

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



## ĐỒ ÁN TỐT NGHIỆP

---

# PHÁT HIỆN VẬT THỂ 3D QUA ẢNH ĐƠN VỚI TIẾP CẬN HỌC SÂU

---

Ngành : KHOA HỌC MÁY TÍNH

Hội đồng: Khoa học máy tính 2

GVHD: TS. Nguyễn Đức Dũng

GVPB: TS. Nguyễn Tiên Thịnh

Sinh viên thực hiện:	Nguyễn Hữu Lợi	1914047
	Ngô Tân Phát Đạt	1913045

TP. HỒ CHÍ MINH, 5/2023



# DUYỆT LUẬN VĂN

ĐẠI HỌC BÁCH KHOA, ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

Văn bản này xác nhận Luận văn thực hiện bởi Nguyễn Hữu Lợi và Ngô Tấn Phát Đạt, với tên “**Phát hiện vật thể 3D qua ảnh đơn với tiếp cận học sâu**”, đã hoàn thành các yêu cầu về luận văn đại học của Khoa Khoa học và Kỹ thuật máy tính, tuân thủ các quy định của trường và các chuẩn mực về tính nguyên bản của luận văn.

## Ký bởi hội đồng

Tên	Chữ ký	Ngày
(Hướng dẫn)		
(Đồng hướng dẫn)		
(Chủ tịch hội đồng)		
(Uỷ viên hội đồng)		

### **Lời cam đoan**

Chúng em xin cam đoan đây là công trình nghiên cứu của riêng chúng em dưới sự hướng dẫn của thầy Nguyễn Đức Dũng. Nội dung nghiên cứu và các kết quả đều là trung thực và chưa từng được công bố trước đây. Các số liệu được sử dụng cho quá trình phân tích, nhận xét được chính chúng tôi thu thập từ nhiều nguồn khác nhau và sẽ được ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, chúng em cũng có sử dụng một số nhận xét, đánh giá và số liệu của các tác giả khác, cơ quan tổ chức khác. Tất cả đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kì sự gian lận nào, chúng em xin hoàn toàn chịu trách nhiệm về nội dung đồ án tốt nghiệp của mình. Trường Đại học Bách Khoa - Đại học Quốc gia TP. Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện.

### **Lời ngỏ**

Chúng em xin gửi lời cảm ơn sâu sắc đến thầy Nguyễn Đức Dũng vì sự kiên nhẫn, thấu hiểu và tận tình, cùng với đó là những góp ý và kinh nghiệm trân quý của thầy trong suốt quá trình hướng dẫn thực hiện nghiên cứu này. Chúng em muốn bày tỏ lòng biết ơn đối với sự giúp đỡ về mặt tài nguyên của FABLAB, đây cũng là một đóng góp quan trọng giúp cho việc thử nghiệm giải pháp đề xuất trở nên khả thi. Cuối cùng, chúng em xin cảm ơn đến gia đình và bạn bè đã luôn đồng hành và hỗ trợ trên con đường này.

## **Tóm tắt nội dung**

Phát hiện đối tượng 3D là một vấn đề nghiên cứu quan trọng trong học sâu. Trong nhiều lĩnh vực, chẳng hạn như lái xe tự hành và robot cầm nắm, hệ thống cần phải thu được thông tin 3D của các đối tượng thông qua cảm biến để từ đó có thể hoạt động, quyết định những hành động tiếp theo. Các phương pháp tiếp cận trong việc phát hiện đối tượng 3D hiện tại có thể được chia thành phương pháp dựa trên tín hiệu của cảm biến LiDAR, dựa trên hình ảnh. Các thuật toán dựa trên LiDAR mang tính chính xác và hiệu quả, nhưng chi phí cao hạn chế việc sử dụng nó trong công nghiệp. Các thuật toán dựa trên hình ảnh được sử dụng rộng rãi hơn vì chi phí thấp và thông tin kết cấu phong phú. Tuy nhiên, cách tiếp cận dựa trên hình ảnh đối mặt với thử thách lớn hơn so với cách tiếp cận dựa trên LiDAR vì phục hồi thông tin 3D từ dữ liệu đầu vào hai chiều vẫn là một vấn đề chưa thể giải quyết, đặc biệt khó khăn hơn khi sử dụng hình ảnh đơn làm dữ liệu đầu vào. Trong báo cáo này, nhóm đã thực hiện khảo sát các nghiên cứu đã có và nêu ra các hướng phát triển cũng như các công trình liên quan trong bài toán phát hiện vật thể 3D trong hình ảnh đơn. Dựa vào đó, nhóm đề xuất giải pháp nhằm nâng cao hiệu suất của mô hình cơ sở và thực hiện thử nghiệm cũng như đánh giá về giải pháp.

# Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>1</b>
1.1	Thực trạng . . . . .	1
1.2	Mục tiêu nghiên cứu . . . . .	1
1.3	Nhiệm vụ nghiên cứu . . . . .	1
1.4	Tổng quan về bố cục . . . . .	2
<b>2</b>	<b>Kiến thức nền tảng</b>	<b>3</b>
2.1	Định nghĩa bài toán . . . . .	3
2.2	Các hệ tọa độ . . . . .	3
2.2.1	Hệ tọa độ thế giới thực (World coordinate system - 3D) . . . . .	4
2.2.2	Hệ tọa độ máy ảnh (Camera coordinate system - 3D) . . . . .	4
2.2.3	Hệ tọa độ hình ảnh (Image coordinate system - 2D) . . . . .	4
2.2.4	Hệ tọa độ điểm ảnh (Pixel coordinate system - 2D) . . . . .	4
2.3	Các cảm biến đầu vào . . . . .	4
2.3.1	LiDAR . . . . .	6
2.3.2	Máy ảnh . . . . .	6
2.3.3	Nhược điểm của máy ảnh và LiDAR . . . . .	6
2.4	Các phép biến đổi mô hình . . . . .	7
2.4.1	Tọa độ thuần nhất (Homogeneous coordinates) . . . . .	7
2.4.2	Phép biến đổi Affine . . . . .	7
2.4.3	Các phép biến đổi hình học trong không gian ba chiều . . . . .	7
2.5	Mô hình máy ảnh . . . . .	9
2.5.1	Hiệu chỉnh điểm chính . . . . .	9
2.5.2	Xoay và dịch máy ảnh . . . . .	10
2.6	Mạng nơ ron nhân tạo . . . . .	11
2.6.1	Tổng quan . . . . .	11
2.6.2	Hàm kích hoạt . . . . .	12
2.6.3	Hàm mất mát . . . . .	12
2.6.4	Gradient Descent . . . . .	13
2.6.5	Lan truyền ngược . . . . .	13
2.6.6	Mạng nơ ron tích chập . . . . .	13
2.7	Bài toán Homogeneous Linear Least Squares . . . . .	15
2.8	Phân loại các hướng phát triển cho bài toán phát hiện vật thể qua ảnh đơn . . . . .	15
2.8.1	Phát hiện vật thể 3D qua hình ảnh đơn . . . . .	15
2.8.1.1	Phương pháp dựa trên điểm neo . . . . .	15
2.8.1.2	Phương pháp điểm neo tự do . . . . .	16
2.8.1.3	Phương pháp tiếp cận hai giai đoạn . . . . .	17
2.8.1.4	Đánh giá tiềm năng và khó khăn . . . . .	18

