loại ảnh.

Trong hệ thống tái tạo 3D, các khung hình sau khi được xử lý tách nền bằng rembg sẽ được đưa vào SuperPoint để trích xuất điểm đặc trưng và vector mô tả tương ứng. Những điểm này sau đó được sử dụng làm đầu vào cho các thuật toán khớp ảnh trong OpenMVG, hỗ trợ quá trình Structure-from-Motion (SfM) nhằm tính toán vị trí camera và tái dựng cấu trúc không gian 3 chiều. Việc sử dụng SuperPoint giúp tăng độ chính xác khi khớp điểm giữa các khung hình, đặc biệt hữu ích trong các điều kiện ảnh phức tạp, nơi các thuật toán cổ điển dễ thất bại.

Về mặt triển khai kỹ thuật, SuperPoint được xây dựng bằng ngôn ngữ Python, sử dụng các thư viện học sâu như TensorFlow để định nghĩa, huấn luyện và chạy mô hình. TensorFlow hỗ trợ xử lý tensor hiệu quả và tối ưu hóa suy luận trên GPU. Bên cạnh đó, OpenCV được tích hợp để xử lý ảnh đầu vào, trực quan hóa các điểm đặc trưng và thực hiện tiền xử lý, trong khi NumPy hỗ trợ thao tác dữ liệu hiệu quả dưới dạng mảng và tensor. Sự phối hợp giữa các thư viện này tạo nên một pipeline mạnh mẽ, hiệu quả và dễ tích hợp vào các hệ thống dựng hình hiện đại.

Nhờ vào sự hỗ trợ từ CNN và các mô hình deep learning như U-2-Net và SuperPoint, hệ thống tái tạo mô hình 3D từ video không chỉ trở nên tự động hóa hơn mà còn đảm bảo chất lượng đầu ra cao hơn, đặc biệt khi phải xử lý dữ liệu đầu vào không lý tưởng như ánh sáng không đều, phong nền phức tạp hoặc chuyển động nhẹ. Đây chính là nền tảng giúp hiện thực hóa một quy trình dựng hình 3D ứng dụng thực tiễn, có thể triển khai hiệu quả ngay cả trên các thiết bị phổ thông.

2.2.4 Tái dựng cấu trúc 3D từ nhiều ảnh

OpenMVG (Multiple View Geometry) là một thư viện mã nguồn mở mạnh mẽ chuyên dụng cho Structure-from-Motion (SfM), quy trình tái tạo cấu trúc không gian 3D và ước lượng vị trí của camera từ tập hợp các ảnh 2D. Quá trình SfM là bước quan trọng trong pipeline dựng mô hình 3D, cho phép chuyển đổi các điểm ảnh 2D thành các điểm 3D trong không gian (openMVG, 2025).

Quy trình xử lý của OpenMVG trong hệ thống tái tạo mô hình 3D bắt đầu bằng việc nhận các điểm đặc trưng được phát hiện từ mô hình SuperPoint, vốn cung cấp thông tin đầu vào chất lượng cao để đảm bảo độ chính xác trong các bước tiếp theo. Sau đó, OpenMVG tiến hành khớp điểm, tức là tìm các điểm tương ứng giữa các ảnh khác nhau dựa trên đặc trưng đã trích xuất. Khi các cặp điểm khớp đã được xác định, hệ thống sẽ ước lượng ma trận camera (camera pose) bằng cách tính toán vị trí và hướng của camera tại thời điểm mỗi ảnh được chụp. Cuối cùng, dựa trên các thông tin về điểm đặc trưng và vị trí camera, OpenMVG thực hiện bước triangulation để tái dựng vị trí 3D của các điểm trong không gian. Toàn bộ quá trình này giúp xây dựng được mô hình không gian ban đầu dưới dạng đám mây điểm thưa thớt, làm nền tảng cho các bước tái tạo chi tiết tiếp theo. OpenMVG đóng vai trò then chốt trong hệ thống dựng mô hình 3D bằng cách xác định cấu trúc không gian 3D ban đầu của vật thể, cung cấp các điểm 3D thô để tiếp tục quá trình tái tạo lưới và kết xuất. Là nền tảng hình học cốt lõi trong pipeline, OpenMVG giúp tái tạo chính xác vị trí các điểm đặc trưng trong không gian và ước tính mô hình hình học một cách chính xác hơn.

OpenMVG được phát triển chủ yếu bằng ngôn ngữ lập trình C++, nhằm tối ưu hóa hiệu suất cho các tác vụ xử lý ảnh và tính toán chuyên sâu trong quy trình Structure from Motion (SfM). Để hỗ trợ hoạt động hiệu quả và linh hoạt trên nhiều nền tảng, OpenMVG sử dụng CMake như một công cụ chính để xây dựng và biên dịch mã nguồn trên các hệ điều hành khác nhau. Về mặt toán học, OpenMVG tích hợp Eigen là một thư viện mạnh mẽ cho các phép toán ma trận và vector, đóng vai trò cốt lõi trong các phép biến đổi hình học và tối ưu hóa. Bên cạnh đó, OpenCV được sử dụng để xử lý ảnh, hỗ trợ trong việc phát hiện và khớp điểm đặc trưng giữa các ảnh đầu vào. Ngoài ra, OpenMVG còn tận dụng Boost, một thư viện C++ phổ biến giúp quản lý bộ nhớ hiệu quả và hỗ trợ xử lý đa luồng, qua đó tăng tốc độ xử lý và cải thiện hiệu năng tổng thể của hệ thống.

OpenMVG là công cụ thiết yếu trong việc xây dựng cấu trúc không gian 3D từ các ảnh, đảm bảo độ chính xác trong việc xác định các điểm 3D và sự tương quan giữa các ảnh khác nhau.

2.2.5 Tái dựng bề mặt chi tiết và tạo lưới 3D

OpenMVS (Multi-View Stereo) là một thư viện mã nguồn mở dùng để tái tạo bề mặt chi tiết của vật thể 3D từ các điểm 3D và ảnh gốc được tạo ra từ quá trình Structure-from-Motion (SfM) của OpenMVG. OpenMVS tập trung vào việc tạo ra lưới 3D (mesh) mượt mà và có độ chi tiết cao từ tập điểm thưa, đồng thời gán màu sắc và

ánh sáng cho mô hình để tạo ra sản phẩm cuối cùng có thể sử dụng trong các ứng dụng trực quan hóa, nghiên cứu hoặc in 3D (cdcseacave, 2025).

Quy trình xử lý của OpenMVS đóng vai trò quan trọng trong giai đoạn tái tạo chi tiết bề mặt mô hình 3D. Đầu tiên, hệ thống tiến hành sinh bản đồ độ sâu (Depth Map) bằng cách sử dụng ảnh từ nhiều góc nhìn để tính toán độ sâu cho từng pixel. Đây là bước cốt lõi giúp suy ra thông tin không gian từ các ảnh 2D thông thường. Tiếp theo, các bản đồ độ sâu từ các góc nhìn khác nhau được kết hợp để tạo thành lưới tam giác (3D Mesh) đại diện cho hình dạng bề mặt của vật thể với độ chính xác cao. Cuối cùng, quá trình gán màu và ánh sáng được thực hiện để phủ màu sắc lên lưới tam giác, giúp mô hình 3D trở nên trực quan, sinh động và gần giống với vật thể thực tế hơn. OpenMVS đóng vai trò quan trọng trong việc hoàn thiện mô hình hóa bề mặt vật thể, giúp tái tạo các chi tiết bề mặt từ bộ khung điểm thưa do OpenMVG tạo ra. Quá trình tái dựng bề mặt chi tiết này giúp tạo ra một mô hình 3D mượt mà và thực tế hơn, phù hợp cho các ứng dụng trực quan và in 3D.

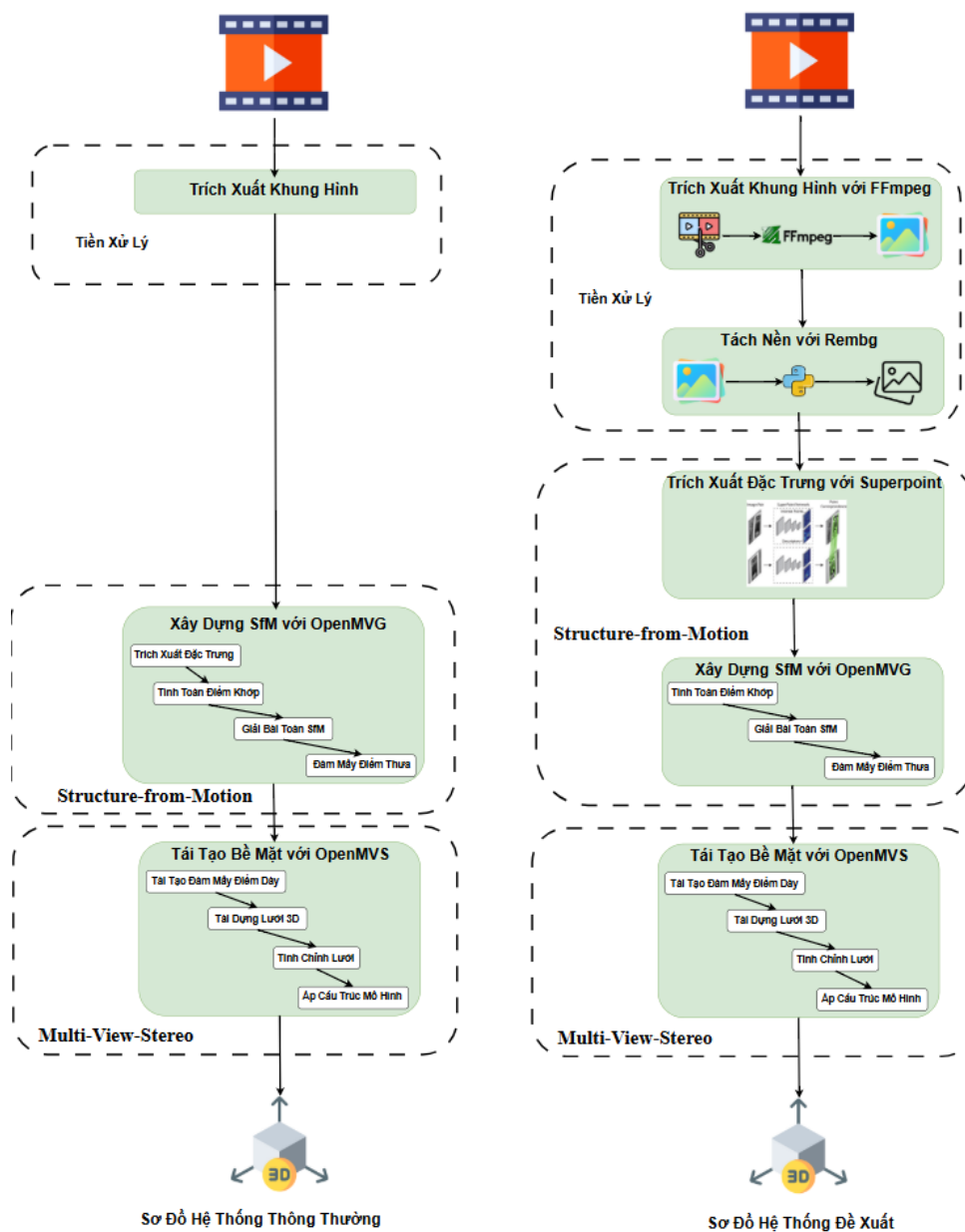
OpenMVS được phát triển chủ yếu bằng ngôn ngữ lập trình C++, nhằm đảm bảo hiệu suất tối ưu khi xử lý các phép toán hình học phức tạp trong quá trình tái dựng mô hình 3D. Hệ thống này tích hợp nhiều thư viện và công cụ hỗ trợ mạnh mẽ. CMake được sử dụng để xây dựng và biên dịch phần mềm trên nhiều nền tảng khác nhau. Eigen đảm nhiệm vai trò xử lý các phép toán ma trận và vector trong quá trình tạo lưới và tính toán độ sâu. OpenCV hỗ trợ việc xử lý ảnh đầu vào, đặc biệt trong khâu tạo bản đồ độ sâu. Bên cạnh đó, Boost góp phần tăng tốc xử lý thông qua khả năng quản lý đa luồng và bộ nhớ hiệu quả. Đối với giao diện, Qt được sử dụng để trực quan hóa kết quả và hỗ trợ hiển thị mô hình. Quá trình tối ưu hóa hình học dựa trên Ceres Solver, vốn chuyên giải các bài toán tối ưu phi tuyến. CGAL cung cấp các thuật toán hình học tính toán như dựng lưới và xử lý bề mặt. Trong khi đó, VCG được dùng để quản lý, xử lý và lưu trữ các mô hình lưới chi tiết. Với mục tiêu tăng tốc, OpenMVS cũng hỗ trợ CUDA của NVIDIA để thực hiện tính toán song song trên GPU. Cuối cùng, GLFW đóng vai trò hỗ trợ quản lý cửa sổ và nhập liệu, giúp người dùng tương tác trực tiếp và trực quan với mô hình 3D.

Tích hợp với hệ thống, OpenMVS nhận dữ liệu từ OpenMVG (đầu ra là các điểm 3D từ quá trình SfM) và thực hiện tái dựng bề mặt chi tiết, giúp tạo ra một mô hình 3D mượt mà và chi tiết hơn, hỗ trợ cho các bước cuối trong pipeline 3D, bao gồm xuất mô hình và trực quan hóa.

CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ GIẢI THUẬT

3.1 KIẾN TRÚC TỔNG THỂ CỦA HỆ THỐNG

Quy trình tái tạo 3D chuyển đổi một chuỗi hình ảnh 2D, thường là từ video thành một mô hình 3D của cảnh hoặc đối tượng được ghi lại. Quá trình này bao gồm nhiều giai đoạn, mỗi giai đoạn xây dựng dựa trên kết quả của giai đoạn trước. Việc hiểu rõ từng bước riêng lẻ và sự phụ thuộc lẫn nhau giữa chúng là rất quan trọng. Mỗi bước đóng một vai trò thiết yếu trong việc đảm bảo chất lượng và độ chính xác tổng thể của mô hình 3D cuối cùng. Việc nắm bắt được những sự phụ thuộc này cho phép đưa ra các quyết định sáng suốt về cài đặt tham số và các tối ưu hóa tiềm năng.



Hình 3.1 Quy trình tổng thể của hệ thống

Hệ thống tái dựng mô hình 3D từ video được xây dựng qua chuỗi các bước xử lý chính. Đầu tiên, video đầu vào được chuyển thành chuỗi ảnh tĩnh bằng FFmpeg. Sau đó, mỗi ảnh được tách nền bằng rembg (sử dụng mô hình U-2-Net) để làm nổi bật vật thể chính. Tiếp theo, đặc trưng của ảnh được trích xuất bằng SuperPoint hoặc các phương pháp truyền thống như SIFT, ORB nhằm xác định các điểm keypoints. Dữ liệu đặc trưng này được đưa vào OpenMVG để xây dựng cấu trúc chuyển động (SfM), ước lượng vị trí camera và tạo đám mây điểm thưa thớt. OpenMVS tiếp tục tái tạo bề mặt chi tiết, tạo ra đám mây điểm dày đặc hoặc lưới tam giác. Cuối cùng, mô hình 3D được xuất ra dưới định dạng phổ biến như PLY, có thể xem bằng các phần mềm như MeshLab.

Bước	Mô tả	Thư viện và Công cụ	Kết quả dự kiến
Trích xuất khung hình	Chuyển đổi video thành một loạt các hình ảnh tĩnh đều nhau.	FFmpeg	Một chuỗi các hình ảnh tĩnh từ video.
Tách nền	Loại bỏ nền các khung hình, giữ lại đối tượng chính để tập chung vào việc tái tạo vật thể.	rembg (U-2-Net)	Hình ảnh đã loại bỏ nền (trong suốt hoặc màu nền trắng).
Trích xuất đặc trưng	Xác định các điểm đặc biệt (keypoints) trong mỗi hình ảnh bằng mô hình deep learning hoặc SIFT, ORB.	SuperPoint, SIFT, ORB	Một tập hợp các keypoints và các mô tả của chúng cho mỗi hình ảnh.
Xây dựng SfM	Ước tính vị trí camera và tạo đám mây điểm ảnh 3D thưa thớt.	OpenMVG	Một đám mây điểm 3D thưa thớt và các tham số camera ước tính.
Tái tạo bề mặt	Làm dày đám mây điểm thưa thớt và tái dựng bề mặt chi tiết và kết xuất lưới tam giác 3D.	OpenMVS	Một đám mây điểm 3D dày đặc hoặc một lưới đa giác.
Kết xuất mô hình 3D	Xuất mô hình 3D cuối cùng ở định dạng tiêu chuẩn (.ply).	Các phần mềm để xem mô hình 3D như meshlab.	Một mô hình 3D có kết cấu ở định dạng như PLY hoặc STL.

Bảng 3.1 Quy trình và Công nghệ sử dụng của hệ thống đề xuất

3.2 TRÍCH XUẤT KHUNG HÌNH TỪ VIDEO VỚI FFMPEG

Việc trích xuất khung hình từ video là cần thiết vì nhiều thuật toán tái tạo 3D hoạt động trên hình ảnh tĩnh. Video, vốn là một chuỗi các khung hình, cần được chia thành các hình ảnh riêng lẻ để xử lý. Điều này cho phép phân tích cảnh từ các góc nhìn rời rạc được ghi lại tại các thời điểm khác nhau. Ffmpeg là một công cụ dòng lệnh mạnh mẽ, được sử dụng chủ yếu cho bước này, để xử lý các tệp đa phương tiện, bao gồm cả video. Nó cung cấp một loạt các chức năng để trích xuất khung hình, chuyển đổi định dạng và áp dụng các bộ lọc. Tính chất đa nền tảng và tính linh hoạt của nó làm cho nó trở thành một lựa chọn phổ biến.