2.8.2	Phát hiện đối tượng 3D bằng hình ảnh đơn được hỗ trợ bởi công cụ ước tính độ sâu có sẵn . . . . .	18
2.8.2.1	Phương pháp dựa trên hình ảnh độ sâu . . . . .	18
2.8.2.2	Phương pháp dựa trên Pseudo-LiDAR . . . . .	19
2.8.2.3	Đánh giá tiềm năng và khó khăn . . . . .	20
<b>3</b>	<b>Tập dữ liệu và chỉ số đánh giá</b>	<b>21</b>
3.1	Tập dữ liệu . . . . .	21
3.1.1	KITTI 3D . . . . .	21
3.1.2	nuScenes . . . . .	22
3.1.3	Waymo Open . . . . .	23
3.2	Chỉ số đánh giá . . . . .	23
3.2.1	Chỉ số chính xác trung bình . . . . .	24
3.2.2	Chỉ số đánh giá của các tập dữ liệu . . . . .	25
3.2.2.1	KITTI 3D . . . . .	25
3.2.2.2	Waymo Open . . . . .	25
3.2.2.3	nuScenes . . . . .	26
<b>4</b>	<b>Các nghiên cứu liên quan</b>	<b>27</b>
4.1	Mô hình đề xuất . . . . .	27
4.1.1	Objects are Different: Flexible Monocular 3D Object Detection . . . . .	27
4.1.2	Geometry Uncertainty Projection Network for Monocular 3D Object Detection . . . . .	28
4.2	Các hướng phát triển mới . . . . .	30
4.2.1	Tích hợp mạng ước tính độ sâu phụ trợ . . . . .	30
4.2.2	Tăng cường dữ liệu đầu vào . . . . .	30
4.2.3	Giới thiệu thêm các ràng buộc mới trong quá trình huấn luyện . . . . .	31
<b>5</b>	<b>Mô hình cơ sở</b>	<b>32</b>
5.1	Tiền xử lý ảnh và tăng cường dữ liệu . . . . .	32
5.2	Hướng tiếp cận . . . . .	34
5.2.1	Sử dụng phân phối thay cho giá trị rời rạc . . . . .	34
5.2.2	Cơ chế học phân tầng . . . . .	35
5.3	Thử nghiệm lại . . . . .	36
<b>6</b>	<b>Giải pháp phát triển</b>	<b>40</b>
6.1	Lý thuyết . . . . .	42
6.1.1	Homography . . . . .	42
6.1.2	Ước tính ma trận homography . . . . .	43
6.2	Hàm mắt mát Homography . . . . .	44
6.2.1	Mô hình hóa các điểm ứng viên . . . . .	44
6.2.2	Tính toán Homography . . . . .	46
6.2.3	Hàm mắt mát . . . . .	46
6.3	Hiện thực hàm mắt mát Homography . . . . .	47
6.3.1	Hàm mắt mát tổng thể . . . . .	47
6.3.2	Chiến lược huấn luyện . . . . .	47
6.3.3	Thử nghiệm . . . . .	48

<b>7</b>	<b>Kết quả và đánh giá</b>	<b>49</b>
7.1	Kết quả định lượng . . . . .	49
7.2	Kết quả định tính . . . . .	50
<b>8</b>	<b>Tổng kết</b>	<b>55</b>
8.1	Thành quả đạt được . . . . .	55
8.2	Ý nghĩa . . . . .	55
8.3	Định hướng phát triển . . . . .	56

# Danh sách hình vẽ

2.1	Minh họa bài toán phát hiện vật thể 3D.	3
2.2	Minh họa các hệ tọa độ.	5
2.3	Sự khác nhau giữa cách mà máy ảnh và Lidar cảm nhận môi trường.	5
2.4	LiDAR có thể tạo bản đồ trực quan xung quanh nó.	6
2.5	Hệ thống Tesla Vision sử dụng máy ảnh với Autopilot.	7
2.6	Mô hình máy ảnh Pinhole.	10
2.7	Phép biến đổi Euclidean giữa thế giới và khung tọa độ máy ảnh.	10
2.8	Minh họa mạng nơ ron nhân tạo.	12
2.9	Minh họa cho từng kiểu chọn learning rate.	13
2.10	Minh họa cho quá trình lan truyền ngược.	14
2.11	Cấu trúc của CNN.	14
2.12	Phương pháp dựa trên điểm neo.	15
2.13	Phương pháp điểm neo tự do.	17
2.14	Phương pháp tiếp cận hai giai đoạn.	17
2.15	Phương pháp dựa trên hình ảnh độ sâu.	19
2.16	Phát hiện đối tượng 3D bằng hình ảnh đơn dựa trên Pseudo-LiDAR.	20
2.17	Mô hình mạng PatchNet.	20
3.1	Hình ảnh từ tập KITTI 3D.	22
3.2	Hình ảnh từ 6 camera tạo nên khung hình toàn cảnh trong tập nuScenes.	23
3.3	Hình minh họa cho công thức IoU.	24
4.1	MonoFlex giải quyết vấn đề dự đoán những vật thể nằm ngoài phân phối.	28
4.2	Thiết kế của mạng GUPNet.	29
4.3	Minh họa việc sai sót về độ cao dẫn đến sai sót của độ sâu bị khuếch đại.	29
5.1	Các kiểu tăng cường dữ liệu.	33
5.2	Hệ thống phân cấp nhiệm vụ của GUP Net.	36
5.3	Hàm lập lịch thời gian đa thức với tham số điều chỉnh.	37
5.4	So sánh về loss của 40 epoch đầu tiên.	37
5.5	Loss của 140 epoch.	38
6.1	Mô tả ràng buộc hình học của homography.	40
6.2	Vị trí của đối tượng mục tiêu bị ảnh hưởng toàn cục bởi các đối tượng khác.	42
6.3	Hình minh họa mối quan hệ hình chiếu giữa hai mặt phẳng Bird Eye's View và hình ảnh.	45
6.4	Điểm trung viền 2D và 3D của một đối tượng.	45
7.1	Trực quan hóa bản đồ nhiệt tâm 2D của vật thể.	51
7.2	Minh họa về trường hợp không được đánh nhãn trong tập kiểm thử KITTI.	52

7.3	Minh họa cải thiện của <b>H-GUPNet</b> so với mô hình cơ sở. . . . .	53
7.4	Trực quan hóa chi tiết kết quả phát hiện vật thể 3D. . . . .	54

# Chương 1

## Giới thiệu

### 1.1 Thực trạng

Thế giới đang ngày càng hiện đại hơn và hàng loạt công nghệ được ra đời để làm cho cuộc sống của con người trở nên dễ dàng hơn. Trong suốt thập kỉ vừa qua, cả hai hướng nghiên cứu hàn lâm và trong công nghiệp đều cố gắng để phát triển các phương tiện tự hành. Là một trong những công nghệ chính cho xe tự lái, việc phát hiện vật thể 3D đã nhận được rất nhiều sự quan tâm vì hệ thống tự hành cần phải thu được thông tin 3D của các đối tượng thông qua cảm biến để từ đó có thể hoạt động, đưa ra quyết định cho những hành động tiếp theo. Các hướng tiếp cận trong việc phát hiện đối tượng 3D hiện tại có thể được chia thành hai phương pháp: xử lý dựa trên tín hiệu của cảm biến LiDAR và xử lý dựa trên hình ảnh. Các thuật toán dựa trên cảm biến LiDAR mang tính chính xác và hiệu quả, nhưng chi phí cao hạn chế việc sử dụng và khai thác chúng trong công nghiệp. Các thuật toán dựa trên hình ảnh được nghiên cứu rộng rãi hơn vì chi phí thấp hơn, mang lại lợi ích cho người sử dụng. Tuy nhiên, cách tiếp cận dựa trên hình ảnh lại đối mặt với thử thách lớn hơn so với cách tiếp cận dựa trên LiDAR vì việc phục hồi thông tin 3D từ dữ liệu đầu vào hai chiều (hình ảnh 2D) vẫn là một vấn đề chưa thể giải quyết, đặc biệt khó khăn hơn khi sử dụng hình ảnh đơn làm dữ liệu đầu vào. Nhiều hướng xử lý đã được đề xuất để cố gắng giải quyết vấn đề xác định đối tượng như là suy ra thông tin độ sâu từ hình ảnh hay là tận dụng các ràng buộc về hình học và thông tin biết trước về hình dạng. Mặc dù thế, bài toán vẫn còn rất xa với lời giải. Vì thế, nhóm em xin được chọn đề tài “**Phát hiện vật thể 3D qua ảnh đơn với tiếp cận học sâu**” để thực hiện nghiên cứu.

### 1.2 Mục tiêu nghiên cứu

Mục tiêu nghiên cứu của nhóm đó là đề xuất một giải pháp phát hiện đối tượng 3D bằng hình ảnh đơn có thời gian xử lý ổn định và có độ chính xác tương đối tốt hơn so với các đề xuất ở những bài báo, nghiên cứu khoa học trước đây. Từ đó, góp phần vào công cuộc phát triển mảng Phát hiện đối tượng 3D bằng hình ảnh đơn nói chung và các ứng dụng thực tế của mảng này như xe tự hành, robot nói riêng.

### 1.3 Nhiệm vụ nghiên cứu

Đề tài tập trung vào các nhiệm vụ sau:

- Nghiên cứu cơ sở về bài toán phát hiện đối tượng 3D trong hình ảnh đơn.

- Khảo sát về các hướng tiếp cận và đánh giá.
- Trên cơ sở nghiên cứu và đánh giá trên, đề xuất một phương pháp phát hiện đối tượng 3D bằng hình ảnh đơn với cải tiến phù hợp với mục tiêu nhóm đã đề ra.
- Thực hiện thí nghiệm, phân tích và đánh giá kết quả của việc thí nghiệm phương pháp đã đề xuất.

## 1.4 Tổng quan về bối cảnh

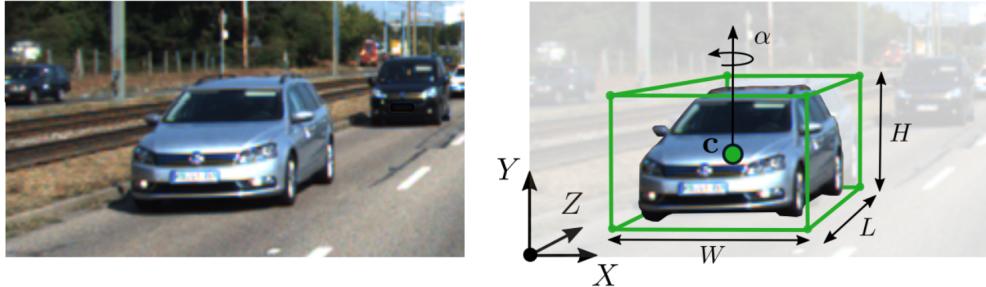
Báo cáo này bao gồm 8 chương. Chương 1 sẽ giới thiệu sơ lược về đề tài cũng như nêu rõ mục tiêu và những nhiệm vụ cần phải hoàn thành xuyên suốt báo cáo. Chương 2 sẽ giúp người đọc có những kiến thức nền tảng thông qua việc nêu lên một số định nghĩa, giải thích một số thông tin cụ thể để hỗ trợ cho quá trình đào sâu ở những chương sau. Chương 3 sẽ giới thiệu về các tập dữ liệu cũng như các chỉ số để đánh giá mô hình thường được sử dụng trong bài toán. Chương 4 sẽ cung cấp thông tin tổng quát về một số bài báo, công trình liên quan, đồng thời đánh giá và thực hiện lựa chọn một nghiên cứu đủ tốt để làm mô hình cơ sở. Chương 5 sẽ đi sâu vào kiến trúc của mô hình cơ sở và mô tả lại quá trình thí nghiệm lại. Chương 6 sẽ tập trung vào cơ sở lý thuyết của giải pháp đề xuất. Chương 7 nêu lên các kết quả của những thí nghiệm và đánh giá. Cuối cùng, chương 8 bao gồm việc tổng kết lại các mục tiêu và nhiệm vụ đã hoàn thành và đề ra một số định hướng cũng như thảo luận về đề tài.

# Chương 2

## Kiến thức nền tảng

### 2.1 Định nghĩa bài toán

Phát hiện đối tượng 3D trong hình ảnh đơn là việc xác định các đối tượng và vẽ các hộp 3D bao quanh trong một tấm ảnh. Đầu vào của bài toán là một tấm ảnh màu RGB 2D được chụp từ máy ảnh và không bao gồm thêm các thông tin bổ sung nào như thông tin về độ sâu, thông tin từ các cảm biến khác hay sử dụng cùng lúc nhiều hình ảnh. Đây cũng là khó khăn lớn nhất của bài toán vì chúng ta không thể suy ngược thông tin trong không gian 3D một cách chính xác chỉ từ một hình ảnh vì thiếu mất thông tin về độ sâu, đồng thời cũng khác với một bài toán phổ biến tương tự cũng sử dụng hình ảnh làm đầu vào nhưng là hai tấm ảnh được chụp cùng lúc từ máy ảnh kép. Khi ấy, sự chênh lệch của vật thể ở những điểm ảnh giữa hai tấm ảnh cũng đã chứa thêm thông tin về độ sâu.



Hình 2.1: Minh họa bài toán phát hiện đối tượng 3D. Bên trái là hình ảnh đầu vào, bên phải là đầu ra với hộp bao quanh. Lưu ý:  $\theta$  trên hình được thay thế bằng  $\alpha$ . Nguồn ảnh : [36].

Dự đoán của mỗi vật thể sẽ được biểu diễn bằng lớp (phân loại) của vật thể đó và hộp bao quanh trong không gian 3D. Thông thường một hộp bao quanh 3D sẽ được tham số hóa bởi vị trí tâm 3D  $[x, y, z]$ , kích thước  $[h, w, l]$  và các hướng  $[\theta, \phi, \psi]$  liên quan đến hệ tọa độ tham chiếu đã được xác định trước. Trong hầu hết bối cảnh của xe tự hành, chỉ có góc hướng tới  $\theta$  nằm xung quanh trục dọc (yaw angle) được xem xét và bài toán trong báo cáo này cũng tương tự. Đầu vào và đầu ra của bài toán được thể hiện ở hình 2.1.