Kết quả của bước này là một thư mục chứa một chuỗi các tệp hình ảnh tĩnh (.jpg hoặc .png) tương ứng với các khung hình của video đầu vào. Số lượng hình ảnh sẽ phụ thuộc vào thời lượng của video và tốc độ trích xuất đã chọn. Việc lựa chọn các tham số trích xuất khung hình có tác động đáng kể đến các bước tiếp theo. Tốc độ khung hình cao hơn mang lại dữ liệu chi tiết hơn nhưng làm tăng thời gian xử lý và yêu cầu lưu trữ. Ngược lại, tốc độ khung hình thấp hơn có thể bỏ lỡ các chi tiết quan trọng.

3.3 TÁCH NỀN VỚI REMBG

Việc tách nền trước khi xử lý giúp tăng độ chính xác và hiệu quả cho các bước như trích xuất và khớp đặc trưng, đồng thời giảm nhiễu và tải tính toán bằng cách tập trung vào vùng đối tượng chính. Thư viện rembg sử dụng mô hình học sâu U-2-Net để loại bỏ nền khỏi hình ảnh. U-2-Net có kiến trúc U lồng nhau với các khối Residual U, giúp trích xuất đặc trưng đa cấp và đa tỷ lệ, từ đó phân đoạn chính xác đối tượng tiền cảnh khỏi nền phức tạp. Kết quả của bước này là một tập hợp các hình ảnh mà nền đã được loại bỏ. Các hình ảnh này có thể có nền trong suốt hoặc nền màu tùy thuộc vào các tham số được sử dụng. Hiệu quả của việc loại bỏ nền ảnh hưởng trực tiếp đến chất lượng của việc trích xuất đặc trưng. Nếu việc loại bỏ nền không chính xác, các đặc trưng không liên quan vẫn có thể được phát hiện, dẫn đến việc khớp sai ở giai đoạn SfM. Chất lượng của mặt nạ được tạo bởi rembg là rất quan trọng cho bước tiếp theo.



Hình 3.2 Ảnh trước khi xóa nền



Hình 3.3 Ảnh sau khi xóa nền

3.4 TRÍCH XUẤT ĐẶC TRƯNG VỚI SUPERPOINT

3.4.1 Tổng quan SuperPoint

Trong quy trình dựng mô hình 3D, trích xuất đặc trưng là bước then chốt để xác định các điểm tương ứng giữa các ảnh, bao gồm phát hiện keypoints và sinh mô tả bất biến với góc nhìn, tỷ lệ và ánh sáng. Việc khớp các mô tả này giúp tái tạo điểm 3D. SuperPoint là mô hình học sâu hiện đại thực hiện đồng thời phát hiện keypoints và sinh mô tả trong một lần suy luận. Nó gồm bộ mã hóa chia sẻ và hai đầu ra, một cho bản đồ keypoint, một cho mô tả đặc trưng. Keypoints được chọn qua ức chế không tối đa (NMS), còn mô tả lấy từ bản đồ đặc trưng dày đặc bằng nội suy. So với các phương pháp truyền thống, SuperPoint cho hiệu quả vượt trội trong điều kiện khó như ít kết cấu hoặc thay đổi góc nhìn lớn. Khả năng xử lý đồng thời keypoint và mô tả giúp nó trở thành lựa chọn hiệu quả cho các hệ thống tái tạo 3D hiện đại.

Mục tiêu của việc so sánh là đánh giá và so sánh khả năng ước lượng phép biến đổi homography giữa các thuật toán mô tả đặc trưng phổ biến, bao gồm SuperPoint, SIFT và ORB. Homography là phép biến đổi hình học cho phép ánh xạ một ảnh sang ảnh khác trong trường hợp cả hai cùng nằm trên một mặt phẳng, thường được sử dụng trong các bài toán như ghép ảnh (image stitching) hoặc tái dựng mặt phẳng. Để thực hiện so sánh một cách khách quan, bộ dữ liệu HPatches được sử dụng, đây là một bộ dữ liệu tiêu chuẩn gồm các cặp ảnh có sự thay đổi về điều kiện ánh sáng và góc nhìn, rất phù hợp để kiểm tra độ ổn định và độ chính xác của các thuật toán mô tả đặc trưng trong thực tế.

Đánh giá theo loại biến đổi (Rpautrat, 2025)

Thuật toán	Illumination changes	Viewpoint changes
SuperPoint (tự triển khai)	0.965	0.712
SuperPoint (MagicLeap)	0.923	0.742
SIFT	0.807	0.766
ORB	0.523	0.414

Bảng 3.2 Đánh giá theo loại biến đổi

Giá trị trong thí nghiệm thể hiện tỷ lệ khớp chính xác khi ước lượng homography giữa hai ảnh, đóng vai trò như một chỉ số đánh giá mức độ hiệu quả của các mô tả đặc trưng. Giá trị này càng cao thì khả năng mô tả và khớp đặc trưng của thuật toán càng tốt, từ đó đảm bảo độ chính xác cao hơn trong các bài toán thị giác máy tính như tái dựng 3D hay ghép ảnh.

Trong quá trình đánh giá hiệu quả các phương pháp trích xuất đặc trưng, kết quả thực nghiệm cho thấy SuperPoint đạt độ chính xác cao nhất khi ảnh bị thay đổi về ánh sáng, chứng minh khả năng chống nhiễu sáng vượt trội. Tuy nhiên, trong trường hợp thay đổi góc nhìn, SIFT lại hoạt động ổn định hơn nhờ khả năng mô tả đặc trưng mạnh

dưới các biến dạng hình học. Ngược lại, ORB cho kết quả kém nhất ở cả hai điều kiện, cho thấy hạn chế rõ rệt về độ chính xác và độ tin cậy khi áp dụng trong các môi trường biến đổi phức tạp.

Đánh giá theo nhiều ngưỡng sai số homography (Rpautrat, 2025)

Correctness threshold (sai số cho phép e)	e = 1	e = 3	e = 5
SuperPoint (tự triển khai)	0.483	0.836	0.910
SuperPoint (MagicLeap)	0.438	0.833	0.914
SIFT	0.498	0.786	0.786
ORB	0.162	0.467	0.564

Bảng 3.3 Đánh giá theo nhiều ngưỡng sai số homography

Ngưỡng độ chính xác (Correctness threshold) e là một tham số quan trọng được sử dụng để đánh giá độ lệch cho phép giữa ma trận homography dự đoán và ma trận homography thực tế. Khi sai số vị trí của các điểm khớp nhỏ hơn một giá trị nhất định (ví dụ dưới 2 pixel), điểm đó được xem là khớp chính xác. Cụ thể, khi $e=1$ thì ngưỡng đánh giá rất nghiêm ngặt yêu cầu sai số cực thấp; với $e=3$ thì mức đánh giá ở mức trung bình còn khi $e=5$ thì tiêu chí được nói lỏng hơn, chấp nhận sai số lớn hơn. Giá trị độ chính xác dưới các ngưỡng này phản ánh tỷ lệ điểm khớp đúng tương ứng với mức độ sai số cho phép. Từ kết quả thực nghiệm, SuperPoint cho thấy hiệu suất tổng thể vượt trội ở các ngưỡng sai số trung bình và nói lỏng ($e = 3$ và $e = 5$), chứng minh khả năng khớp đặc trưng ổn định trong điều kiện sai số cho phép cao hơn. Trong khi đó, SIFT tỏ ra chính xác hơn một chút khi yêu cầu độ chính xác nghiêm ngặt với sai số rất nhỏ ($e = 1$). ORB lại thể hiện hiệu quả thấp nhất trong mọi trường hợp, cho thấy hạn chế rõ rệt về độ chính xác và độ ổn định trong quá trình khớp đặc trưng.

Sau khi thực hiện so sánh cho thấy SuperPoint là một phương pháp hiện đại với độ chính xác cao và khả năng khớp ổn định, đặc biệt hiệu quả khi ngưỡng sai số được nói lỏng. Đáng chú ý, phiên bản SuperPoint được tự huấn luyện thậm chí có thể vượt qua cả mô hình pretrained như MagicLeap. Trong khi đó, SIFT vẫn thể hiện sức mạnh vượt trội trong các tình huống yêu cầu độ chính xác cao với sai số rất nhỏ và khi có sự thay đổi lớn về góc nhìn. ORB tuy có ưu điểm về tốc độ và độ nhẹ, nhưng độ chính xác thấp, do đó chỉ phù hợp với các ứng dụng thời gian thực đơn giản và không đòi hỏi cao về chất lượng khớp đặc trưng. Mục đích của SuperPoint là một mô hình học sâu sử dụng mạng nơ-ron tích chập (CNN) để phát hiện điểm đặc trưng (keypoints) và tạo vector mô tả (descriptors). Hai thành phần này đóng vai trò nền tảng cho các bước xử lý tiếp theo như khớp đặc trưng và ước lượng homography.

SuperPoint bao gồm ba phần quan trọng, đầu tiên là VGGBlock một khối mạng CNN cơ bản kết hợp các lớp Conv2D, ReLU (tùy chọn) và BatchNorm để trích xuất đặc trưng từ ảnh đầu vào. Thứ hai là SuperPoint class, mô hình chính kết hợp ba thành phần gồm backbone (mạng trích xuất đặc trưng), detector (phát hiện keypoints), và descriptor (tạo vectors mô tả). Bên cạnh đó, một số hàm hỗ trợ đóng vai trò quan trọng

trong quá trình xử lý đặc trưng, bao gồm `sample_descriptors` dùng để nội suy vector đặc trưng tại vị trí các keypoints nhằm đảm bảo tính chính xác cao trong biểu diễn. Ngoài ra còn có `batched_nms` thực hiện Non-Maximum Suppression để loại bỏ các keypoints trùng lặp hoặc kém nổi bật và `select_top_k_keypoints` giúp chọn ra những điểm đặc trưng đáng tin cậy nhất dựa trên giá trị scores, từ đó cải thiện hiệu quả khớp hình ảnh.

Đầu ra của mô hình SuperPoint bao gồm Keypoints, Descriptors và Scores, mỗi thành phần đều mang ý nghĩa riêng biệt và quan trọng trong quá trình trích xuất và khớp đặc trưng giữa các ảnh.

Keypoints là tọa độ của các điểm đặc trưng được phát hiện trong ảnh, được biểu diễn dưới dạng danh sách các tensor PyTorch, mỗi tensor có kích thước $(N_i, 2)$, trong đó N_i là số keypoints trong ảnh thứ i . Mỗi điểm là một cặp giá trị thực (x, y) , thể hiện vị trí pixel cụ thể trong ảnh, đại diện cho các vùng nổi bật và ổn định dưới các biến đổi như thay đổi ánh sáng và góc nhìn. Keypoints có dạng `[[123.45, 234.67], [345.12, 156.89], ...]`. Các tọa độ này không bao gồm thông tin về kích thước (scale) hay hướng (orientation) của điểm đặc trưng.

Descriptors là mô tả cục bộ xung quanh từng keypoint, dùng để so khớp giữa các ảnh khác nhau. Đầu ra là danh sách các tensor PyTorch, mỗi tensor có kích thước $(N_i, 256)$, trong đó mỗi dòng là một vector đặc trưng 256 chiều. Ví dụ như `[[0.123, 0.456, ..., 0.234], [0.567, 0.890, ..., 0.678], ...]`. Các vector này là số thực và đã được chuẩn hóa theo chuẩn L2 (tổng bình phương các thành phần bằng 1), giúp tăng độ ổn định và khả năng phân biệt. Với độ dài 256 chiều nhiều hơn SIFT thường chỉ có 128, mô tả của SuperPoint có khả năng biểu diễn đặc trưng chi tiết và phong phú hơn.

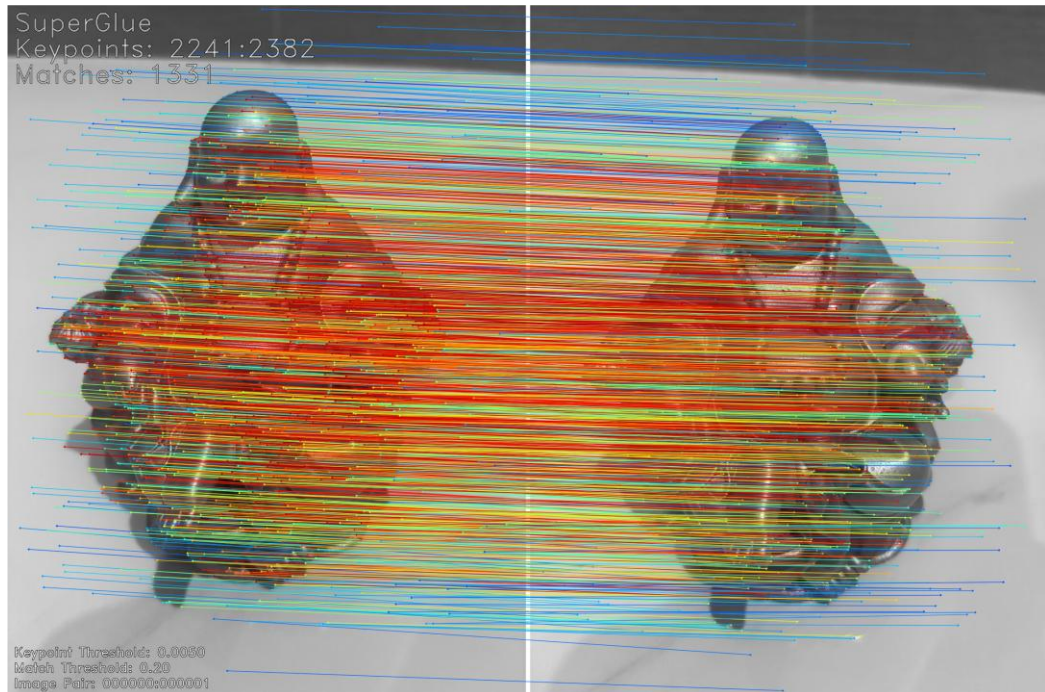
Scores là chỉ số thể hiện độ tin cậy của mỗi keypoint, được lưu dưới dạng danh sách các tensor PyTorch có kích thước $(N_i,)$, trong đó mỗi phần tử là một giá trị số thực đại diện cho mức độ nổi bật của điểm đặc trưng. Ví dụ như `[0.95, 0.87, 0.92, ...]`. Những giá trị này có thể được dùng để lọc ra các điểm đặc trưng đáng tin cậy nhất bằng cách chọn top-k điểm có score cao nhất, qua đó tăng chất lượng quá trình khớp đặc trưng và giảm nhiễu.

Lưu ý, SuperPoint không tự động tạo matches (các cặp điểm khớp giữa các ảnh). Cần thực hiện bước khớp đặc trưng riêng. Đầu ra thường được lưu dưới dạng NumPy arrays (.npy) hoặc file .txt hoặc .json nếu lưu thủ công.

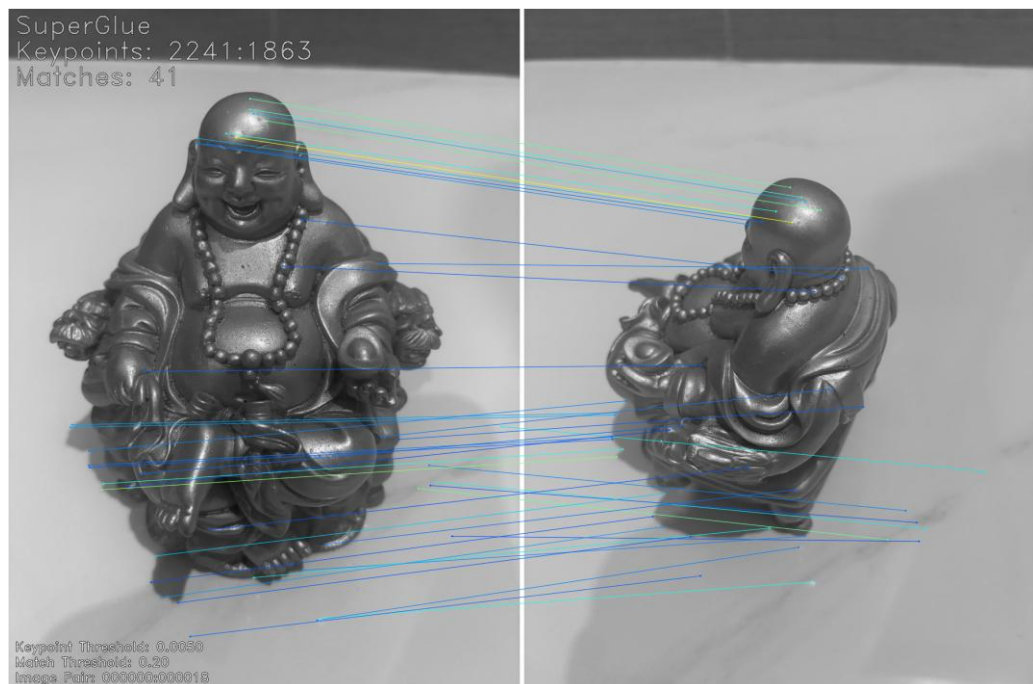
Ví dụ kết quả đầu ra

```
{
  'keypoints': [ torch.tensor([[123.45, 234.67], [345.12, 156.89], ...]), ... ],
  'keypoint_scores': [ torch.tensor([0.95, 0.87, ...]), ... ],
  'descriptors': [ torch.tensor([[0.123, 0.456, ..., 0.234], ...]), ... ]
}
```

Kết quả khớp đặc trưng



Hình 3.4 Khớp đặc trưng giữa ảnh 1 và ảnh 2



Hình 3.5 Khớp đặc trưng giữa ảnh 1 và ảnh 19

3.4.2 Sự khác biệt giữa đầu ra SuperPoint và đầu vào OpenMVG

Do SuperPoint là một thành phần được nghiên cứu và tích hợp thêm vào nhằm cải thiện hiệu quả của hệ thống, nên dữ liệu đầu ra của nó có định dạng khác so với định dạng đầu vào chuẩn mà OpenMVG yêu cầu. Do đó, cần thực hiện bước chuyển đổi định dạng để đảm bảo khả năng tương thích giữa hai thành phần này.