### 2.2 Các hệ tọa độ

Trong bài toán này, máy ảnh được sử dụng để lấy các điểm trong thế giới thực và chiếu chúng xuống mặt phẳng 2D để trở thành hình ảnh đầu vào. Đầu ra của bài toán một phần tập

trung vào tìm tâm 3D của vật thể, tức là từ hình ảnh đầu vào cần suy ngược lại tọa độ 3D ở thế giới thực. Các hệ tọa độ liên quan trong bài toán tiếp theo sẽ được làm rõ bên dưới.

### **2.2.1 Hệ tọa độ thế giới thực (World coordinate system - 3D)**

Là hệ tọa độ tương ứng với thế giới thực. Một điểm trong hệ tọa độ này có thể được kí hiệu là  $[x, y, z]$ . Mục tiêu của bài toán là tìm được tọa độ của vật thể trong thế giới thực, từ đó suy ra được tọa độ của hộp bao quanh 3D vật thể.

### **2.2.2 Hệ tọa độ máy ảnh (Camera coordinate system - 3D)**

Là hệ tọa độ được đo lường tương đối so với gốc và hướng của máy ảnh. Trục z của hệ tọa độ máy ảnh thường hướng ra ngoài hoặc hướng vào trong ống kính máy ảnh. Hệ tọa độ thế giới thực và hệ tọa độ máy ảnh có thể chuyển đổi qua lại bằng các phép tịnh tiến và phép xoay. Ma trận biến đổi các điểm trong hệ tọa độ thế giới thực thành hệ tọa độ máy ảnh được gọi là ma trận ngoại máy ảnh. Ma trận ngoại máy ảnh thay đổi nếu vị trí vật lý hoặc hướng của máy ảnh thay đổi.

### **2.2.3 Hệ tọa độ hình ảnh (Image coordinate system - 2D)**

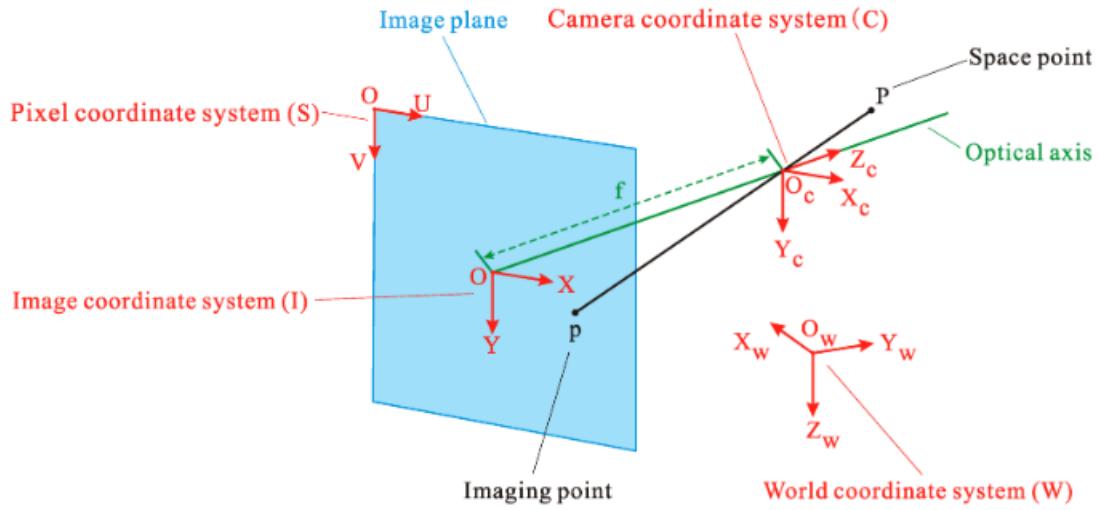
Là hệ tọa độ mà các điểm 3D trong hệ tọa độ máy ảnh đã được chiếu xuống mặt phẳng 2D (theo hướng trục z). Mặt phẳng 2D này chính là hình ảnh được chụp lại bởi máy ảnh. Phép chiếu này là một phép biến đổi mất dữ liệu, có nghĩa là không thể đảo ngược được việc chiếu từ hệ tọa độ máy ảnh đến mặt phẳng 2D. Đó là do thông tin độ sâu đã bị mất, vì thế khi nhìn vào hình ảnh của máy ảnh, ta không thể biết được độ sâu thực tế của các điểm. Tuy nhiên, ở bài toán này, tọa độ 3D của vật thể được suy ngược thông qua một độ sâu cũng được suy ra từ hình ảnh đầu vào.

### **2.2.4 Hệ tọa độ điểm ảnh (Pixel coordinate system - 2D)**

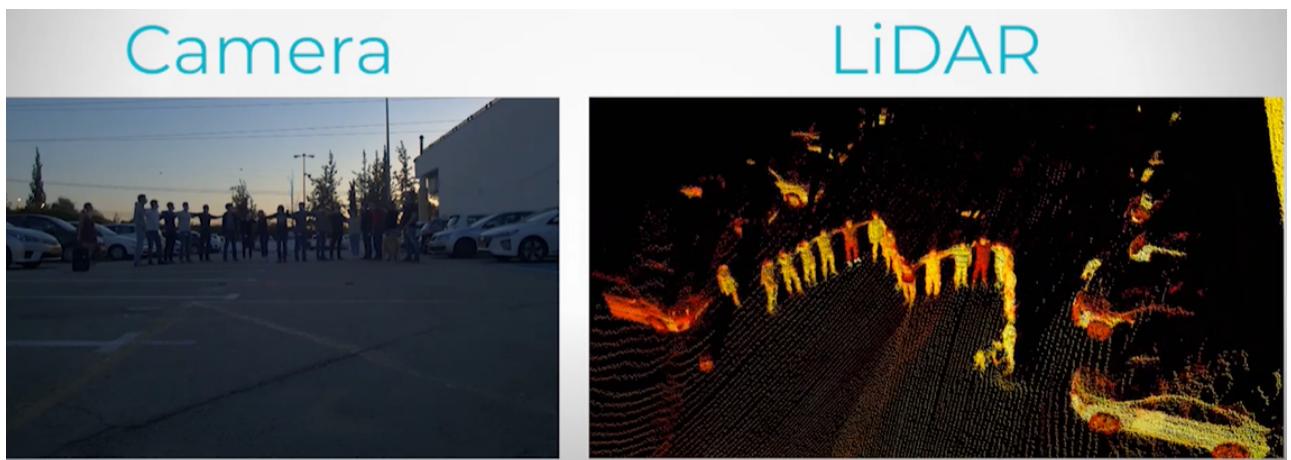
Hệ tọa độ này biểu diễn các giá trị nguyên bằng cách rời rạc hóa những điểm trong hệ tọa độ hình ảnh. Hệ tọa độ điểm ảnh của một hình ảnh là những giá trị rời rạc nằm trong phạm vi có được từ việc chia hệ tọa độ hình ảnh cho chiều rộng và chiều dài điểm ảnh (các thông số của máy ảnh). Ta có thể chuyển đổi từ hệ tọa độ máy ảnh sang hệ tọa độ điểm ảnh (through qua trung gian là hệ tọa độ hình ảnh) bằng cách sử dụng ma trận nội máy ảnh.

## **2.3 Các cảm biến đầu vào**

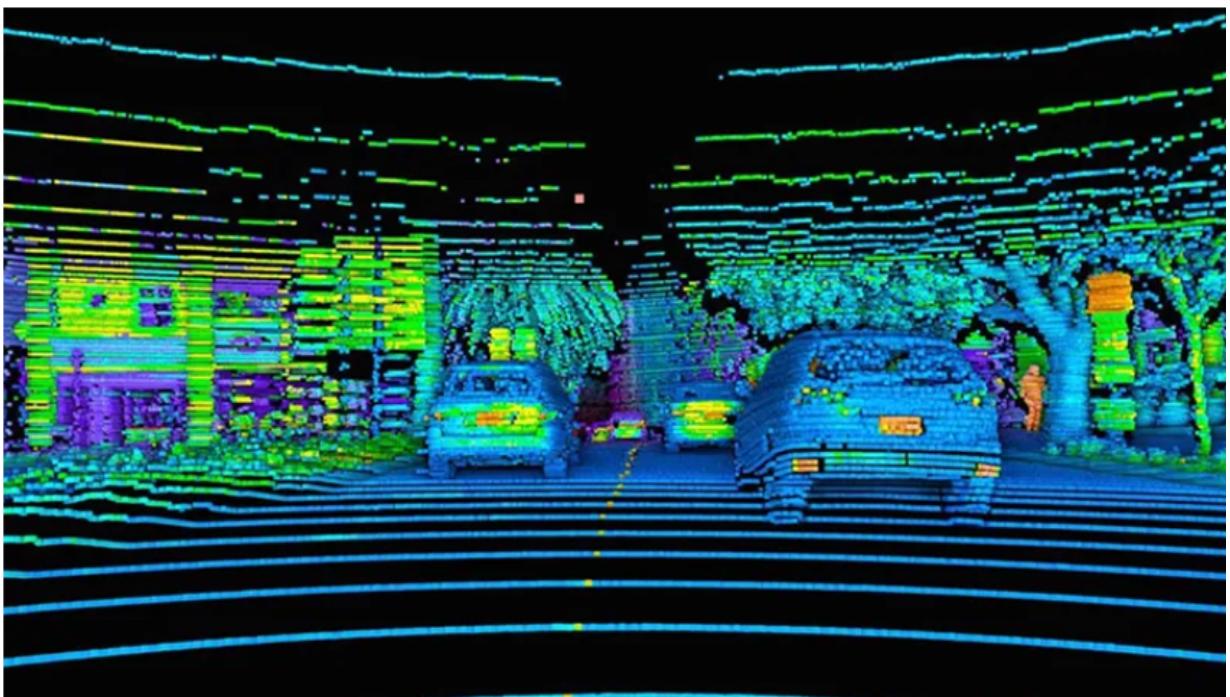
Các mô hình phát hiện vật thể 3D trong hình ảnh đơn ghi nhận thông tin của môi trường bằng một trong hai cơ chế cảm biến chính. Máy ảnh là một trong những phương tiện phổ biến nhất vì chúng có thể ghi lại môi trường một cách trực quan giống như cách con người làm với thị giác. LiDAR, một công nghệ tương đối mới hơn được sử dụng trong ô tô sử dụng một loạt các xung ánh sáng để đo khoảng cách. Sự khác nhau giữa hai cảm biến máy ảnh và LiDAR được thể hiện trong hình 2.3.



Hình 2.2: Minh họa các hệ tọa độ. Nguồn ảnh : [67].



Hình 2.3: Sự khác nhau giữa cách mà máy ảnh và LiDAR cảm nhận môi trường. Nguồn ảnh : Innoviz Technologies.



Hình 2.4: LiDAR có thể tạo bản đồ trực quan xung quanh nó. Nguồn ảnh : medium.

### 2.3.1 LiDAR

LiDAR có thể giúp ô tô tự lái tạo ra một bản đồ trực quan dựa trên số liệu mà nó nhận được từ các xung ánh sáng. Hệ thống LiDAR gửi hàng nghìn xung mỗi giây để tạo bản đồ 3D bằng phần mềm tích hợp nhằm cung cấp cho ô tô thông tin về môi trường xung quanh. Điều này cung cấp chế độ xem 360 độ giúp xe di chuyển trong mọi điều kiện. LiDAR được sử dụng phối hợp với máy ảnh trong ô tô tự lái, vì vậy bản thân LiDAR không phải là một giải pháp độc lập.

### 2.3.2 Máy ảnh

Thay vì các xung ánh sáng, máy ảnh sử dụng dữ liệu trực quan được trả về từ hệ thống quang học trong ống kính tới phần mềm tích hợp để phân tích thêm. Với sự phát triển của mạng lưới thần kinh và thuật toán thị giác máy tính, các đối tượng có thể được xác định để cung cấp thông tin về ô tô khi nó đang lái.

### 2.3.3 Nhược điểm của máy ảnh và LiDAR

LiDAR có thể nhìn thấy các vật thể ngay cả trong điều kiện thời tiết nguy hiểm, nhưng không phải lúc nào nó cũng đáng tin cậy. LiDAR bị ảnh hưởng bởi độ ổn định của bước sóng và độ nhạy của máy dò. LiDAR cũng đắt hơn và cần nhiều không gian hơn để triển khai. Một vấn đề khác với LiDAR là nhận dạng hình ảnh, thứ mà máy ảnh làm tốt hơn nhiều. LiDAR yêu cầu xử lý dữ liệu nhiều hơn trong phần mềm để tạo hình ảnh và nhận dạng đối tượng.

Máy ảnh mặc dù rẻ hơn và đáng tin cậy hơn như một hệ thống tầm nhìn, nhưng máy ảnh không có tính năng phát hiện phạm vi của LiDAR. Máy ảnh cũng không thể nhìn đủ rõ trong điều kiện thời tiết.



Hình 2.5: Hệ thống Tesla Vision sử dụng máy ảnh với Autopilot. Nguồn ảnh: Tesla.

## 2.4 Các phép biến đổi mô hình

### 2.4.1 Tọa độ thuần nhất (Homogeneous coordinates)

Trong toán học, tọa độ thuần nhất thường được sử dụng trong phép chiếu hình học. Giả sử, ta có một điểm  $(x, y)$  trong mặt phẳng hình ảnh. Để biểu diễn điểm này trong tọa độ đồng nhất, ta có thể thêm một tọa độ có giá trị 1 vào vị trí tọa độ thứ ba  $(x, y, 1)$ . Nếu tọa độ đồng nhất của một điểm nhân cho một hệ số khác 0 thì tọa độ kết quả vẫn biểu diễn cho điểm ban đầu. Tọa độ đồng nhất trong không gian chiếu được sử dụng vì nó thuận tiện hơn so với việc sử dụng hệ tọa độ Descartes trong không gian Euclid. Việc sử dụng tọa độ đồng nhất cho phép ta thực hiện kết hợp các phép biến đổi như phép xoay, phép tịnh tiến, phép biến đổi tỉ lệ trong 1 phép nhân ma trận thay vì thực hiện lần lượt từng phép riêng lẻ như nhân ma trận (đối với phép xoay và biến đổi tỉ lệ) và phép cộng (đối với phép tịnh tiến).