Thành phần	SuperPoint (Đầu ra)	OpenMVG (Đầu vào)	Khác biệt
Keypoints	Tensor (N_i , 2) chứa $[x, y]$ (float). Không có scale, orientation.	File .feat chứa $[x, y, \text{scale}, \text{orientation}]$. Scale và orientation thường cần thiết.	SuperPoint thiếu scale và orientation. OpenMVG yêu cầu định dạng file cụ thể.
Descriptors	Tensor (N_i , 256) chứa vector 256 chiều (float, chuẩn hóa L2).	Vector trong file .feat, thường 128 chiều (SIFT) hoặc tùy chỉnh, lưu cùng keypoints.	SuperPoint dùng 256 chiều, OpenMVG thường dùng 128 chiều.
Scores	Tensor (N_i) chứa độ tin cậy (float).	Không yêu cầu.	Scores không được sử dụng trong OpenMVG, có thể bỏ qua.
Matches	Không có, phải tạo riêng (dùng OpenCV).	File .matches.f chứa các cặp chỉ số keypoints giữa các ảnh.	SuperPoint không tạo matches, OpenMVG yêu cầu matches.
Định dạng lưu trữ	Tensor PyTorch hoặc NumPy arrays (.npy), có thể lưu thành file văn bản thủ công.	File .feat (keypoints + descriptors) và .matches.f (matches) ở định dạng văn bản.	SuperPoint lưu dạng linh hoạt, OpenMVG yêu cầu định dạng file cụ thể.

Bảng 3.4 Đầu ra của SuperPoint và đầu vào của OpenMVG

Để sử dụng đầu ra từ SuperPoint trong các bước xử lý tiếp theo của pipeline OpenMVG, cần thực hiện chuyển đổi dữ liệu để đảm bảo tương thích về mặt định dạng và cấu trúc. Việc tích hợp này bao gồm các bước xử lý keypoints, descriptors, khớp đặc trưng, và lưu trữ dữ liệu theo chuẩn của OpenMVG.

Xử lý Keypoints, SuperPoint chỉ cung cấp tọa độ của các điểm đặc trưng dưới dạng $[x, y]$, tuy nhiên lại không bao gồm thông tin về scale (tỉ lệ) và orientation (hướng). Đây là một thiếu sót nếu muốn tích hợp với các hệ thống như OpenMVG, vốn có thể yêu cầu đầy đủ ba yếu tố này. Để khắc phục, ta gán giá trị mặc định cho hai thuộc tính bị thiếu là $\text{scale} = 1$ và $\text{orientation} = 0$. Sau đó, các tọa độ $[x, y]$ được chuyển

đổi từ tensor sang định dạng file .feat, giúp dễ dàng lưu trữ và tích hợp vào pipeline xử lý tiếp theo.

Xử lý Descriptors, SuperPoint tạo ra các vector đặc trưng (descriptors) có kích thước 256 chiều, trong khi đó OpenMVG thường yêu cầu descriptors với kích thước 128 chiều. Điều này gây ra sự không tương thích. Một giải pháp đơn giản là sử dụng PCA (Principal Component Analysis) để giảm chiều từ 256 xuống 128 trong giai đoạn tiền xử lý. Ngoài ra, nếu có quyền can thiệp vào mô hình, có thể thêm một tầng linear (fully connected) ở cuối kiến trúc SuperPoint để chuyển trực tiếp từ 256 về 128 chiều trước khi xuất descriptors.

Tạo Matches, SuperPoint không tạo các cặp điểm khớp (matches) giữa các ảnh – một bước bắt buộc trong quá trình xử lý Structure-from-Motion (SfM) của OpenMVG. Do đó, ta cần áp dụng một thuật toán khớp đặc trưng để so sánh các descriptors giữa các cặp ảnh. Một lựa chọn phổ biến là sử dụng Brute-Force Matching kèm với kiểm tra tỷ lệ của Lowe để loại bỏ các khớp không chắc chắn. Với dữ liệu lớn, có thể dùng FLANN để tăng tốc quá trình tìm kiếm. Kết quả cuối cùng là các cặp điểm tương ứng giữa các ảnh, sẽ được lưu vào file .matches.f, phục vụ cho các bước xử lý tiếp theo trong pipeline OpenMVG.

Bảng tóm tắt cách khắc phục cụ thể

Khác biệt	Cách khắc phục
Thiếu scale và orientation	Gán scale=1.0, orientation=0.0 trong file .feat.
Descriptors 256 chiều thay vì 128 chiều	Giữ nguyên 256 chiều hoặc giảm còn 128 bằng PCA hoặc linear layer.
Thiếu matches	Tạo bằng OpenCV (Brute-Force hoặc FLANN), lưu vào file .matches.f.
Định dạng lưu trữ	Viết script Python chuyển từ tensor sang file .feat và .matches.f theo chuẩn OpenMVG.

Bảng 3.5 Tóm tắt cách khắc phục

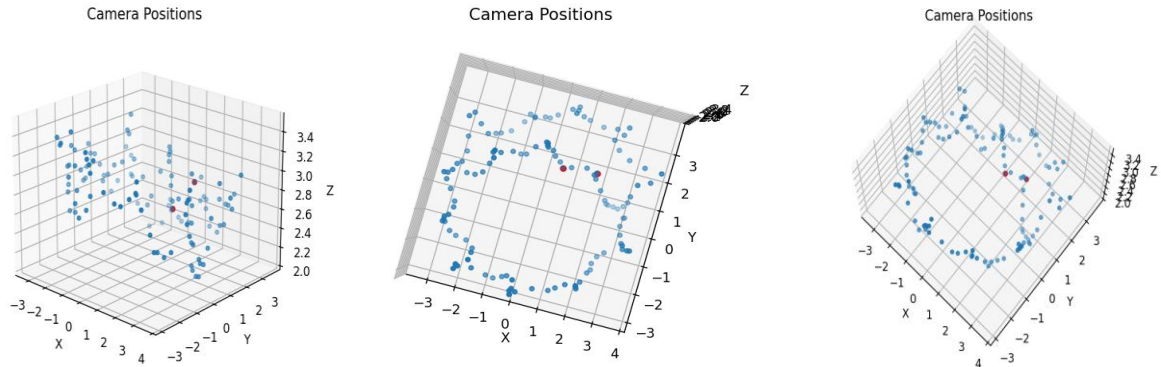
Kết quả của bước này là đối với mỗi hình ảnh đầu vào, sẽ có một tập hợp các keypoints được phát hiện (được biểu diễn bằng tọa độ pixel của chúng) và một vector mô tả tương ứng cho mỗi keypoint. Chúng thường được lưu trữ ở định dạng có cấu trúc có thể được sử dụng trong giai đoạn khớp đặc trưng tiếp theo. Việc lựa chọn bộ trích xuất đặc trưng ảnh hưởng đáng kể đến số lượng và chất lượng của các điểm khớp được tìm thấy giữa các hình ảnh. Các điểm khớp chính xác và mạnh mẽ hơn ở giai đoạn này dẫn đến việc tái tạo SfM đáng tin cậy hơn. Tuy nhiên do đầu ra của superpoint tương đối khác với đầu vào của OpenMVG nên cần quá trình tùy chỉnh kết quả cho phù hợp.

3.5 XÂY DỰNG SFM VỚI OPENMVG

Structure-from-Motion (SfM) là một kỹ thuật trong thị giác máy tính nhằm tái dựng cấu trúc hình học 3D của một cảnh từ nhiều ảnh 2D chụp ở các góc nhìn khác

nhau. SfM không chỉ xây dựng đám mây điểm thưa mà còn ước lượng được vị trí hướng của camera và tọa độ không gian 3D của các điểm đặc trưng trong cảnh.

Vị trí camera được trích xuất từ video



Hình 3.6 Vị trí camera được trích xuất từ video

3.5.1 Khởi tạo dữ liệu

Bước khởi đầu trong quy trình tái tạo mô hình 3D là tạo ra tệp cấu hình SfM (sfm_data.json) từ thư mục ảnh đầu vào. Quá trình này bao gồm các công việc chính như đọc siêu dữ liệu của ảnh (tên tệp, kích thước, các thông số nội tại của camera như tiêu cự, điểm ảnh chính giữa và độ méo ống kính), nhóm các ảnh có cùng thông số nội tại thành từng thiết bị để tối ưu hóa quá trình xử lý, và cuối cùng là sinh ra tệp sfm_data.json, đây là dữ liệu đầu vào quan trọng cho các bước tiếp theo.

Ví dụ 2 thành phần chính trong file sfm_data.json

```
"views": [
  { "key": 0,
    "value": { "filename": "0001.jpg", "width": 720, "height": 1280, "id_intrinsic": 0,
    "id_pose": 0 }
  },
  ...
],
"intrinsics": [
  { "key": 0,
    "value":
      { "polymorphic_name": "pinhole_radial_k3", "focal_length": 600.0,
    "principal_point": [360.0,
      640.0], "disto_k3": [0.0, 0.0, 0.0] }
  }
]
```

Trong hệ thống OpenMVG, thành phần Views đại diện cho các ảnh đầu vào được trích xuất từ video. Mỗi ảnh được lưu trữ kèm theo các thuộc tính kỹ thuật quan trọng. Cụ thể, thuộc tính filename như "0001.jpg" cho biết tên tệp ảnh. Các thông số width và height lần lượt chỉ ra chiều rộng và chiều cao của ảnh, ví dụ như 720 x 1280 pixel. Ngoài ra, mỗi ảnh còn liên kết đến hai chỉ số là id_intrinsic dùng để chỉ định thông số nội tại của camera, và id_pose biểu thị tư thế (pose) của camera sẽ được ước lượng trong quá trình tái dựng 3D.

Bên cạnh đó, Intrinsics là thành phần mô tả các thông số nội tại của camera – tức các đặc điểm cố định của hệ thống quang học ảnh hưởng đến việc chụp hình. Trong ví dụ, polymorphic_name được thiết lập là "pinhole_radial_k3", nghĩa là hệ thống sử dụng mô hình camera pinhole với biến dạng thấu kính bậc ba. Thông số focal_length là 600.0 pixel, biểu thị tiêu cự của ống kính. principal_point được đặt là [360.0, 640.0], tương ứng với điểm ảnh trung tâm của cảm biến. Cuối cùng, disto_k3 chứa các hệ số biến dạng thấu kính, và trong trường hợp này có giá trị [0.0, 0.0, 0.0], tức là không có hiện tượng méo hình ảnh.

Việc tách riêng Views và Intrinsics cho phép hệ thống tái sử dụng các thông số camera cho nhiều ảnh khác nhau và tính toán chính xác các bước trong pipeline Structure from Motion. Đây là một phần quan trọng trong việc tái dựng mô hình 3D từ ảnh tĩnh hoặc video.

3.5.2 Tính toán đặc trưng

Ở bước phát hiện đặc trưng, hệ thống thu được hai loại tệp đầu ra quan trọng gồm tệp .feat và .desc. Trong đó tệp .feat lưu trữ thông tin chi tiết về các điểm đặc trưng (keypoints) được phát hiện trong ảnh, bao gồm tọa độ (x, y), tỉ lệ (scale) và hướng (orientation). Trong khi đó, tệp .desc chứa các descriptor – là các vector đặc trưng đại diện cho vùng lân cận quanh từng keypoint, được sử dụng để so khớp giữa các ảnh trong các bước xử lý tiếp theo.

Việc sử dụng chế độ ULTRA giúp cải thiện đáng kể mật độ và chất lượng của các điểm đặc trưng, từ đó nâng cao độ chính xác trong khớp ảnh và tái cấu trúc mô hình. Khi tích hợp thêm thuật toán như SuperPoint hoặc sử dụng mặt nạ để loại bỏ các vùng nền không cần thiết, hệ thống cho ra kết quả rõ nét hơn, giàu thông tin hơn và tối ưu cho quá trình tái tạo mô hình 3D.

❖ Ví dụ file .feat

301.094 195.952 0.900794 1.13831
...
407.727 966.302 40.0226 1.16295

Trong đó các số lần lượt là: tọa độ x, tọa độ y, scale, hướng

3.5.3 Tính toán điểm khớp

Kết quả của bước so khớp đặc trưng là hai tệp dữ liệu quan trọng gồm `matches.putative.bin` và `matches.f.bin`. Trong đó tệp `matches.putative.bin` chứa danh sách các cặp điểm đặc trưng được khớp giữa các ảnh dựa trên độ tương đồng giữa các descriptor, phản ánh các ghép nối tiềm năng nhưng chưa được xác thực về mặt hình học. Trong khi đó, `matches.f.bin` là kết quả đã được lọc hình học, chỉ giữ lại các khớp chính xác thông qua các ràng buộc như epipolar hoặc mô hình ma trận cơ bản.

Việc áp dụng tiêu chí lọc tỷ lệ khoảng cách gần nhất và các phương pháp lọc hình học đã giúp loại bỏ các ghép nối không chắc chắn, góp phần nâng cao độ tin cậy của các cặp điểm tương ứng. Nhờ đó, dữ liệu đầu ra từ bước này có chất lượng cao hơn, sẵn sàng phục vụ cho các bước tái dựng cấu trúc 3D tiếp theo.

3.5.4 Giải bài toán SfM

Đây là bước quan trọng nhất trong quy trình, xây dựng mô hình 3D thưa (sparse) và ước lượng chuyển động của camera thông qua thuật toán SfM tăng dần (Incremental SfM). Quá trình bắt đầu bằng việc khởi tạo từ một cặp ảnh ban đầu (seed pair) có nhiều điểm khớp tin cậy. Sau đó, từng ảnh mới sẽ được thêm vào bằng cách ước lượng pose (vị trí và hướng) của camera, tam giác hóa (triangulate) các điểm 3D mới từ các điểm khớp, và thực hiện điều chỉnh bó (Bundle Adjustment) để tối ưu hóa toàn bộ mô hình. Quá trình này lặp lại cho đến khi tất cả ảnh đều được tích hợp vào mô hình 3D.

Đầu ra của bước SfM bao gồm hai tệp chính là `sfm_data.bin` chứa toàn bộ thông tin tái dựng như pose của các camera, cấu trúc 3D của cảnh (các điểm không gian) và thông số nội tại (intrinsics); và `cloud_and_poses.ply` là tệp đám mây điểm thưa (sparse point cloud) kèm theo vị trí các camera, có thể sử dụng để trực quan hóa mô hình trong các phần mềm như Meshlab.

3.6 TÁI TẠO BỀ MẶT VỚI OPENMVS

Giai đoạn MVS trong hệ thống sử dụng là OpenMVS để tái tạo mô hình 3D dày đặc từ đám mây điểm thưa thớt và thông tin camera do SfM cung cấp. OpenMVS khai thác tính nhất quán giữa các hình ảnh để ước lượng độ sâu chính xác, từ đó xây dựng đám mây điểm chi tiết, tạo lưới tam giác và kết cấu mô hình. Chất lượng đầu ra phụ thuộc lớn vào độ chính xác của SfM; nếu vị trí camera sai hoặc đặc trưng kém, mô hình có thể bị lỗi. Ngoài ra, việc loại bỏ nền bằng `rembg` giúp hệ thống tập trung vào đối tượng cần tái tạo, tăng độ chính xác và giảm nhiễu từ các yếu tố không liên quan. Kết quả là một mô hình 3D giàu chi tiết, phục vụ hiệu quả cho các mục tiêu trực quan hóa.

3.6.1 Chuyển đổi dữ liệu từ OpenMVG sang OpenMVS

Sau khi hoàn tất quá trình tái tạo mô hình 3D thưa bằng OpenMVG, bước tiếp theo là chuyển đổi kết quả sang định dạng mà OpenMVS có thể sử dụng để thực hiện tái tạo điểm dày đặc. Việc này được thực hiện với nhiệm vụ chính là chuyển đổi toàn bộ dữ liệu tái dựng từ OpenMVG sang định dạng của OpenMVS. Cụ thể, việc này

chuyển đổi các thông số nội tại và ngoại tại của camera, đám mây điểm thưa, cùng với liên kết đến các ảnh đầu vào giúp OpenMVS có thể tiếp tục quá trình dựng mô hình 3D chi tiết.

Đầu ra của bước chuyển đổi là tệp `scene.mvs`, đây là tệp trung tâm trong pipeline của OpenMVS. Nó chứa toàn bộ dữ liệu cần thiết để tiếp tục quá trình tái dựng chi tiết, bao gồm thông tin về camera, đám mây điểm thưa và đường dẫn tới các ảnh đầu vào. Tệp này đóng vai trò là điểm khởi đầu cho các bước xử lý tiếp theo như tái tạo điểm dày đặc, dựng lưới và tạo kết cấu bề mặt.

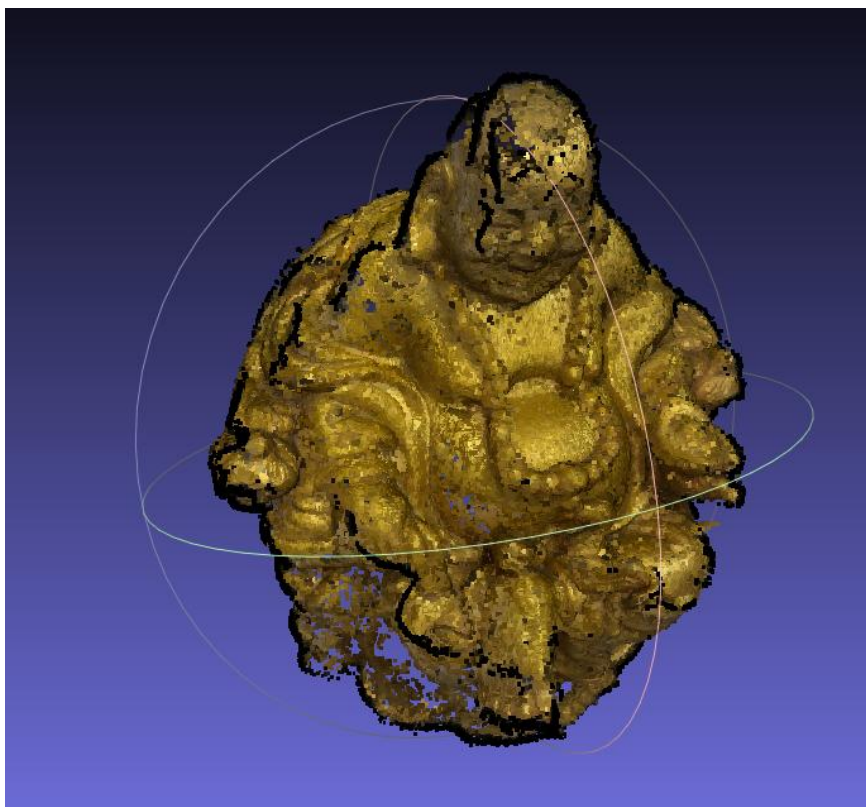
Vai trò của bước chuyển đổi dữ liệu là làm cầu nối quan trọng giữa quá trình SfM sử dụng OpenMVG và bước tái dựng hình học chi tiết với OpenMVS. Việc chuyển đổi chính xác đảm bảo toàn bộ thông tin về camera, đám mây điểm và ảnh đầu vào được giữ nguyên và phù hợp với định dạng yêu cầu của OpenMVS. Nếu bước này bị bỏ qua hoặc xảy ra sai sót, toàn bộ pipeline sẽ bị gián đoạn và không thể tiếp tục thực hiện các bước tái tạo nâng cao, dẫn đến kết quả cuối cùng không đầy đủ hoặc không chính xác.

3.6.2 Tái tạo đám mây điểm dày đặc

Sau khi chuyển đổi dữ liệu từ OpenMVG, hệ thống tiến hành tái dựng đám mây điểm dày đặc bằng công cụ `DensifyPointCloud` của OpenMVS. Đây là bước quan trọng trong pipeline nhằm nâng cao độ chi tiết của mô hình 3D. Công cụ này sử dụng thuật toán Multi-View Stereo để ước lượng các điểm 3D mới dựa trên ảnh đầu vào và thông số camera đã biết. Nhờ đó, mật độ điểm được tăng từ đám mây thưa lên thành đám mây điểm dày, giúp mô hình trở nên chi tiết và chính xác hơn, tạo tiền đề cho bước tái dựng bề mặt tiếp theo.

Đầu ra của bước tái dựng đám mây điểm dày đặc là tệp `dense_scene.ply`, chứa mô hình đám mây điểm với mật độ cao được lưu dưới định dạng `.ply`. Mô hình này gồm hàng triệu điểm 3D, mỗi điểm có tọa độ không gian $[x, y, z]$ và màu sắc RGB tương ứng lấy từ ảnh gốc, giúp thể hiện chi tiết hình dạng và màu sắc thực của vật thể trong không gian ba chiều.

Vai trò của bước tái dựng đám mây điểm dày đặc là cốt lõi trong pipeline tái tạo 3D, khi nó biến dữ liệu hình học sơ khai từ SfM thành một tập hợp chi tiết và phong phú các điểm thể hiện bề mặt vật thể. Đầu ra của bước này chính là nền tảng quan trọng để xây dựng lưới 3D liên tục và thêm kết cấu trong các bước tiếp theo như `ReconstructMesh`, `RefineMesh` và `TextureMesh`, giúp mô hình cuối cùng vừa chính xác về hình dạng vừa chân thực về màu sắc.



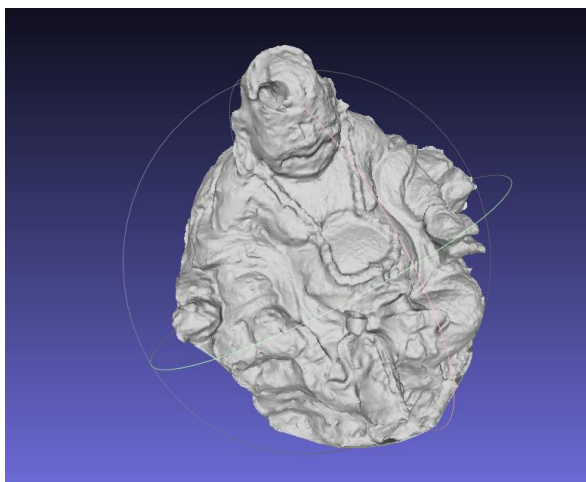
Hình 3.7 Đám mây điểm dày

3.6.3 Tái dựng lưới 3D

Sau khi thu được đám mây điểm dày đặc từ bước trước, hệ thống tiến hành xây dựng lưới tam giác từ tập hợp các điểm này, giúp chuyển đổi dữ liệu từ dạng điểm rời rạc sang một bề mặt liên tục có cấu trúc. Quá trình này hỗ trợ việc hiển thị và xử lý kết cấu dễ dàng hơn. Chức năng chính bao gồm xây dựng lưới tam giác 3D dựa trên đám mây điểm dày đặc, thường sử dụng các thuật toán như Poisson Surface Reconstruction để tạo ra bề mặt mượt mà, khép kín từ các điểm không đều. Hệ thống tự động xác định các mặt tam giác giữa các điểm gần nhau, tạo nên các bề mặt có thể trực quan hóa và xử lý tiếp trong các bước sau.

Đầu ra của bước xây dựng lưới tam giác là tệp mesh.ply, chứa mô hình lưới 3D bao gồm các thành phần chính là các đỉnh (vertices) với tọa độ 3D và các mặt (faces) là danh sách các tam giác kết nối các đỉnh đó. Tệp được lưu dưới định dạng .ply phổ biến, tương thích với nhiều phần mềm xử lý và hiển thị 3D như MeshLab.

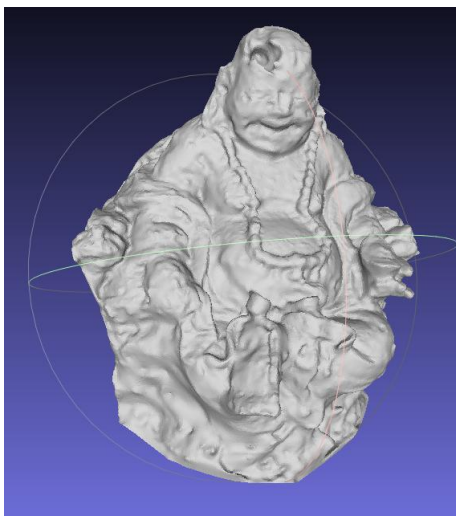
Vai trò của bước xây dựng lưới tam giác là chuyển đổi dữ liệu điểm rời rạc thành một cấu trúc hình học liên tục, có thể sử dụng cho các mục đích hiển thị trực quan, in 3D hoặc trích xuất thông tin hình học. Đây cũng là nền tảng quan trọng cho các bước tối ưu hóa lưới và dựng kết cấu tiếp theo, giúp nâng cao chất lượng hình ảnh và độ chân thực của mô hình 3D cuối cùng.



Hình 3.8 Lưới 3D

3.6.4 Tinh chỉnh lưới

Sau khi đã tạo lưới 3D từ đám mây điểm, bước tiếp theo là tối ưu và tinh chỉnh lưới nhằm nâng cao chất lượng mô hình. Quá trình này giúp loại bỏ nhiễu, sửa các lỗ hổng và điều chỉnh các chi tiết sai lệch trên bề mặt. Chức năng chính bao gồm làm mịn bề mặt để mô hình trở nên liền mạch và đẹp mắt hơn về mặt thị giác, đồng thời căn chỉnh lại lưới sao cho khớp chính xác với dữ liệu ảnh gốc, đảm bảo độ chính xác hình học cao hơn cho mô hình cuối cùng. Đầu ra của bước tối ưu và tinh chỉnh lưới là tệp mesh_refined.ply, chứa mô hình 3D đã được làm mượt và xử lý kỹ lưỡng hơn. Mô hình này giữ cấu trúc tương tự như mesh.ply nhưng loại bỏ các đỉnh dư thừa và nhiễu, các mặt tam giác được sắp xếp hợp lý hơn, giúp chuẩn bị tốt cho việc áp kết cấu (texture) ở các bước tiếp theo nhằm nâng cao độ chân thực của mô hình. Vai trò của bước tối ưu và tinh chỉnh lưới là nâng cao chất lượng mô hình 3D trước khi tiến hành dựng kết cấu. Quá trình này không chỉ giúp mô hình mượt mà và chính xác hơn về mặt hình học mà còn tăng tính thẩm mỹ, góp phần làm cho mô hình có giá trị ứng dụng cao hơn trong các lĩnh vực như hiển thị, in 3D hay thực tế ảo.

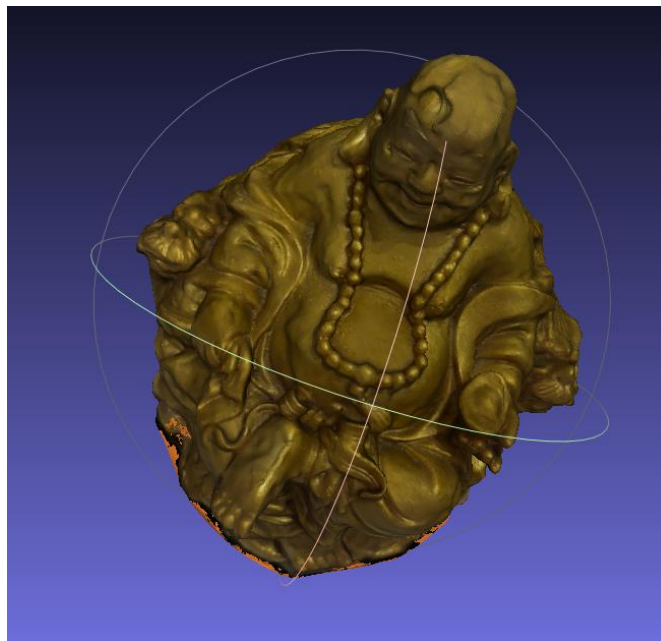


Hình 3.9 Lưới 3D đã tinh chỉnh

3.6.5 Áp kết cấu cho mô hình

Sau khi có lưới 3D được tinh chỉnh, bước cuối cùng là phủ kết cấu (texture) lên mô hình nhằm tạo màu sắc và chi tiết chân thực. Quá trình này bao gồm dựng bản đồ UV cho lưới 3D, chiếu các ảnh RGB gốc lên bề mặt mô hình sao cho chính xác với hình học, đồng thời áp dụng thuật toán lựa chọn ảnh phù hợp nhất cho từng vùng bề mặt, giúp mô hình hiển thị sống động và gần giống vật thể thật.

Đầu ra của bước phủ kết cấu bao gồm tệp mesh_textured.ply là mô hình lưới 3D hoàn chỉnh với màu sắc tự nhiên từ ảnh gốc, trong đó mỗi đỉnh và mặt đều được gán thông tin kết cấu. Bên cạnh đó còn có tệp texture (.png) chứa dữ liệu hình ảnh dùng để “sơn” lên mô hình, được tự động tạo và lưu kèm cùng tệp .ply, giúp mô hình 3D hiển thị thực và sinh động hơn.



Hình 3.10 Mô hình hoàn thiện



Hình 3.11 Ảnh kết cấu

3.6.6 Kết xuất mô hình 3D

OpenMVS tạo ra mô hình 3D dày đặc dưới dạng đám mây điểm hoặc lưới tam giác có kết cấu, và có thể xuất ra các định dạng phổ biến như PLY, OBJ và STL. Trong đó, PLY thường dùng để lưu đám mây điểm và lưới có màu; OBJ hỗ trợ kết cấu, vật liệu và được sử dụng rộng rãi; còn STL phù hợp cho in 3D nhưng không lưu màu sắc. Chất lượng mô hình phụ thuộc vào ảnh đầu vào, hiệu chuẩn camera và độ chính xác từ SfM. Tùy mục đích sử dụng, người dùng có thể chọn định dạng như PLY cho phân tích, OBJ cho trực quan hóa, hoặc STL cho in 3D.

Các định dạng này chứa thông tin hình học gồm các đỉnh và mặt, một số còn lưu thêm màu đỉnh, pháp tuyến, UV mapping và kết cấu. Sau khi xuất, mô hình có thể được xử lý, trực quan hóa hoặc chỉnh sửa bằng các phần mềm như MeshLab, CloudCompare, Blender hoặc phần mềm thương mại như Agisoft Metashape và RealityCapture.

Quy trình tái tạo mô hình 3D được thực hiện thông qua chuỗi các bước chính bao gồm, trước hết FFmpeg được dùng để trích xuất khung hình từ video đầu vào. Tiếp theo, rembg (sử dụng mô hình U-2-Net) loại bỏ nền nhằm giúp quá trình tái tạo tập trung vào đối tượng chính. Đặc trưng hình ảnh được trích xuất bằng SuperPoint, một thuật toán học sâu mạnh mẽ cho việc phát hiện và mô tả điểm đặc trưng. Sau đó, OpenMVG thực hiện ước lượng vị trí camera và tái tạo đám mây điểm thưa thớt thông qua kỹ thuật Structure-from-Motion. Cuối cùng, OpenMVS đảm nhận giai đoạn làm dày, tạo đám mây điểm dày đặc hoặc lưới tam giác có kết cấu, sẵn sàng cho các ứng dụng trực quan hóa hoặc phân tích.

Chất lượng mô hình 3D phụ thuộc chặt chẽ vào từng bước trong chuỗi xử lý, từ chất lượng khung hình, hiệu quả tách nền, độ chính xác của đặc trưng đến khả năng hiệu chỉnh và tái tạo hình học. Mặc dù còn tồn tại những thách thức như ảnh mờ, bề mặt phản xạ, hoặc thiếu góc nhìn, việc kết hợp các công cụ như FFmpeg, rembg, SuperPoint, OpenMVG và OpenMVS cho phép xây dựng một hệ thống tái tạo 3D hiệu quả, linh hoạt và có thể mở rộng theo nhu cầu nghiên cứu hoặc ứng dụng thực tế.

CHƯƠNG 4 XÂY DỰNG MÔ HÌNH VÀ ĐÁNH GIÁ KẾT QUẢ

Để có cái nhìn khách quan về hiệu quả của hệ thống, nghiên cứu đã tiến hành thực nghiệm trong nhiều điều kiện khác nhau. Cụ thể, mô hình 3D được so sánh khi chạy trên môi trường có GPU và không có GPU, đồng thời được đối chiếu với kết quả tạo ra từ phương pháp Gaussian Splatting. Tất cả các thực nghiệm đều được thực hiện trên cùng thiết bị, cùng môi trường và sử dụng chung một video đầu vào nhằm đảm bảo tính nhất quán trong so sánh.

4.1 THIẾT LẬP MÔI TRƯỜNG THỬ NGHIỆM ĐỀ XUẤT

Việc xây dựng một môi trường thử nghiệm được kiểm soát đóng vai trò then chốt trong bất kỳ nghiên cứu khoa học nghiêm túc nào, đặc biệt là trong lĩnh vực mô hình hóa 3D, nơi các yếu tố như phần mềm, phần cứng và cấu hình hệ thống có thể ảnh hưởng đáng kể đến kết quả đầu ra. Để đảm bảo tính ổn định, nhất quán và khả năng tái lập, các thực nghiệm được triển khai trên môi trường máy ảo. Máy ảo cung cấp khả năng cô lập hệ thống khỏi các yếu tố bên ngoài, đồng thời cho phép tái tạo lại một cấu hình phần cứng, phần mềm đồng nhất giữa các lần chạy thử nghiệm.

Việc sử dụng máy ảo cho phép thiết lập chính xác các thông số quan trọng như số lõi CPU, dung lượng RAM và khả năng truy cập GPU đây là những yếu tố có ảnh hưởng trực tiếp đến hiệu năng của các thuật toán xử lý và tái tạo mô hình 3D. Ngoài ra, việc lựa chọn phần mềm và công cụ mô hình hóa 3D cũng cần được cân nhắc kỹ lưỡng, dựa trên mục tiêu nghiên cứu, độ phức tạp của mô hình cần tái tạo và yêu cầu về chất lượng đầu ra.

Việc chuẩn hóa môi trường thực nghiệm thông qua máy ảo không chỉ đảm bảo tính nhất quán và tái lập của quá trình nghiên cứu mà còn giúp kiểm soát chặt chẽ các biến số có thể ảnh hưởng đến kết quả. Cấu hình hệ thống, phần mềm sử dụng và đặc điểm dữ liệu đầu vào cần được ghi nhận một cách chi tiết nhằm đảm bảo tính minh bạch, chính xác và hợp lệ của quá trình thực nghiệm.

4.2 SO SÁNH KẾT QUẢ TRONG ĐIỀU KIỆN CÓ VÀ KHÔNG CÓ GPU

Trong quá trình xây dựng mô hình 3D tự động, việc sử dụng GPU hay chỉ dựa vào CPU có ảnh hưởng đáng kể đến hiệu suất và chất lượng đầu ra. GPU được thiết kế chuyên biệt cho các tác vụ xử lý dữ liệu song song, đặc biệt là trong các lĩnh vực như thị giác máy tính, học sâu và tái tạo hình học. Do đó, hệ thống có GPU thường vượt trội rõ rệt so với hệ thống không GPU ở nhiều khía cạnh trong việc tạo mô hình 3D.

Một trong những bước quan trọng trong pipeline là trích xuất đặc trưng ảnh, nơi các thuật toán hiện đại như SuperPoint được sử dụng. Đây là một mô hình học sâu yêu cầu khả năng tính toán cao và gần như không khả thi để chạy hiệu quả trên CPU. Khi có GPU, SuperPoint có thể xử lý hàng trăm ảnh nhanh chóng, đảm bảo trích xuất đặc trưng chính xác, từ đó cải thiện đáng kể độ chi tiết và độ ổn định trong quá trình dựng mô hình. Ngược lại, nếu chạy SuperPoint trên CPU, tốc độ chậm hơn nhiều, làm giới

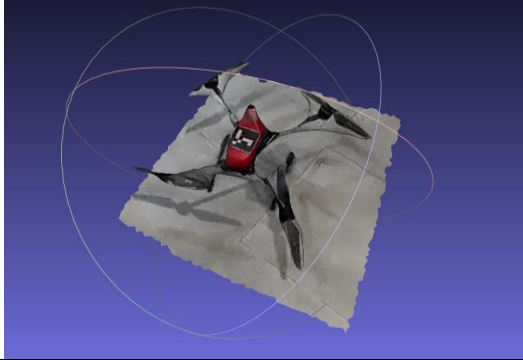
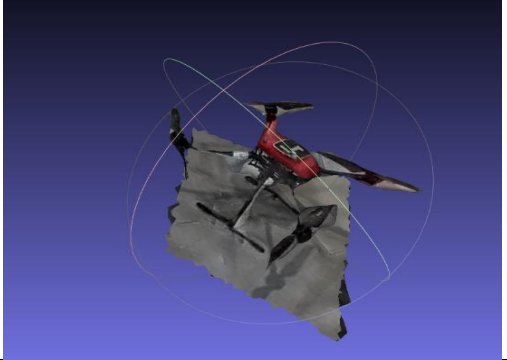
hạn số ảnh được xử lý hoặc kéo dài thời gian xử lý toàn hệ thống, từ đó ảnh hưởng đến kết quả đầu ra.