### 2.4.2 Phép biến đổi Affine

Phép biến đổi Affine là phép biến đổi tọa độ điểm đặc trưng của đối tượng thành tập tương ứng các điểm mới để tạo ra các hiệu ứng cho toàn đối tượng. – Ví dụ: phép biến đổi tọa độ với chỉ 2 điểm đầu cuối của đoạn thẳng tạo thành 2 điểm mới mà khi nối chúng với nhau tạo thành đoạn thẳng mới.

### 2.4.3 Các phép biến đổi hình học trong không gian ba chiều

Phép biến đổi hình học trong không gian ba chiều có thể được biểu diễn như sau :

$$M' = T \cdot M \quad (2.1)$$

Trong đó  $M, M'$  có dạng :

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}, \text{ và } T \text{ gọi là ma trận biến đổi có dạng tổng quát } \begin{bmatrix} a_1 & a_2 & a_3 & m \\ b_1 & b_2 & b_3 & n \\ c_1 & c_2 & c_3 & p \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

### Các phép biến đổi hình học

#### 1. Phép bất biến

- Biến điểm M thành chính nó :

$$M(x, y, z) \xrightarrow{T} M'(x', y', z') \equiv M(x, y, z)$$

- Ma trận biến đổi :

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 2. Phép tịnh tiến

- Tịnh tiến M một vector  $\vec{v}(m, n, p)$  thành điểm M' :

$$M(x, y, z) \xrightarrow{T} M'(x', y', z') \equiv M'(x + m, y + n, z + p)$$

- Ma trận biến đổi :

$$T = \begin{bmatrix} 1 & 0 & 0 & m \\ 0 & 1 & 0 & n \\ 0 & 0 & 1 & p \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 3. Phép biến đổi tỉ lệ tại gốc tọa độ

- Co dãn so với gốc tọa độ:

$$M(x, y, z) \xrightarrow{T} M'(x', y', z') \equiv M'(xS_x, yS_y, zS_z)$$

Trong đó  $S_x, S_y, S_z$  là các hệ số tỉ lệ khác 0.

- Ma trận biến đổi :

$$T = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

#### 4. Phép quay

Khi điểm M quay quanh các trục Ox, Oy, Oz một góc  $\alpha$  thành điểm M', ta có các ma trận biến đổi lần lượt như sau :

- Phép quay quanh trục Ox. Ma trận biến đổi :

$$T_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Phép quay quanh trục Oy. Ma trận biến đổi :

$$T_y = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Phép quay quanh trục Oz. Ma trận biến đổi :

$$T_z = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 5. Phép biến đổi kết hợp

- Điểm M qua phép biến đổi  $T_1$  thành  $M_1$ ,  $M_1$  qua phép biến đổi  $T_2$  thành  $M_2$ , suy ra tồn tại một phép biến đổi biến M thành  $M_2$ .
- $T$  được gọi là phép biến đổi kết hợp giữa  $T_1$  và  $T_2$ , khi đó :

$$T = T_1 \times T_2$$

## 2.5 Mô hình máy ảnh

Mô hình máy ảnh đơn giản nhất là mô hình Pinhole (hay còn gọi là mô hình máy ảnh lỗ kim) mô tả mối quan hệ toán học của phép chiếu các điểm trong không gian ba chiều lên mặt phẳng hình ảnh. Đặt tâm hình chiếu là gốc của một hệ tọa độ Euclide và mặt phẳng  $Z = f$  được gọi là mặt phẳng ảnh hoặc mặt phẳng tiêu cự. Bằng mô hình máy ảnh Pinhole, một điểm trong không gian có tọa độ  $(X, Y, Z)^T$  được ánh xạ tới điểm trên mặt phẳng hình ảnh  $(\frac{fX}{Z}, \frac{fY}{Z}, f)^T$  bằng cách sử dụng các hình tam giác như trong hình 2.6. Bỏ qua tọa độ hình ảnh cuối cùng, ánh xạ từ không gian thế giới 3D sang tọa độ hình ảnh 2D là:

$$(X, Y, Z)^T = \left( \frac{fX}{Z}, \frac{fY}{Z} \right)^T \quad (2.2)$$

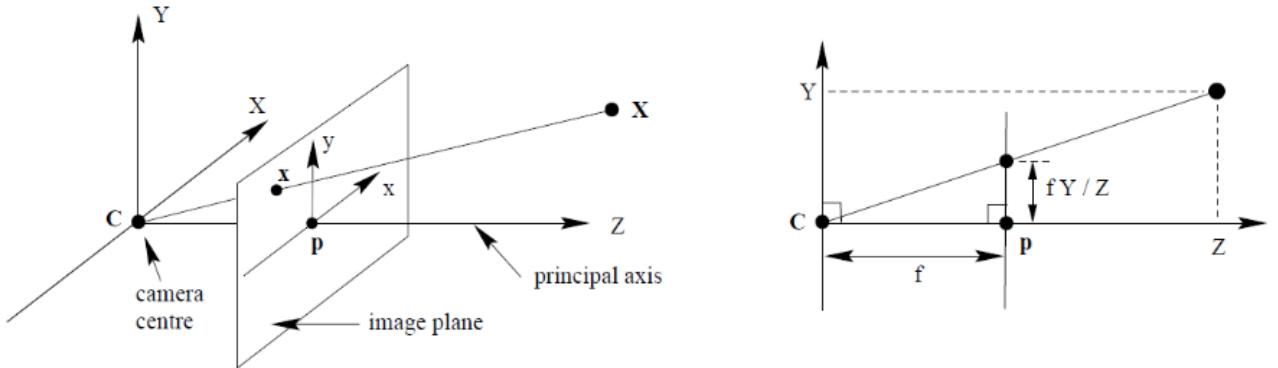
Giả sử các điểm trong thế giới thực và hình ảnh được biểu diễn theo tọa độ thuần nhất, thì phép chiếu trung tâm có thể được biểu diễn đơn giản dưới dạng ánh xạ tuyến tính giữa các tọa độ thuần nhất của chúng theo phép nhân ma trận :