Về hiệu suất xử lý, GPU giúp tăng tốc đáng kể các bước như tái cấu trúc chuyển động (Structure from Motion), tái tạo đám mây điểm dày và dựng lưới 3D. Trong cùng điều kiện dữ liệu, hệ thống không có GPU có thể mất nhiều lần thời gian hơn để hoàn thành quá trình xử lý.

Về chất lượng mô hình, khi sử dụng GPU, hệ thống có khả năng xử lý nhiều ảnh hơn với độ chính xác cao, từ đó tạo ra mô hình chi tiết, bề mặt mịn và ít lỗi hơn so với khi không có GPU. Xét về độ phức tạp hình học, GPU cho phép dựng các mô hình với mật độ điểm cao và số lượng đa giác lớn hơn, phản ánh đầy đủ hơn hình dạng vật thể thực tế. Về tính ổn định của hệ thống, trong điều kiện không có GPU, hệ thống dễ gặp lỗi do quá tải CPU hoặc thiếu hụt RAM. Ngược lại, GPU giúp phân phối tải tính toán hiệu quả hơn, đảm bảo quy trình diễn ra mượt mà và ít gián đoạn. Cuối cùng, xét về chi phí và hiệu quả, mặc dù GPU đòi hỏi đầu tư ban đầu cao hơn, nhưng lại giúp tiết kiệm đáng kể thời gian và nguồn lực xử lý trong dài hạn, đặc biệt là trong các dự án có khối lượng lớn hoặc yêu cầu cao về chất lượng mô hình.

Để đánh giá sự khác biệt giữa hai điều kiện này, cần sử dụng cả số liệu định lượng và hình ảnh minh họa trực quan. Các chỉ số như thời gian xử lý, số lượng điểm hoặc đa giác, mật độ lưới, độ chính xác hình là những tiêu chí cụ thể giúp định lượng mức độ hiệu quả của từng hệ thống. Đồng thời, các ảnh so sánh từ cùng một góc nhìn cần được đưa ra để minh họa rõ ràng sự khác biệt về chi tiết, độ sắc nét và tính toàn vẹn của mô hình. Đồng thời, để đảm bảo tính khách quan trong quá trình so sánh, tất cả các thử nghiệm đều sử dụng chung một video đầu vào có độ phân giải thấp (600×338). Các thực nghiệm được thực hiện trên cùng một máy ảo với cấu hình phần cứng cố định gồm GPU NVIDIA GeForce RTX 3090 PCIe 24GB, hệ điều hành Ubuntu 20.04, bộ xử lý 16 nhân CPU, 32 GB RAM và ổ cứng SSD dung lượng 100 GB.

Việc so sánh kết quả được dựa trên Vertices và Faces, Vertices (các đỉnh) là tập hợp các điểm trong không gian ba chiều dùng để mô tả hình dạng hình học của vật thể. Mỗi đỉnh chứa thông tin tọa độ không gian (x, y, z) và có thể bao gồm thêm các thuộc tính như màu sắc, vector pháp tuyến hoặc các thông tin bổ sung khác tùy theo mục đích sử dụng. Faces (các mặt) là tập hợp các đa giác, thường là tam giác hoặc tứ giác, được tạo thành bằng cách kết nối các đỉnh lại với nhau. Các mặt này xác định cấu trúc bề mặt của mô hình 3D và là thành phần quan trọng để hiển thị vật thể dưới dạng lưới trong không gian ba chiều.

	Có GPU	Không có GPU
Mô hình		
Vartices	14436	12035
Faces	28868	23914

Bảng 4.1 Kết quả đánh giá có GPU và không có GPU

Kết quả cho thấy, việc tận dụng GPU giúp cải thiện đáng kể về số lượng điểm tái thiết và mức độ chi tiết của mô hình 3D đầu ra. Tuy nhiên, do hạn chế về chất lượng của video đầu vào, hiệu quả cải thiện khi sử dụng GPU chưa đạt mức tối ưu. Cụ thể, số lượng mặt tái tạo chỉ đạt khoảng 15–20% so với tiềm năng tối đa nếu video đầu vào có độ phân giải cao hơn. Điều này cho thấy chất lượng dữ liệu đầu vào vẫn là yếu tố quyết định hàng đầu đối với hiệu quả của toàn bộ pipeline xử lý, dù phần cứng có được tăng cường.

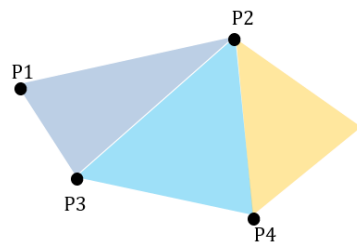
Hiện tại, vẫn chưa có tài liệu chính thức nào đưa ra đánh giá định lượng cụ thể về hiệu suất và chất lượng giữa các hệ thống có và không có GPU trong quy trình tái tạo mô hình 3D từ video được quay bằng điện thoại di động thông thường. Do đó, kết quả này mang tính thực nghiệm và gợi mở cho các nghiên cứu so sánh sâu hơn trong tương lai.

4.3 KẾT QUẢ HỆ THỐNG ĐỀ XUẤT SO VỚI GAUSSIAN SPLATTING

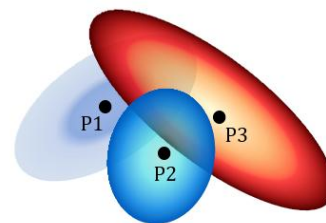
Gaussian Splatting là một phương pháp kết xuất mới trong lĩnh vực đồ họa máy tính và thị giác máy tính, được giới thiệu vào năm 2023. Nó thuộc nhóm kỹ thuật rendering từ ảnh, có khả năng tái hiện một cảnh 3D từ các ảnh chụp đa góc nhìn một cách cực kỳ nhanh chóng và chân thực, phục vụ tốt cho các ứng dụng thời gian thực như AR/VR, game hoặc mô phỏng ảo (graphdeco-inria, 2024).

Gaussian Splatting là một kỹ thuật kết xuất 3D hiện đại không sử dụng mô hình lưới tam giác truyền thống mà dựa trên tập hợp các điểm Gaussian trong không gian ba chiều. Mỗi điểm Gaussian là một ellipsoid mang thông tin về vị trí, màu sắc, hướng, kích thước, độ mờ và mật độ riêng, đóng vai trò như phần tử cơ bản để biểu diễn và hiển thị mô hình 3D. Quy trình xử lý bao gồm ba giai đoạn chính: trích xuất đặc trưng từ nhiều ảnh đầu vào chụp từ các góc nhìn khác nhau; sinh ra các điểm Gaussian với đầy đủ thông tin hình học và trực quan; và kết xuất thời gian thực bằng kỹ thuật

rasterization, tận dụng hiệu suất cao của GPU để tạo ảnh 2D từ góc nhìn người dùng. Gaussian Splatting tích hợp nhiều công nghệ hiện đại như trí tuệ nhân tạo và học sâu để tái tạo cảnh và tối ưu tham số điểm, sử dụng CUDA cùng GPU rendering để tăng tốc xử lý, ứng dụng Structure-from-Motion để tính toán vị trí camera và cấu trúc không gian, cùng với tối ưu hóa dựa trên gradient để tinh chỉnh thuộc tính điểm nhằm nâng cao độ chính xác của cảnh tái tạo. Kỹ thuật này nổi bật nhờ khả năng hiển thị cảnh 3D mượt mà và chân thực theo thời gian thực, tuy nhiên không tạo ra mô hình lưới nên không phù hợp cho các ứng dụng kỹ thuật như đo đạc hay in 3D. Khi so sánh với hệ thống truyền thống sử dụng các công cụ như SuperPoint, OpenMVG và OpenMVS, Gaussian Splatting có ưu thế về tốc độ xử lý do không cần tái tạo lưới hình học, nhưng yêu cầu phần cứng GPU mạnh để đạt hiệu suất tối ưu. Trong khi đó, hệ thống truyền thống tạo ra mô hình lưới hoàn chỉnh có thể chỉnh sửa và đo đạc, hoạt động tốt hơn trên cấu hình phổ thông và phù hợp hơn với môi trường giáo dục hoặc ứng dụng kỹ thuật. Do đó, việc lựa chọn phương pháp phụ thuộc vào mục tiêu sử dụng: Gaussian Splatting phù hợp cho trình diễn trực quan thời gian thực, trong khi hệ thống tái tạo lưới thích hợp cho các ứng dụng kỹ thuật và nghiên cứu.



Hình 4.1 Minh họa mesh hệ thống đề xuất



Hình 4.2 Minh họa điểm Gaussian 3D



Hình 4.3 Mô hình từ hệ thống đề xuất



Hình 4.4 Mô hình từ Gaussian Splatting

Mặc dù kết quả từ Gaussian Splatting cho thấy mô hình có chất lượng hình ảnh đẹp hơn và mượt mà hơn, nhưng thực chất đây chỉ là một mô hình đám mây điểm Gaussian. Điều này có nghĩa là mô hình không phải là một mô hình 3D hoàn chỉnh, mà chỉ là tập hợp các điểm ảnh được phân tán trong không gian 3D, chưa thể coi là một mô hình vật thể 3D thực sự có thể thao tác, phân tích hay in 3D.

Bảng so sánh

Tiêu chí	Hệ thống nghiên cứu	Gaussian Splatting
Thiết bị	<ul style="list-style-type: none"> - GPU: Có thể dùng GPU tầm trung (RTX 2060) hoặc không có - CPU: Quan trọng vì sẽ xử lý chính nếu không có GPU - Ram: $\geq 16\text{GB}$ 	<ul style="list-style-type: none"> -GPU: Bắt buộc, mạnh (RTX 3090+) VRAM $\geq 24\text{GB}$ - CPU: Quan trọng phụ, dùng để hỗ trợ - Ram: $\geq 32\text{GB}$
Tốc độ xử lý	Trung bình	Trung bình
Kết quả	Có thể tạo mô hình 3d mesh faces (lưới)	Đám mây điểm Gaussian (không phải lưới)
Ứng dụng	Trong hầu hết các ứng có thể dùng đối với mô hình 3d	Kết xuất thời gian thực, AR/VR
Ưu điểm	Dễ tiếp cận và sử dụng trong nhiều tình huống	Kết quả có độ chân thực cao
Nhược điểm	Do hệ thống đã được phát triển khá lâu nên nhiều công nghệ mới khó tích hợp và cập nhật	Không tạo mô hình có thể phân tích hoặc in 3D
Khả năng mở rộng	Dễ triển khai hơn	Hạn chế trong môi trường yếu

Bảng 4.2 So sánh tổng quát Hệ thống nghiên cứu và Gaussian Splatting

Gaussian Splatting là lựa chọn ưu việt cho các ứng dụng yêu cầu tốc độ và độ chân thực trong kết xuất, trong khi hệ thống phát triển có lợi thế về khả năng tái tạo và thao tác với mô hình 3D truyền thống. Mỗi phương pháp có ưu điểm riêng và phù hợp với các mục tiêu sử dụng khác nhau.

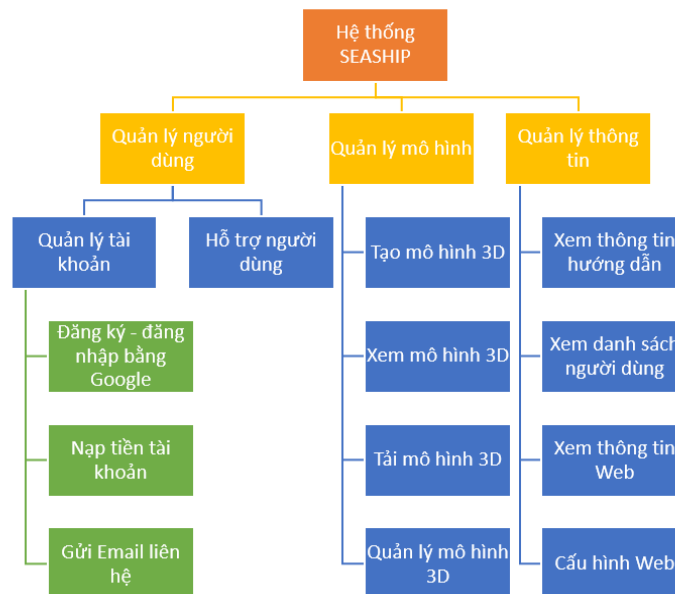
Việc so sánh kết quả đạt được với sự khác nhau của cấu hình thiết bị và với các thuật toán khác nhau như Gaussian Splatting giúp làm rõ các ưu điểm và nhược điểm của hệ thống. Qua đó, ta có thể nhận diện các điểm mạnh như tốc độ xử lý nhanh, nhưng cũng dễ dàng thấy các hạn chế như yêu cầu phần cứng cao hoặc khả năng xử lý với dữ liệu phức tạp. So sánh này giúp xác định các hướng cải thiện cho hệ thống, chẳng hạn như tối ưu phần mềm, nâng cấp phần cứng hoặc cải tiến thêm các thuật toán AI, từ đó giúp hệ thống phát triển mạnh mẽ và hiệu quả hơn trong tương lai.

CHƯƠNG 5 SẢN PHẨM NGƯỜI DÙNG

5.1 CHỨC NĂNG SẢN PHẨM

Sơ đồ chức năng

Sản phẩm được xây dựng với ba nhóm chức năng chính: quản lý người dùng, quản lý mô hình, và quản lý thông tin. Trong đó, phân hệ quản lý người dùng cung cấp các chức năng như đăng ký và đăng nhập bằng tài khoản Google, nạp tiền vào hệ thống, gửi email liên hệ cũng như hỗ trợ người dùng trong quá trình sử dụng. Phân hệ quản lý mô hình cho phép thực hiện các thao tác như tạo mới, xem, tải về và quản lý các mô hình 3D đã tạo. Bên cạnh đó, phân hệ quản lý thông tin hỗ trợ người dùng tra cứu hướng dẫn sử dụng, xem danh sách người dùng, thông tin về trang web và cấu hình hệ thống. Tất cả các chức năng được bố trí một cách hợp lý, khoa học nhằm đảm bảo quá trình sử dụng diễn ra thuận tiện, hiệu quả và thân thiện với người dùng.

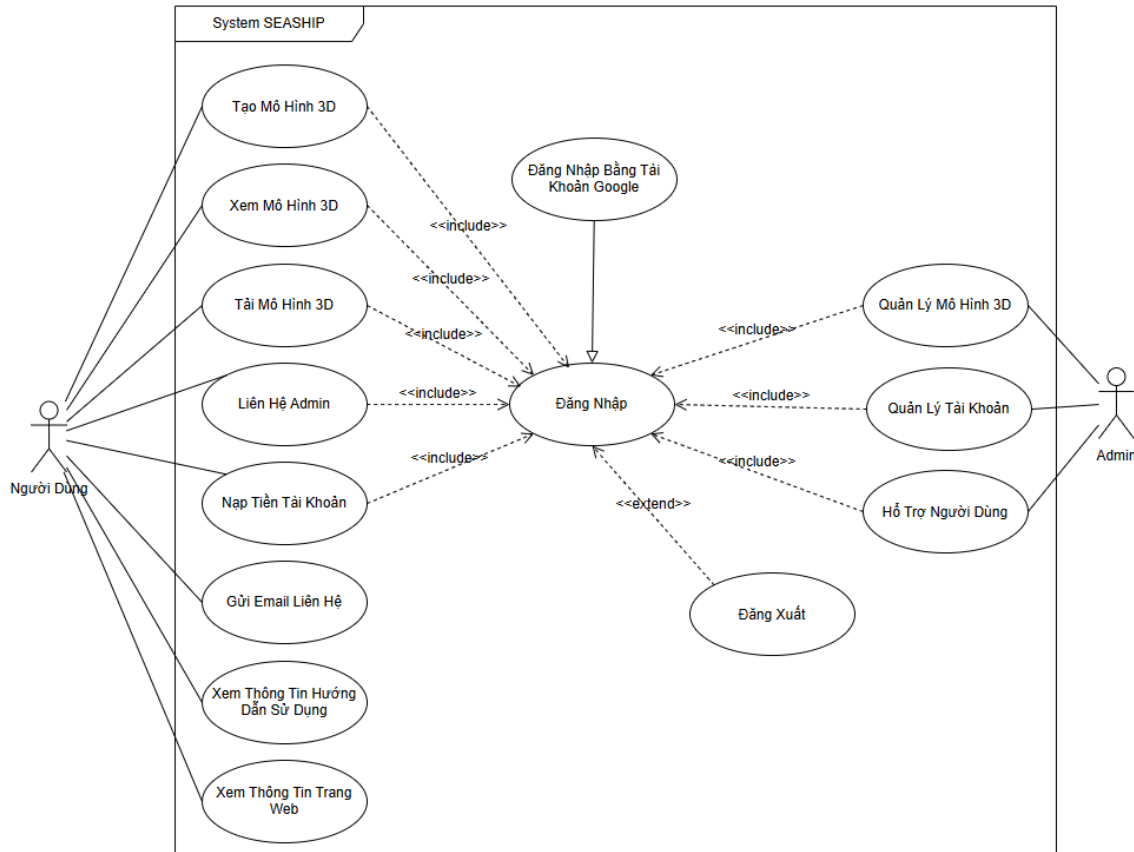


Hình 5.1 Sơ đồ chức năng

Sơ đồ Use Case

Hệ thống bao gồm đầy đủ các chức năng và hành vi tương tác cụ thể, được thiết kế để phục vụ cả người dùng thông thường và quản trị viên (admin). Sau khi đăng nhập, người dùng có thể thực hiện nhiều thao tác như tạo, xem, tải mô hình 3D, nạp tiền vào tài khoản, gửi email liên hệ, liên hệ với admin cũng như tra cứu thông tin hướng dẫn sử dụng và thông tin trang web. Việc đăng nhập được hỗ trợ thông qua tài khoản Google, kèm theo chức năng mở rộng là đăng xuất. Đối với admin, hệ thống cung cấp các công cụ quản lý mạnh mẽ bao gồm: quản lý mô hình 3D, quản lý tài khoản người dùng và hỗ trợ kỹ thuật cho người dùng. Các chức năng được tổ chức và liên kết logic thông qua các quan hệ “include” và “extend”, đảm bảo sự rõ ràng trong

quy trình vận hành và mang lại trải nghiệm sử dụng hiệu quả, linh hoạt cho cả người dùng cuối và quản trị viên hệ thống.



Hình 5.2 Sơ đồ Usecase

Đối với Người dùng

Đăng nhập bằng tài khoản Google, hệ thống cho phép người dùng đăng nhập thông qua tài khoản Google, giúp đơn giản hóa quá trình xác thực và nâng cao tính bảo mật. Việc xác thực được thực hiện dựa trên giao thức OAuth 2.0, đảm bảo thông tin cá nhân của người dùng được bảo vệ một cách an toàn trong suốt quá trình đăng nhập.

Tải lên video, người dùng có thể tải lên video từ thiết bị cá nhân với định dạng chuẩn MP4. Trước khi tiến hành xử lý, hệ thống sẽ kiểm tra tính hợp lệ của tệp video, bao gồm định dạng, dung lượng và khả năng đọc dữ liệu, nhằm đảm bảo chất lượng đầu vào cho quy trình tái dựng mô hình 3D.

Xử lý và phân tích video, sau khi video được tải lên, hệ thống sẽ tự động trích xuất các khung hình (frames) từ video đầu vào. Các khung hình này sẽ được đưa vào quy trình xử lý nhằm thực hiện tái dựng mô hình 3D, bao gồm các bước như phát hiện đặc trưng, khớp điểm, dựng cấu trúc không gian và tạo lưới 3D.

Quản lý và tùy chỉnh mô hình 3D, người dùng có thể xem trước mô hình 3D trực tiếp trên giao diện web một cách trực quan. Các tính năng cơ bản như xoay góc nhìn,

phóng to và thu nhỏ mô hình đều được hỗ trợ, giúp người dùng dễ dàng kiểm tra chi tiết mô hình trước khi quyết định tải xuống.

Tải xuống mô hình 3D, sau khi quá trình xử lý hoàn tất, hệ thống cho phép người dùng tải về mô hình 3D dưới định dạng chuẩn .ply. File mô hình này có thể mở và chỉnh sửa bằng các phần mềm phổ biến như MeshLab hoặc Blender, phục vụ cho các mục đích học tập, nghiên cứu hoặc sản xuất.

Các chức năng phụ khác, bên cạnh các chức năng chính, website cũng được xây dựng đầy đủ các tính năng hỗ trợ người dùng như gửi email liên hệ, nạp tiền tài khoản, và chatbox realtime để trao đổi trực tiếp với bộ phận hỗ trợ kỹ thuật. Ngoài ra, website có hệ thống các trang cơ bản như Trang chủ, Giới thiệu, Liên hệ và Quản trị, đảm bảo trải nghiệm người dùng chuyên nghiệp và toàn diện.

Đối với Quản trị viên

Đăng nhập tài khoản quản trị hệ thống, quản trị viên truy cập hệ thống bằng tài khoản Google đã được cấp quyền quản trị. Hệ thống sẽ kiểm tra quyền truy cập của tài khoản, đảm bảo chỉ những người có quyền mới được phép vào giao diện quản trị, từ đó đảm bảo tính bảo mật và kiểm soát hệ thống hiệu quả.

Quản lý tài khoản người dùng, trong giao diện quản trị, quản trị viên có thể xem và thống kê danh sách người dùng đã đăng nhập bằng tài khoản Google. Các thông tin cơ bản như địa chỉ email, tên người dùng và thời điểm đăng nhập được hiển thị để hỗ trợ việc quản lý người dùng một cách thuận tiện.

Giám sát hoạt động hệ thống, hệ thống cho phép theo dõi số lượng video được người dùng tải lên cũng như số lượng mô hình 3D được tạo thành công. Ngoài ra, quản trị viên có thể xem chi tiết danh sách các mô hình 3D bao gồm tên mô hình, người dùng tạo ra và thời điểm tạo, giúp dễ dàng giám sát tình trạng hoạt động và hiệu suất của hệ thống.

Quản lý cấu hình hệ thống, giao diện website có thể được quản lý linh hoạt thông qua khu vực cấu hình. Quản trị viên có thể chỉnh sửa nhanh chóng các thành phần như logo, tiêu đề trong phần Header, hoặc cập nhật thông tin bản quyền, liên hệ và liên kết nhanh trong Footer. Việc thay đổi các nội dung như chính sách, thông tin liên hệ cũng có thể thực hiện mà không cần truy cập mã nguồn, giúp tiết kiệm thời gian và tăng tính linh hoạt.

Hỗ trợ khách hàng, hệ thống tích hợp tính năng hỗ trợ khách hàng qua chatbox realtime, được triển khai bằng Socket.IO. Tính năng này cho phép người dùng trao đổi trực tiếp với bộ phận hỗ trợ kỹ thuật, giúp giải đáp thắc mắc và xử lý sự cố một cách nhanh chóng, nâng cao trải nghiệm người dùng trên website.

Yêu cầu phi chức năng

Bảo mật, hệ thống chỉ cho phép người dùng đăng nhập thông qua tài khoản Google, giúp giảm thiểu rủi ro bảo mật liên quan đến mật khẩu truyền thống. Bên cạnh đó, toàn bộ quá trình xác thực và truyền tải dữ liệu được đảm bảo an toàn bằng các

giao thức bảo mật hiện đại như HTTPS và OAuth 2.0, bảo vệ thông tin cá nhân và nội dung người dùng khỏi các mối đe dọa bên ngoài.

Khả năng mở rộng, Hệ thống được xây dựng với kiến trúc linh hoạt, sẵn sàng mở rộng để đáp ứng nhu cầu xử lý khối lượng lớn người dùng và dữ liệu trong tương lai. Việc triển khai các thành phần riêng biệt như xử lý video, dựng mô hình 3D và lưu trữ được tối ưu hóa để dễ dàng mở rộng quy mô theo chiều ngang hoặc dọc, đảm bảo hiệu suất hoạt động ổn định ngay cả khi lượng truy cập tăng cao.

Giao diện người dùng, giao diện website được thiết kế thân thiện, trực quan nhằm phục vụ mọi đối tượng người dùng, kể cả những người không có nền tảng kỹ thuật. Các thao tác chính như đăng nhập, tải lên video và tạo mô hình 3D được đơn giản hóa, giúp người dùng dễ dàng sử dụng hệ thống mà không cần hướng dẫn phức tạp, từ đó nâng cao trải nghiệm và hiệu quả sử dụng.

5.2 CÁC THÀNH PHẦN CỦA SẢN PHẨM

Hệ thống xây dựng tự động vật thể 3D từ video sử dụng trí tuệ nhân tạo (AI) chủ yếu bao gồm các thành phần để xử lý, phân tích dữ liệu và tái tạo mô hình 3D một cách trực quan, chính xác và tối ưu. Các thành phần này tương tác chặt chẽ nhằm đảm bảo chất lượng đầu ra và trải nghiệm người dùng.

Frontend (Giao diện người dùng - ReactJS)

Giao diện người dùng (frontend) được xây dựng bằng ReactJS – một thư viện JavaScript hiện đại, cho phép phát triển các thành phần UI một cách linh hoạt và hiệu quả. ReactJS đóng vai trò là cầu nối trực quan giữa người dùng và hệ thống, giúp tối ưu trải nghiệm tương tác thông qua khả năng cập nhật dữ liệu theo thời gian thực, tái sử dụng component, và quản lý trạng thái ứng dụng một cách rõ ràng, mạch lạc.

Trang chủ của hệ thống có nhiệm vụ hiển thị thông tin giới thiệu tổng quan về nền tảng, bao gồm mục tiêu, tính năng chính và các lợi ích mà hệ thống mang lại cho người dùng. Đây là điểm tiếp cận đầu tiên giúp người dùng hiểu rõ hơn về chức năng và giá trị của website.

Trang đăng nhập cho phép người dùng sử dụng tài khoản Google để đăng nhập vào hệ thống một cách nhanh chóng và an toàn, thông qua giao thức xác thực OAuth 2.0.

Trang hướng dẫn cung cấp nội dung chi tiết về cách sử dụng hệ thống, từ cách đăng nhập, upload video, đến các bước tạo và tải mô hình 3D. Giao diện được thiết kế trực quan nhằm giúp người dùng dễ tiếp cận, kể cả với những người chưa có kinh nghiệm kỹ thuật.

Trang liên hệ – gửi email hỗ trợ người dùng gửi các yêu cầu, phản hồi hoặc góp ý trực tiếp tới quản trị viên thông qua biểu mẫu email được tích hợp sẵn, giúp tăng cường kết nối và cải thiện chất lượng dịch vụ.

Trang tài khoản cho phép người dùng quản lý các mô hình 3D đã tạo trước đó. Ngoài ra, tại đây cũng cung cấp chức năng tải lên video mới để tiến hành quy trình dựng mô hình 3D, phục vụ nhu cầu số hóa vật thể một cách thuận tiện.

Trang xem trước mô hình cho phép hiển thị kết quả mô hình 3D đã dựng từ video người dùng upload. Người dùng có thể tương tác trực tiếp với mô hình trên giao diện web bằng cách xoay, thu phóng hoặc quan sát từ nhiều góc nhìn khác nhau.

Các trang quản trị (Admin) được thiết kế riêng cho người quản trị hệ thống với các chức năng như thống kê hoạt động hệ thống, quản lý tài khoản người dùng, mô hình 3D và cấu hình website. Đồng thời hỗ trợ tính năng chăm sóc khách hàng, xử lý phản hồi và quản lý nội dung một cách hiệu quả.

Về giao diện tổng thể, hệ thống được thiết kế theo phong cách hiện đại, thân thiện với người dùng. Màu sắc được chọn hài hòa, bố cục hợp lý, font chữ rõ ràng giúp người dùng dễ dàng thao tác và trải nghiệm mượt mà trên cả máy tính và thiết bị di động.

Backend (Xử lý hệ thống - NodeJS)

Hệ thống xử lý (backend) được phát triển bằng Node.js với framework Express, đóng vai trò là lớp trung gian xử lý và cung cấp dữ liệu giữa frontend và các thành phần xử lý mô hình 3D phía sau. Backend chịu trách nhiệm xử lý các chức năng chính của website, đảm bảo an toàn, đồng bộ và hiệu quả trong quá trình hoạt động.

Hệ thống hỗ trợ quản lý xác thực và người dùng thông qua cơ chế đăng nhập bằng tài khoản Google sử dụng giao thức OAuth2. Sau khi xác thực thành công, thông tin người dùng sẽ được lưu trữ trong cơ sở dữ liệu nhằm phục vụ cho việc quản lý mô hình cá nhân và theo dõi hoạt động. Hệ thống cũng hỗ trợ phân quyền người dùng và quản trị viên (admin) để đảm bảo kiểm soát truy cập phù hợp với vai trò.

Trong quản lý và tạo mô hình, hệ thống cho phép người dùng tải lên video cá nhân. Sau khi nhận được video, hệ thống sẽ tự động kích hoạt quy trình xử lý nhằm dựng mô hình 3D từ video đầu vào. Kết quả được xuất ra dưới dạng file .PLY và có thể xem trước trực tiếp trên giao diện web. Ngoài ra, người dùng có thể tải cả video và mô hình 3D về thiết bị cá nhân.

Chức năng gửi liên hệ và góp ý được tích hợp dưới dạng biểu mẫu email từ giao diện frontend. Các thông tin này sẽ được lưu trữ vào hệ thống hoặc chuyển tiếp đến quản trị viên thông qua thư viện Nodemailer, giúp đảm bảo việc tiếp nhận phản hồi người dùng một cách hiệu quả và có hệ thống.

Hệ thống cũng hỗ trợ tích hợp thanh toán – nạp tiền qua ZaloPay, cho phép người dùng nạp tiền vào tài khoản cá nhân để có thể sử dụng dịch vụ dựng mô hình 3D. ZaloPay API được sử dụng để tạo mã QR thanh toán và xác minh giao dịch. Tất cả các bước thanh toán đều được thực hiện qua kết nối bảo mật và đảm bảo tuân thủ các quy định về thanh toán điện tử hiện hành.

Trong phần hệ thống quản trị (Admin), quản trị viên có thể xem thống kê tổng quan như số lượng người dùng, số mô hình đã được tạo, và số lượng phản hồi góp ý. Họ cũng có thể quản lý danh sách tài khoản người dùng, thực hiện các thao tác như khóa, xóa tài khoản, đồng thời kiểm soát toàn bộ các mô hình được tạo ra và thực hiện các điều chỉnh cấu hình liên quan đến quyền hạn và nội dung website.

Cuối cùng, hệ thống hỗ trợ khách hàng được xây dựng bằng Socket.IO để triển khai boxchat realtime. Tính năng này cho phép người dùng gửi tin nhắn đến quản trị viên và nhận phản hồi ngay lập tức, giúp tăng cường trải nghiệm sử dụng và hỗ trợ người dùng một cách nhanh chóng, hiệu quả.

Database (Cơ sở dữ liệu - MySQL)

MySQL làm hệ quản trị cơ sở dữ liệu chính, với vai trò lưu trữ toàn bộ thông tin liên quan đến người dùng, video, mô hình 3D, lịch sử giao dịch và các cấu hình hệ thống. MySQL được lựa chọn nhờ tính ổn định cao, hiệu suất tốt, dễ mở rộng và cộng đồng hỗ trợ rộng lớn.

Bảng models3d lưu trữ thông tin các mô hình 3D do người dùng tạo. Các trường chính bao gồm id (khóa chính), link_video (đường dẫn video gốc), link_3d (đường dẫn tới mô hình .PLY), created_at, updated_at và users_email – là khóa ngoại tham chiếu đến trường email trong bảng users, nhằm xác định ai là người đã tạo mô hình.

Bảng users quản lý thông tin người dùng. Mỗi bản ghi bao gồm email (khóa chính), name (tên người dùng), money (số dư tài khoản), role (phân quyền, ví dụ: client hoặc admin), cùng với created_at và updated_at để theo dõi thời gian tạo và cập nhật tài khoản. Bảng này không sử dụng khóa ngoại.

Bảng payments lưu lại các giao dịch nạp tiền của người dùng. Trường id là khóa chính định danh giao dịch. Các trường khác gồm order_TotalPrice, createdAt, updatedAt, và userId – đây là khóa ngoại liên kết đến trường email trong bảng users, giúp xác định người thực hiện giao dịch.

Bảng messages phục vụ cho hệ thống chat thời gian thực. Mỗi tin nhắn có id (khóa chính), text (nội dung tin nhắn), created_at, updated_at, senderId (người gửi – khóa ngoại tham chiếu đến users.email) và conversationId (người nhận – cũng là khóa ngoại tham chiếu đến users.email).

Bảng settings lưu trữ các thiết lập cấu hình của hệ thống như logo, thông tin liên hệ, chính sách,... Bảng gồm các trường gồm id (khóa chính), setting_key, setting_value và description. Bảng này không sử dụng khóa phụ vì các cài đặt là dữ liệu tĩnh hoặc được quản trị viên điều chỉnh thủ công.

Thiết kế này đảm bảo tính liên kết chặt chẽ giữa các bảng, hỗ trợ quản lý người dùng, mô hình, thanh toán và trò chuyện một cách linh hoạt và có thể mở rộng trong tương lai.

5.3 THỰC THỂ

5.3.1 Thực thể models3d

Thực thể dữ liệu "models3d" bao gồm một bảng duy nhất có tên là "models3d" với các cột chứa thông tin chi tiết về các mô hình 3D. Cột "id" là khóa chính và được định dạng là INT. Các thông tin khác bao gồm email liên kết người dùng ("users_email" định dạng VARCHAR(255)), đường dẫn video ("link_video" định dạng VARCHAR(255)), đường dẫn mô hình 3D ("link_3d" định dạng VARCHAR(255)), ngày tạo ("created_at" định dạng DATETIME), và ngày cập nhật ("updated_at" định dạng DATETIME). Bảng này được lập chỉ mục (Indexes) để tối ưu hóa truy vấn.

Tên trường	Kiểu dữ liệu	Ghi chú
id	INT	Khóa chính
users_email	VARCHAR(255)	Khóa ngoại liên kết users.email
link_video	VARCHAR(255)	Đường dẫn video đã upload
link_3d	VARCHAR(255)	Đường dẫn mô hình 3D (PLY, v.v.)
created_at	DATETIME	Ngày tạo mô hình
updated_at	DATETIME	Ngày cập nhật mô hình

Bảng 5.1 Thực thể models3d

5.3.2 Thực thể users

Thực thể dữ liệu "users" bao gồm một bảng duy nhất có tên là "users" với các cột chứa thông tin chi tiết về người dùng. Cột "email" là khóa chính và được định dạng là VARCHAR(255). Các thông tin khác bao gồm tên người dùng ("name" định dạng VARCHAR(255)), số tiền ("money" định dạng FLOAT), vai trò ("role" định dạng VARCHAR(255)), ngày tạo ("created_at" định dạng DATETIME), và ngày cập nhật ("updated_at" định dạng DATETIME).

Tên trường	Kiểu dữ liệu	Ghi chú
email	VARCHAR(255)	Email người dùng (khóa chính/duy nhất)
name	VARCHAR(255)	Tên người dùng
money	FLOAT	Số dư hoặc chi phí liên quan
role	VARCHAR(255)	Vai trò (admin, user, v.v.)
created_at	DATETIME	Ngày tạo tài khoản
updated_at	DATETIME	Ngày cập nhật thông tin

Bảng 5.2 Thực thể users

5.3.3 Thực thể payments

Thực thể dữ liệu "payments" bao gồm một bảng duy nhất có tên là "payments" với các cột chứa thông tin chi tiết về các giao dịch thanh toán. Cột "id" là khóa chính và được định dạng là VARCHAR(255). Các thông tin khác bao gồm giá trị đơn hàng nạp vào tài khoản ("order_TotalPrice" định dạng FLOAT), mã người dùng ("userId" định dạng VARCHAR(255)), ngày tạo ("createdAt" định dạng DATETIME), và ngày cập nhật ("updatedAt" định dạng DATETIME).