$$\begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (2.3)$$

### 2.5.1 Hiệu chỉnh điểm chính

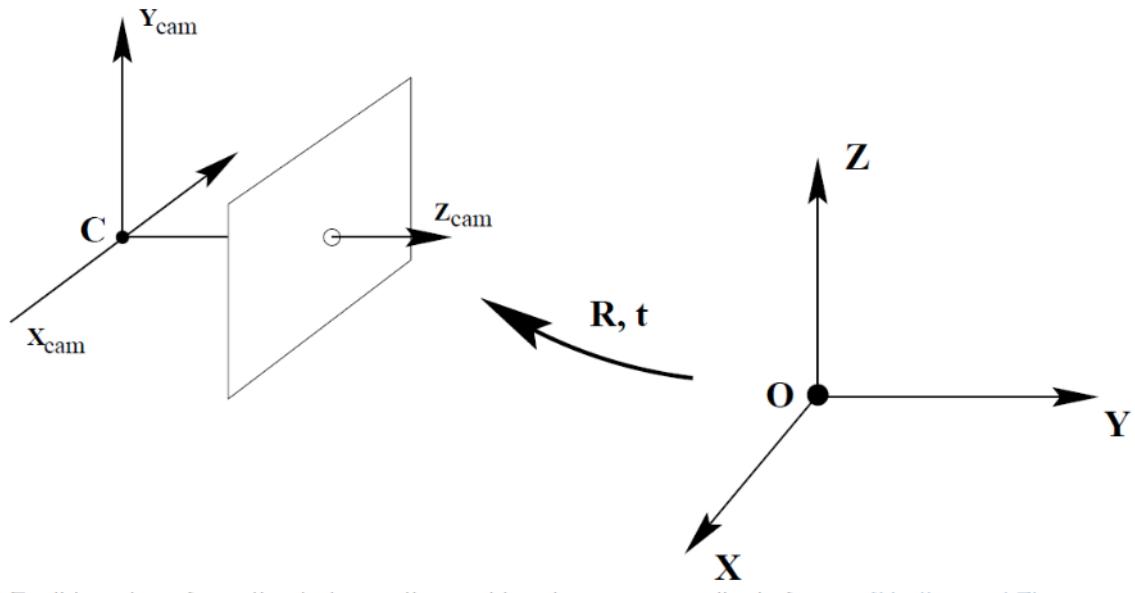
Về lý thuyết, gốc tọa độ trong mặt phẳng hình ảnh được giả định là tại điểm chính. Điều này có thể không đúng trong thực tế, do đó, phương trình 2.3 được điều chỉnh lại như sau:

$$\begin{bmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \quad (2.4)$$



Hình 2.6: Mô hình máy ảnh Pinhole. Tâm chiếu được gọi là tâm máy ảnh hay tâm quang học. Đường kẻ từ tâm máy ảnh vuông góc với mặt phẳng ảnh gọi là trực chính hay tia chính. Điểm mà trực chính cắt mặt phẳng ảnh gọi là điểm chính. Mặt phẳng qua tâm máy ảnh song song với mặt phẳng ảnh được gọi là mặt phẳng chính của máy ảnh. C là tâm máy ảnh và P là điểm chính. Tâm máy ảnh ở đây được đặt tại gốc tọa độ. Nguồn ảnh : hedivision.

Ma trận đầu tiên ở phía bên phải của biểu thức 2.4 được gọi là ma trận hiệu chỉnh máy ảnh, thường được ký hiệu bằng  $K$ . Trong đó  $(p_x, p_y)$  là tọa độ của điểm chính,  $(f_x, f_y)$  biểu diễn độ dài tiêu cự của máy ảnh theo kích thước pixel theo trục x và trục y tương ứng.



Hình 2.7: Phép biến đổi Euclide giữa thế giới và khung tọa độ máy ảnh. Nguồn ảnh : hedivision.

### 2.5.2 Xoay và dịch máy ảnh

Các chỉ số *cam* trong các phương trình trên là để chỉ ra rằng các điểm được biểu diễn dưới dạng tọa độ máy ảnh. Các điểm trong không gian được xác định bởi hệ tọa độ thế giới thực. Tọa độ máy ảnh và tọa độ thế giới có liên quan với nhau bằng các phép xoay và dịch . Như được thể hiện trong Hình 2.7, nếu  $\mathbf{X} = (X, Y, Z, 1)^T$  là tọa độ của điểm trong hệ tọa độ thế giới, sau đó  $\mathbf{X}_{cam}$  được biến đổi bởi:

$$\mathbf{X}_{cam} = [R \ t] \mathbf{X} \quad (2.5)$$

$\mathbf{R}$  ở đây là ma trận xoay  $3 \times 3$  và  $\mathbf{t}$  là ma trận dịch  $3 \times 1$ . Kết hợp mọi thứ lại với nhau, công thức ánh xạ của máy ảnh Pinhole trong hệ tọa độ thế giới  $\mathbf{x}$  được xác định như sau:

$$\mathbf{x} = \mathbf{K} [\mathbf{R} \ \mathbf{t}] \mathbf{X} \quad (2.6)$$

Vì vậy, ma trận máy ảnh Pinhole chung,  $\mathbf{P}$ , có thể được biểu diễn bằng:

$$\mathbf{P} = \mathbf{K} [\mathbf{R} \ \mathbf{t}] = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix} \quad (2.7)$$

và nó có 9 bậc tự do.

1. Ba cho  $\mathbf{K}$ , cụ thể là,  $f$ ,  $p_x$ , và  $p_y$ .
2. Ba cho ma trận xoay  $\mathbf{R}$ .
3. Ba cho ma trận dịch  $\mathbf{t}$ .
  - Thông số máy ảnh bên trong,  $\mathbf{K}$ , hiển thị hướng bên trong của máy ảnh và nó được cố định.
  - Các tham số bên ngoài,  $\mathbf{R}$  và  $\mathbf{t}$  hiển thị hướng và vị trí của máy ảnh đối với hệ tọa độ thế giới.