Tên trường	Kiểu dữ liệu	Ghi chú
id	VARCHAR(255)	Khóa chính (ID thanh toán)
Oder_TotalPrice	FLOAT	Tổng số tiền
UserID	VARCHAR(255)	Khóa ngoại liên kết users.email
createdAt	DATETIME	Ngày tạo giao dịch
updated_at	DATETIME	Ngày cập nhật

Bảng 5.3 Thực thể payments

5.3.4 Thực thể messages

Thực thể dữ liệu "messages" bao gồm một bảng duy nhất có tên là "messages" với các cột chứa thông tin chi tiết về các tin nhắn. Cột "id" là khóa chính và được định dạng là INT. Các thông tin khác bao gồm mã người gửi ("senderId" định dạng VARCHAR(255)), mã người nhận ("conversationId" định dạng VARCHAR(255)), nội dung tin nhắn ("text" định dạng VARCHAR(255)), ngày tạo ("created_at" định dạng DATETIME), và ngày cập nhật ("updated_at" định dạng DATETIME). Bảng này được lập chỉ mục (Indexes) để tối ưu hóa truy vấn.

Tên trường	Kiểu dữ liệu	Ghi chú
id	INT	Khóa chính
conversationId	VARCHAR(255)	Mã hội thoại (có thể nhóm nhiều tin nhắn)
senderID	VARCHAR(255)	ID người gửi (liên kết đến users.email)
text	VARCHAR(255)	Nội dung tin nhắn
created_at	DATETIME	Ngày gửi
updated_at	DATETIME	Ngày chỉnh sửa

Bảng 5.4 Thực thể messages

5.3.5 Thực thể settings

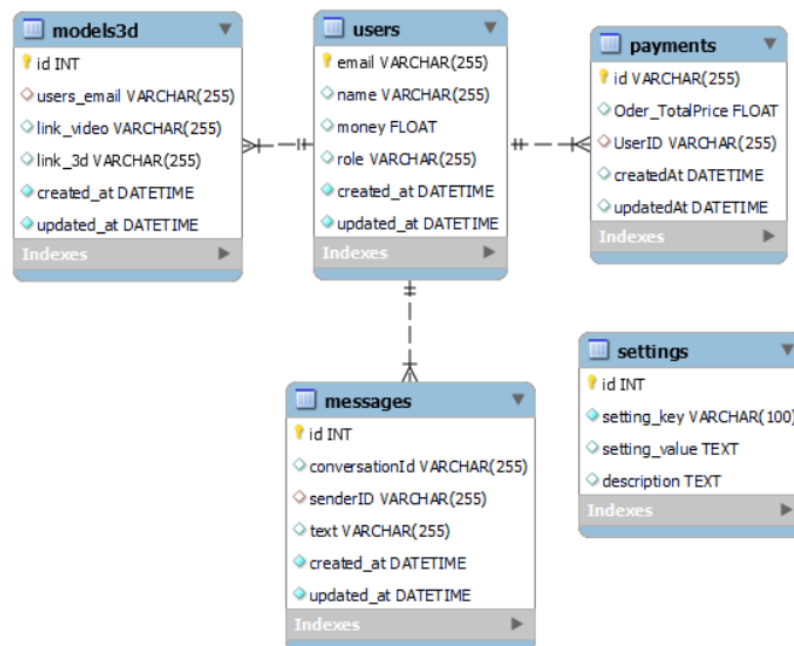
Thực thể dữ liệu "settings" bao gồm một bảng duy nhất có tên là "settings" với các cột chứa thông tin chi tiết về các thiết lập hệ thống. Cột "id" là khóa chính và được định dạng là INT. Các thông tin khác bao gồm khóa thiết lập ("setting_key" định dạng VARCHAR(100)), giá trị thiết lập ("setting_value" định dạng TEXT), và mô tả ("description" định dạng TEXT). Bảng này được lập chỉ mục (Indexes) để tối ưu hóa truy vấn.

Tên trường	Kiểu dữ liệu	Ghi chú
id	INT	Khóa chính
setting_key	VARCHAR(100)	Tên cấu hình
setting_value	TEXT	Giá trị cấu hình
description	TEXT	Mô tả ngắn về cấu hình

Bảng 5.5 Thực thể settings

5.4 SƠ ĐỒ ERD

Sơ đồ ERD gồm các bảng như users lưu thông tin người dùng như email, tên, vai trò và số dư, models3d lưu video và mô hình 3D do người dùng tạo, liên kết với bảng users, payments ghi lại lịch sử giao dịch thanh toán và messages quản lý các tin nhắn giữa người dùng trong hệ thống ngoài ra còn có settings chứa các thiết lập cấu hình hệ thống. Các bảng liên kết với nhau giúp quản lý dữ liệu chặt chẽ hơn.

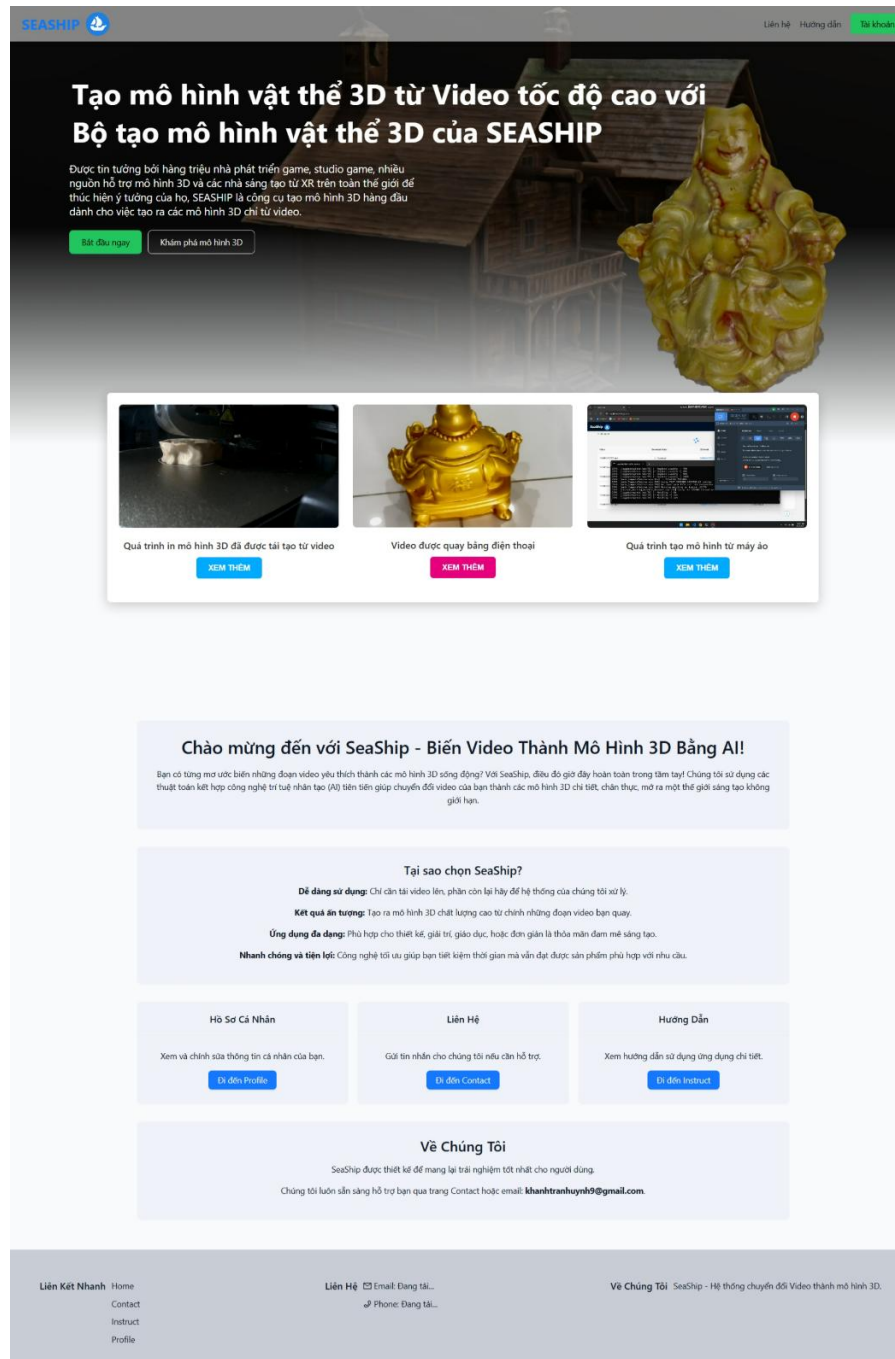


Hình 5.3 Sơ đồ ERD

5.5 GIAO DIỆN

Giao diện trang chủ

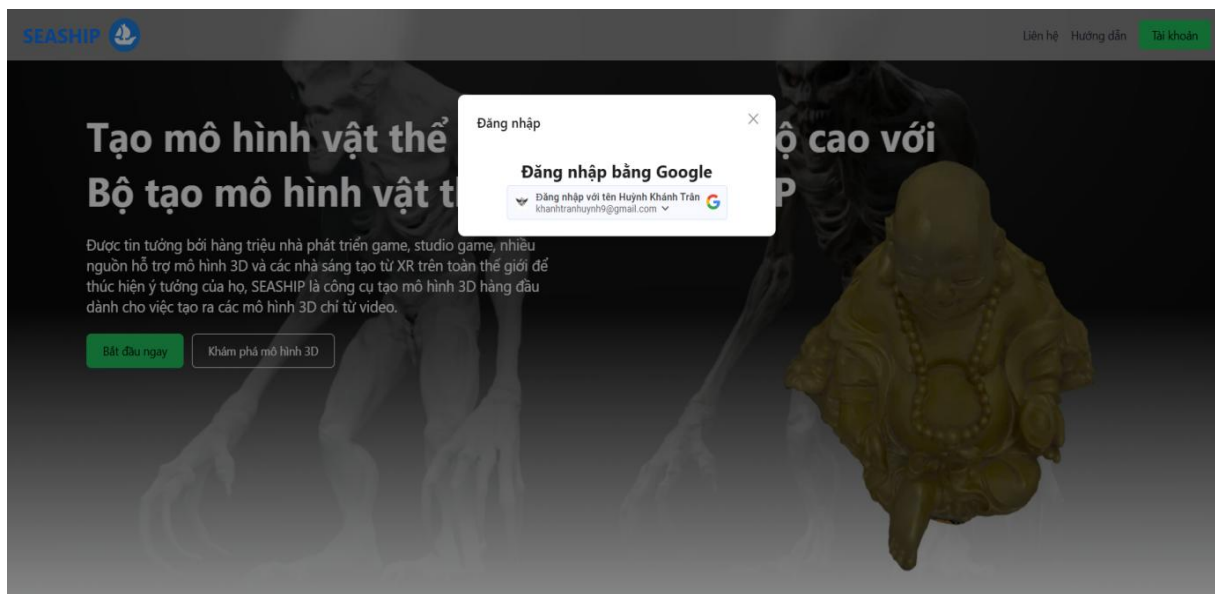
Giao diện trang chủ được thiết kế thân thiện, hiển thị thông tin và hình ảnh về hệ thống. Các mục như "Liên hệ", "Hướng dẫn" và "Tài khoản" được bố trí trực quan, cho phép người dùng dễ dàng thực hiện các thao tác đơn giản để truy cập thông tin và sử dụng các chức năng của hệ thống.



Hình 5.4 Giao diện trang chủ

Giao diện trang đăng nhập, đăng ký

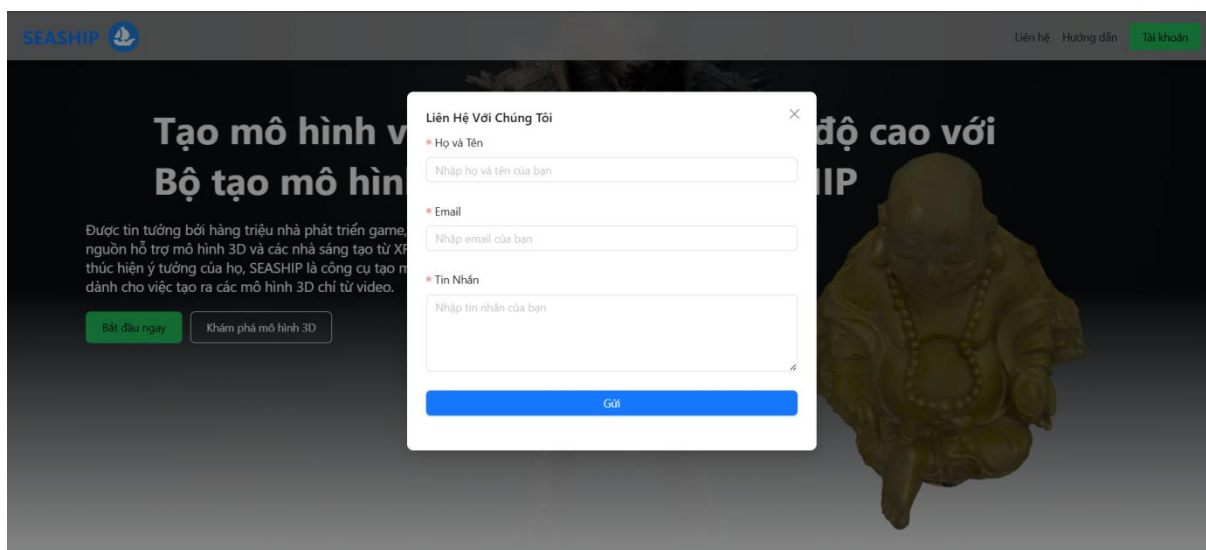
Giao diện đăng nhập và đăng ký được tối ưu hóa, hỗ trợ người dùng đăng nhập nhanh chóng thông qua tài khoản Google. Hệ thống tự động kiểm tra tính hợp lệ của thông tin đăng nhập và chuyển hướng người dùng về trang chủ nếu thông tin chính xác.



Hình 5.5 Giao diện trang đăng nhập, đăng ký

Giao diện trang liên hệ

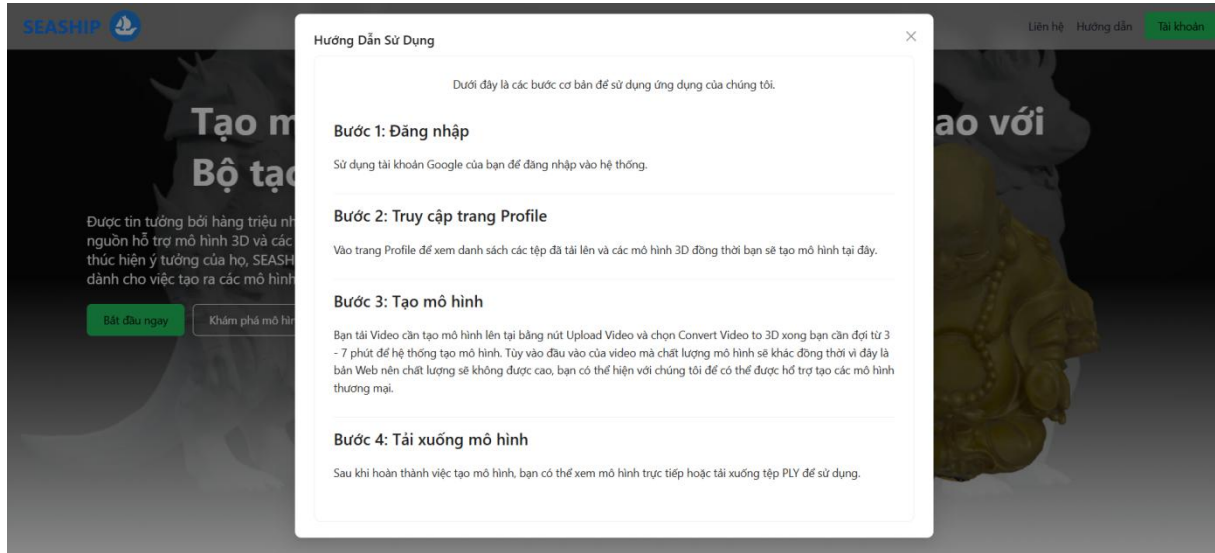
Trang liên hệ cho phép người dùng gửi yêu cầu hoặc phản hồi qua email, đảm bảo tính tiện lợi và dễ tiếp cận.



Hình 5.6 Giao diện trang liên hệ

Giao diện trang hướng dẫn

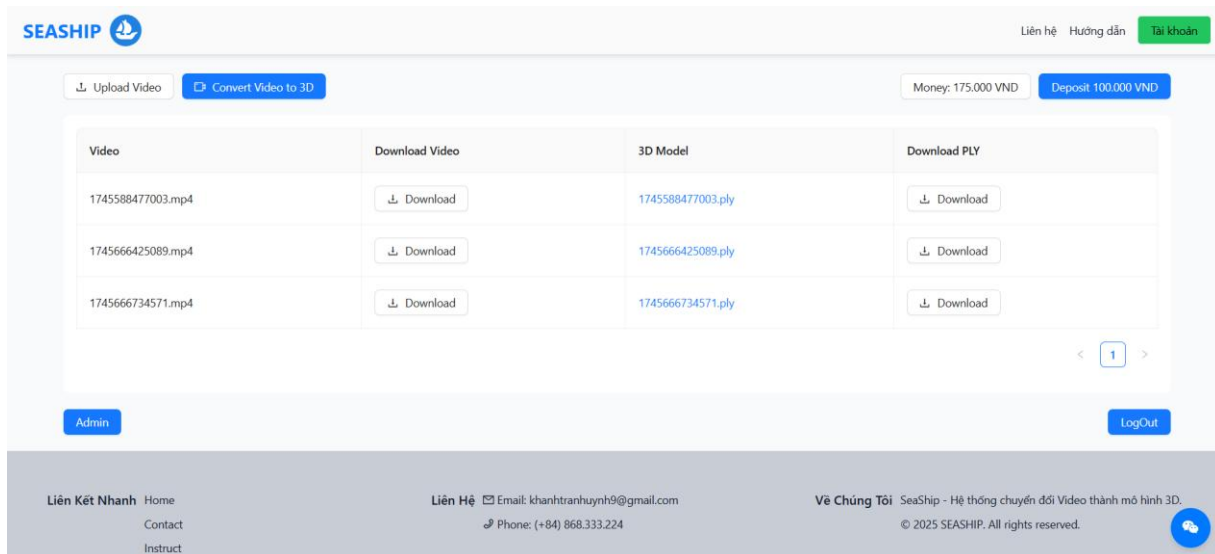
Trang hướng dẫn được thiết kế để hỗ trợ người dùng làm quen và sử dụng hệ thống một cách dễ dàng, với các hướng dẫn chi tiết và trực quan.



Hình 5.7 Giao diện trang hướng dẫn

Giao diện trang tài khoản người dùng

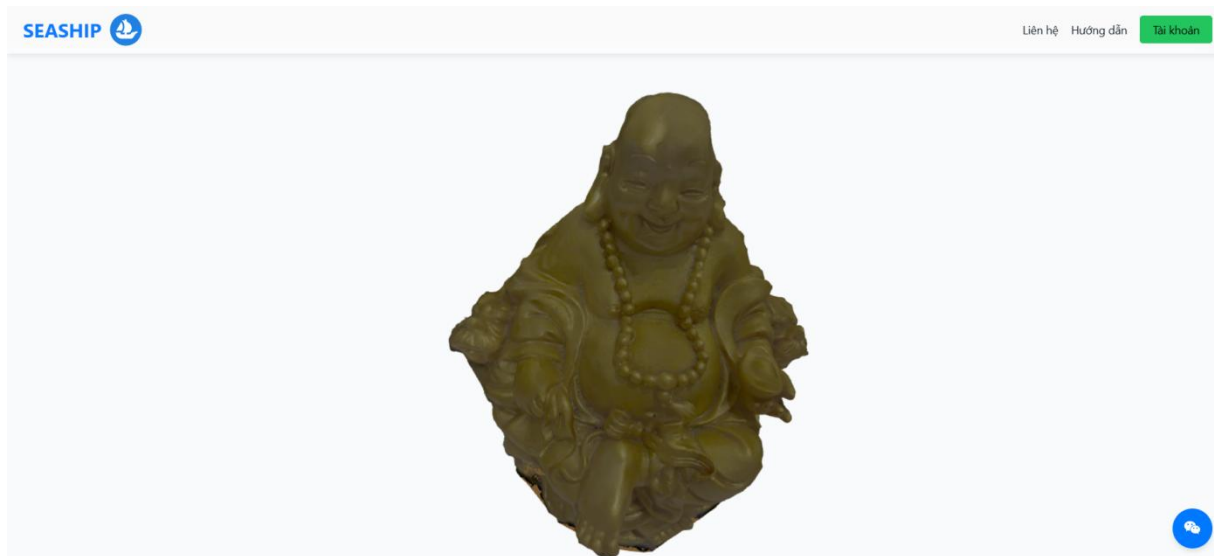
Trang tài khoản hiển thị thông tin về các mô hình 3D và video của người dùng. Tại đây, người dùng có thể chuyển đổi video thành mô hình 3D, xem trước và tải mô hình về thiết bị. Giao diện được thiết kế để đảm bảo trải nghiệm mượt mà và tiện lợi.



Hình 5.8 Giao diện trang tài khoản người dùng

Giao diện trang xem mô hình

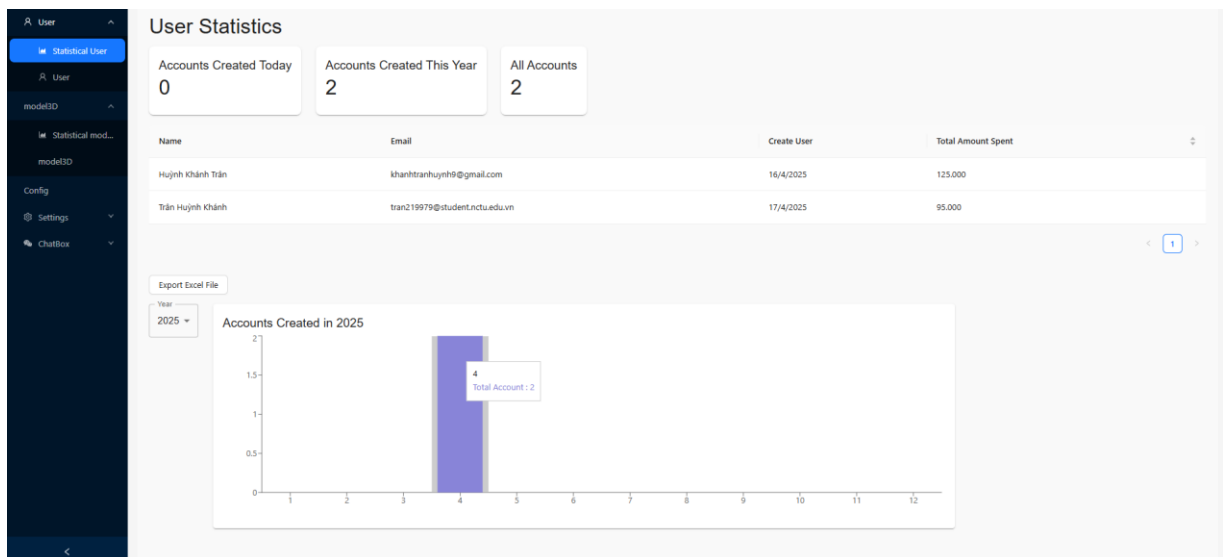
Trang xem mô hình cho phép người dùng xem trước và tương tác với các mô hình 3D, mang lại trải nghiệm trực quan và sinh động.



Hình 5.9 Giao diện trang xem mô hình

Giao diện các trang admin

Trang Admin này cho phép admin quản lý hết tất cả các vấn đề của Web từ thống kê danh sách quản lý User và mô hình 3D đến việc hỗ trợ người dùng và cấu hình Web.



Hình 5.10 Giao diện các trang admin

CHƯƠNG 6 TỔNG KẾT

6.1 KẾT LUẬN

Dựa trên quá trình thực hiện và kết quả đạt được, có thể khẳng định rằng nghiên cứu đã thành công trong việc hoàn thiện hệ thống tái tạo lưới 3D và tạo ra sản phẩm mô hình vật thể 3D thực tế từ video đầu vào. Hệ thống không chỉ đáp ứng được các mục tiêu đặt ra ban đầu như nâng cao độ chính xác hình học, tăng tính chân thực của mô hình và giảm thời gian xử lý, mà còn cho thấy khả năng xử lý hiệu quả nhiều loại dữ liệu khác nhau. Việc hoàn thiện sản phẩm minh chứng cho tính khả thi và hiệu quả của giải pháp giúp cho việc tái tạo mô hình vật thể 3D từ video một cách tự động, đồng thời mở ra tiềm năng ứng dụng rộng rãi trong các lĩnh vực như giáo dục, thiết kế, kỹ thuật và số hóa vật thể phục vụ đào tạo từ xa.

Về tính khả thi và ứng dụng, hệ thống mô hình hóa 3D do nghiên cứu xây dựng cho thấy tiềm năng ứng dụng cao nếu đáp ứng được các yếu tố then chốt như chi phí hợp lý, triển khai thuận tiện và không đòi hỏi tài nguyên tính toán quá lớn. Trên thực tế, công nghệ mô hình 3D đang được ứng dụng rộng rãi trong nhiều lĩnh vực như xây dựng, thiết kế công nghiệp, kiểm tra kỹ thuật, tiếp thị trực quan, đào tạo mô phỏng và in 3D. Việc ứng dụng mô hình 3D không chỉ nâng cao hiệu quả giao tiếp và phối hợp giữa các bên liên quan mà còn hỗ trợ ra quyết định chính xác trong thiết kế và rút ngắn chu trình phát triển sản phẩm. Hệ thống nghiên cứu đã chứng minh được khả năng tích hợp linh hoạt và ứng dụng thực tiễn hiệu quả, đây có thể xem là một đóng góp đáng kể cả về mặt học thuật lẫn tiềm năng triển khai trong môi trường công nghiệp.

6.2 HƯỚNG PHÁT TRIỂN TƯƠNG LAI

GPU đóng vai trò trọng yếu trong việc nâng cao hiệu suất và rút ngắn đáng kể thời gian xử lý trong các hệ thống tái tạo mô hình 3D hiện đại. Với kiến trúc xử lý song song cực mạnh, GPU có thể thực hiện hàng nghìn tác vụ đồng thời - một khả năng mà CPU truyền thống khó có thể sánh kịp. Điều này đặc biệt quan trọng khi áp dụng các kỹ thuật tính toán chuyên sâu như NeRF, Gaussian Splatting hoặc các mô hình học sâu trong AI, vốn yêu cầu khối lượng xử lý lớn. Bên cạnh việc tăng tốc xử lý, GPU còn hỗ trợ khả năng hiển thị thời gian thực và tích hợp các hiệu ứng đồ họa nâng cao, giúp nhà phát triển dễ dàng kiểm tra, điều chỉnh và tối ưu mô hình trong suốt quá trình triển khai. Việc tận dụng GPU không chỉ là lựa chọn tối ưu hóa hiệu suất mà còn là xu thế tất yếu trong kỷ nguyên tính toán hiện đại, góp phần quyết định đến thành công của toàn bộ quy trình tái tạo 3D.

Các phương pháp hiện đại như Neural Radiance Fields (NeRF) và Gaussian Splatting đang định hình lại cách thức tái tạo mô hình 3D với độ chân thực và chi tiết vượt trội. NeRF tận dụng sức mạnh của mạng nơ-ron để học biểu diễn cảnh không gian 3 chiều từ một tập hợp ảnh 2D, từ đó tạo ra các góc nhìn mới với độ liên mạch cao, ánh sáng tự nhiên và chiều sâu sống động. Trong khi đó, Gaussian Splatting tiếp cận theo hướng khác việc sử dụng các hạt Gaussian bán trong suốt để mô phỏng bề mặt, mang lại khả năng tái tạo tinh vi các chi tiết nhỏ, đồng thời đảm bảo tốc độ kết xuất cực

nhANH. Đặc biệt, Gaussian Splatting nổi bật ở khả năng cân bằng giữa chất lượng hình ảnh cao và hiệu suất xử lý thời gian thực, mở ra tiềm năng ứng dụng mạnh mẽ trong các lĩnh vực như thực tế ảo, đào tạo kỹ thuật và y khoa số.

Tính năng	NeRF	Gaussian Splatting
Chất lượng kết xuất	Chất lượng cao, có khả năng tạo ra các hiệu ứng phức tạp như phản xạ và khúc xạ	Chất lượng vượt trội, tạo ra các đám mây điểm hoàn chỉnh hơn với độ chi tiết đặc biệt
Tốc độ kết xuất	Có thể chậm hơn do kết xuất dò tia trên mỗi pixel	Nhanh hơn đáng kể do sử dụng kết xuất dựa trên raster hóa GPU
Xử lý hình học phức tạp	Hiệu quả trong việc mô hình hóa các cảnh phức tạp và tạo các khung nhìn mới	Rất chính xác trong việc ước tính các bề mặt và cấu trúc tinh tế
Khả năng tổng hợp khung nhìn	Xuất sắc trong việc tạo các khung nhìn mới từ các khung nhìn hiện có	Tập trung vào kết xuất thời gian thực bằng cách lưu trữ thông tin 3D dưới dạng điểm Gaussian
Thời gian huấn luyện	Có thể tốn thời gian	Thời gian huấn luyện nhanh do tối ưu hóa trực tiếp các thuộc tính Gaussian
Hạn chế tiềm năng	Có thể gặp khó khăn với độ nhất quán đa khung nhìn, dẫn đến ước tính độ sâu không chính xác và kết xuất mờ	Sử dụng VRAM cao và khả năng tương thích chưa hoàn toàn với các quy trình kết xuất hiện có

Bảng 6.1 So sánh NeRF và Gaussian Splatting

Mặc dù dữ liệu đầu vào là video, tuy nhiên việc ghi lại đầy đủ tất cả các góc nhìn của vật thể trong môi trường thực tế là điều không dễ dàng, đặc biệt với những bề mặt khuất hoặc góc quay bị hạn chế. Nghiên cứu này tập trung vào việc tái tạo mô hình 3D dựa trên phân hình ảnh quan sát được, mà chưa xử lý các vùng bị che khuất hoặc chưa được ghi nhận thuộc vào nhóm bài toán tạo sinh (Generative). Trong tương lai, để nâng cao chất lượng hình học và khắc phục những giới hạn trên, có thể tích hợp thêm các mô hình học sâu hiện đại như GRNet (Gridding Residual Network) và Generative nhằm thực hiện point cloud completion – quá trình hoàn thiện đám mây điểm từ dữ liệu thiếu hụt. Đây là các mô hình tạo sinh mạnh mẽ, có khả năng học và suy diễn cấu trúc hình

học ẩn dựa trên dữ liệu quan sát được. GRNet khai thác không gian 3 chiều thông qua biểu diễn voxel hóa, giúp mô hình hiểu rõ hơn về hình dạng tổng thể, trong khi PoinTr sử dụng kiến trúc Transformer, cho phép học ngữ cảnh toàn cục và biểu diễn hình học chính xác hơn. Việc ứng dụng các mô hình này hứa hẹn sẽ nâng cao đáng kể độ hoàn thiện và tính liên tục của dữ liệu 3D, tạo tiền đề vững chắc cho quá trình tái tạo lưới tam giác với độ chi tiết cao, phục vụ hiệu quả cho các ứng dụng thị giác máy tính và thực tế ảo tăng cường.

TÀI LIỆU THAM KHẢO

- [1] Van-Linh Nguyen và cộng sự (n.d). Automatic 3d Reconstruction And Rendering Based On Sparse Images. Nam Can Tho University.
- [2] Massimiliano Pepe và cộng sự (2022). UAV Platforms and the SfM-MVS Approach in the 3D Surveys and Modelling: A Review in the Cultural Heritage Field. MDPI. (2022, tháng 12).
- [3] Liang Ma và cộng sự (2023). Application of artificial intelligence in 3D printing physical organ models. Elsevier. (2023, tháng 12). Truy cập tại: <https://www.sciencedirect.com/science/article/pii/S2590006423002521>
- [4] Ronen Basri và cộng sự (2017). A Survey of Structure from Motion. Arxiv. (2017, 9 tháng 5). Truy cập tại: <https://arxiv.org/pdf/1701.08493>
- [5] Chuanzhi Xu và cộng sự (2025). A Survey of 3D Reconstruction with Event Cameras: From Event-based Geometry to Neural 3D Rendering. Arxiv. (2025, 13 tháng 5). Truy cập tại: <https://arxiv.org/html/2505.08438v1>
- [6] Liang Zhang và cộng sự (2021). Multi-view stereo in the Deep Learning Era: A comprehensive review. Elsevier. (2021, tháng 12). Truy cập tại: <https://www.sciencedirect.com/science/article/abs/pii/S0141938221001062>
- [7] Dening Lu và cộng sự (2023). NeRF: Neural Radiance Field in 3D Vision, Introduction and Review. Arxiv. (2023, 30 tháng 11). Truy cập tại: <https://arxiv.org/pdf/2210.00379>
- [8] Michael Rubloff (2025). Radiance Fields (Gaussian Splatting and NeRFs). Radiancefields. (2025, 21 tháng 5). Truy cập tại: <https://radiancefields.com/>
- [9] Mr.thanduc (2018). Lịch sử phát triển của Công nghệ in 3D. mrthanduc. (2018, 20 tháng 4). Truy cập tại: <https://mrthanduc.wordpress.com/2018/04/20/lich-su-phat-trien-cua-cong-nghe-in-3d/>
- [10] Hưng Nguyễn (2024). Deep Learning là gì? Tổng quan về Deep Learning từ A-Z. vietnix. (2024, 1 tháng 8). Truy cập tại: <https://vietnix.vn/deep-learning-la-gi/>
- [11] Thủy Nguyễn (2024). Convolutional neural network: Khái niệm. cấu trúc. cách xây dựng. bizfly. (2024, 26 tháng 10). Truy cập tại: <https://bizfly.vn/techblog/convolutional-neural-network.html>

- [12] Kunal Dawn (2024). Enhancing Image Segmentation using U2-Net: An Approach to Efficient Background Removal. Learnopencv. (2024, 11 tháng 6). Truy cập tại: <https://learnopencv.com/u2-net-image-segmentation/>
- [13] Fangjinhua Wang và cộng sự (2024). Learning-based Multi-View Stereo: A Survey. Arxiv. (2024, 9 tháng 12). Truy cập tại: <https://arxiv.org/abs/2408.15235>
- [14] Duy (2018). Tìm hiểu FFMPEG và cách thức sử dụng FFMPEG. Viblo. (2018, 21 tháng 12). Truy cập tại: <https://viblo.asia/p/tim-hieu-ffmpeg-va-cach-thuc-su-dung-ffmpeg-jvElaPOxZkw>
- [15] Vohungvi (2024). Tách background sản phẩm bằng thư viện rembg. Thigiacmaytinh. (2024, 11 tháng 1). Truy cập tại: <https://thigiacmaytinh.com/tach-background-san-pham-bang-thu-vien-rembg/>
- [16] Rpautrat (2025). SuperPoint. Github. rpautrat. (2025, 5 tháng 5). Truy cập tại: <https://github.com/rpautrat/SuperPoint>
- [17] openMVG (2025). OpenMVG (open Multiple View Geometry). Github. oepnMVG. (2025, 20 tháng 3). Truy cập tại: <https://github.com/openMVG/openMVG>
- [18] cdcseacave (2025). OpenMVS: open Multi-View Stereo reconstruction library. Github. Cdcseacave. (2025, 12 tháng 4). Truy cập tại: <https://github.com/cdcseacave/openMVS>
- [19] graphdeco-inria (2024). 3D Gaussian Splatting for Real-Time Radiance Field Rendering. Github. graphdeco-inria. (2024, 30 tháng 10). Truy cập tại: <https://github.com/graphdeco-inria/gaussian-splatting>