## 2.6 Mạng nơ ron nhân tạo

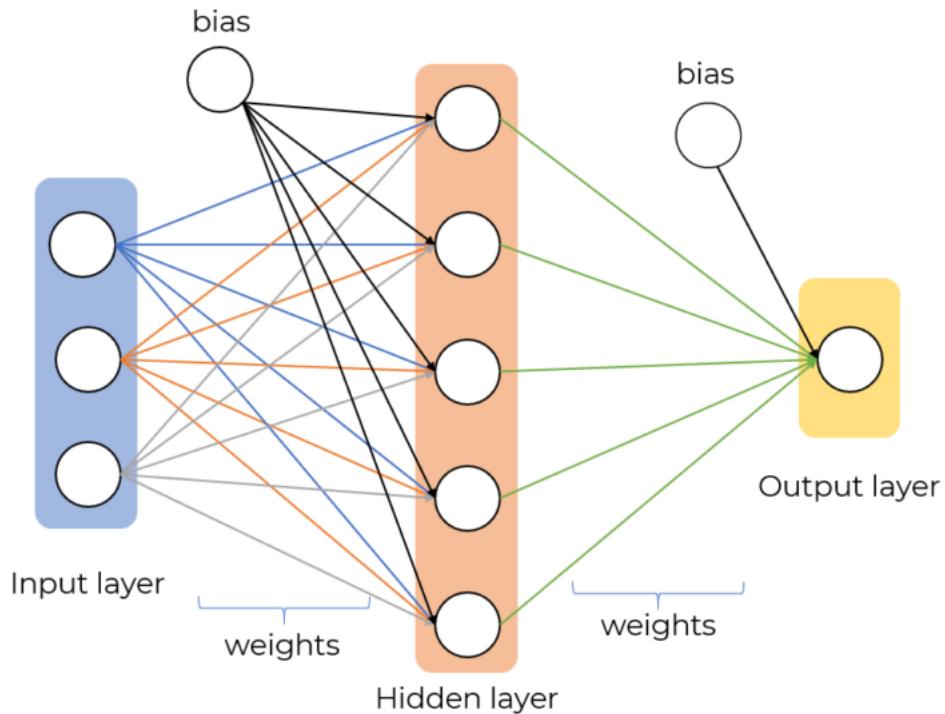
### 2.6.1 Tổng quan

Mạng nơ ron nhân tạo (Neural Network) là một trong những khái niệm cơ bản nhất của học sâu. Chúng được lấy cảm hứng từ cách các nơ ron trong não người hoạt động. Các nơ-ron này được kết nối với nhau nhằm phục vụ việc xử lý và truyền tín hiệu. Mạng nơ ron nhân tạo đang là mô hình học máy phổ biến nhất trong thời gian gần đây. Với khả năng xấp xỉ hàm mạnh mẽ, chúng được sử dụng trong nhiều bài toán liên quan đến dữ liệu dạng bảng, hình ảnh, âm thanh, v.v.. Hình 2.8 minh họa mạng nơ ron nhân tạo.

Mạng nơ ron nhân tạo có thể được xem như một hàm phức tạp. Dạng đơn giản nhất là Feed-forward Network (mạng chuyển tiếp). Mạng được tổ chức dưới dạng các lớp (layer), gồm lớp đầu vào (input), các lớp ẩn ở giữa (hidden-layers) và lớp đầu ra (output). Mỗi lớp sẽ bao gồm nhiều nơ ron (được gọi là node), nhận đầu vào là một véc tơ có kích thước cố định, thực hiện tính toán và sau đó tạo ra một véc tơ đầu ra cũng có kích thước cố định. Lớp đầu vào sẽ chứa các node chứa các giá trị từ dữ liệu đầu vào (ví dụ như giá trị pixel trong một tấm ảnh). Gọi  $\mathbf{h}^{(l-1)} \in \mathbb{R}^m$  là đầu vào của lớp thứ  $l$  và  $\mathbf{h}^{(l)} \in \mathbb{R}^n$  là đầu ra, cách thức hoạt động của một lớp có thể được mô tả như sau:

$$\mathbf{h}^{(l)} = f(\mathbf{W}^{(l)} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \quad (2.8)$$

trong đó,  $\mathbf{W}^{(l)} \in \mathbb{R}^{m \times n}$  và  $\mathbf{b}^{(l)} \in \mathbb{R}^n$  được gọi là ma trận trọng số và véc tơ bias ở lớp thứ  $l$ ,  $f$  là một hàm phi tuyến được gọi là hàm kích hoạt. Sự phi tuyến này giúp mạng nơ ron có khả năng xấp xỉ hàm.



Hình 2.8: Minh họa mạng nơ ron nhân tạo. Nguồn ảnh : pyimagedata.

### 2.6.2 Hàm kích hoạt

Hàm kích hoạt mang đến sự phi tuyến trong quá trình hoạt động của mạng nơ ron. Nó giúp mô hình có thể tổng quát hóa hoặc thích nghi được với sự đa dạng của dữ liệu để phân biệt các đầu ra. Nếu không có hàm kích hoạt, đầu ra sẽ chỉ là một tổ hợp tuyến tính của các giá trị đầu vào. Một số các hàm kích hoạt thường được dùng có thể kể đến như hàm sigmoid ( $\sigma$ ), tanh, ReLU, LeakyReLU, Maxout, ELU, v.v. Công thức của một số hàm được liệt kê như sau:

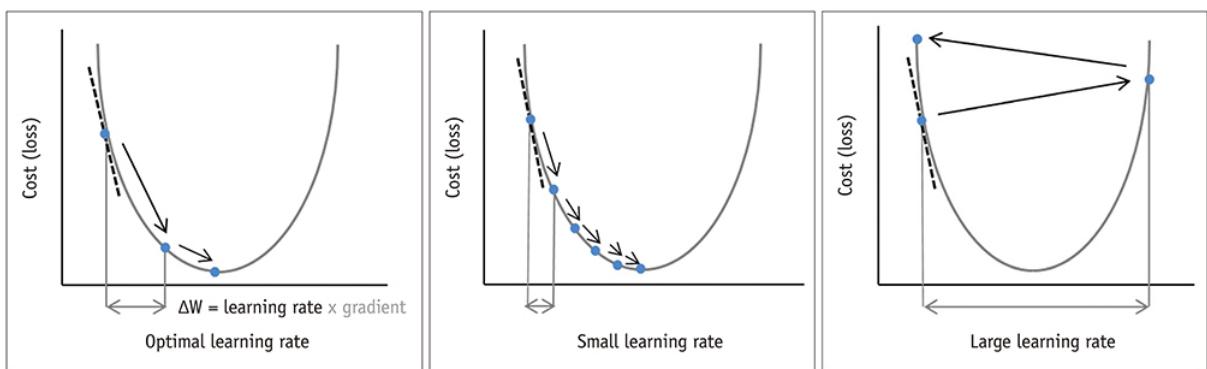
$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.9)$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.10)$$

$$ReLU(x) = \max(x, 0) \quad (2.11)$$

### 2.6.3 Hàm mất mát

Việc huấn luyện mạng nơ ron nhân tạo là một quá trình tối ưu các tham số của các lớp trong mạng nhằm mục đích tối thiểu giá trị của hàm mất mát. Hàm mất mát định nghĩa độ sai lệch của giá trị mà mô hình dự đoán so với giá trị thực tế, thông thường là tổng hoặc trung bình của sai số từ các đối tượng của dữ liệu huấn luyện. Việc cố gắng tìm nghiệm cho tất cả tham số nhằm mục đích thỏa mãn cực tiểu toàn cục của hàm mất mát là rất phức tạp. Vì thế, hầu hết các thuật toán cố gắng tối thiểu hàm mất mát dựa vào gradient  $\nabla L$  của hàm mất mát đối với các tham số của mạng nơ ron. Điều này sẽ được thảo luận chi tiết hơn ở phần 2.6.4.



Hình 2.9: Minh họa cho từng kiểu chọn learning rate. Bên trái là learning rate tối ưu, ở giữa là learning quá nhỏ, bên phải là learning rate quá lớn. Nguồn: Analytics India Magazine.

#### 2.6.4 Gradient Descent

Gradient Descent là một thuật toán tối ưu lặp. Mục tiêu của thuật toán này là tìm các giá trị tối ưu của tham số mô hình nhằm mục đích tối thiểu giá trị của hàm mất mát. Thuật toán hoạt động bằng cách lặp lại việc điều chỉnh tham số theo hướng ngược lại với hướng đạo hàm của hàm mất mát. Gradient Descent có nhiều biến thể khác như Stochastic Gradient Descent (SGD), Mini-batch SGD. Ngoài ra, để điều chỉnh tốc độ đi đến cực trị nhanh hay chậm nhờ vào hệ số learning rate (tốc độ học) ( $\eta$ ). Từ đó, công thức được thiết lập như sau:

$$\theta_{new} = \theta - \eta \nabla_{\theta} \quad (2.12)$$

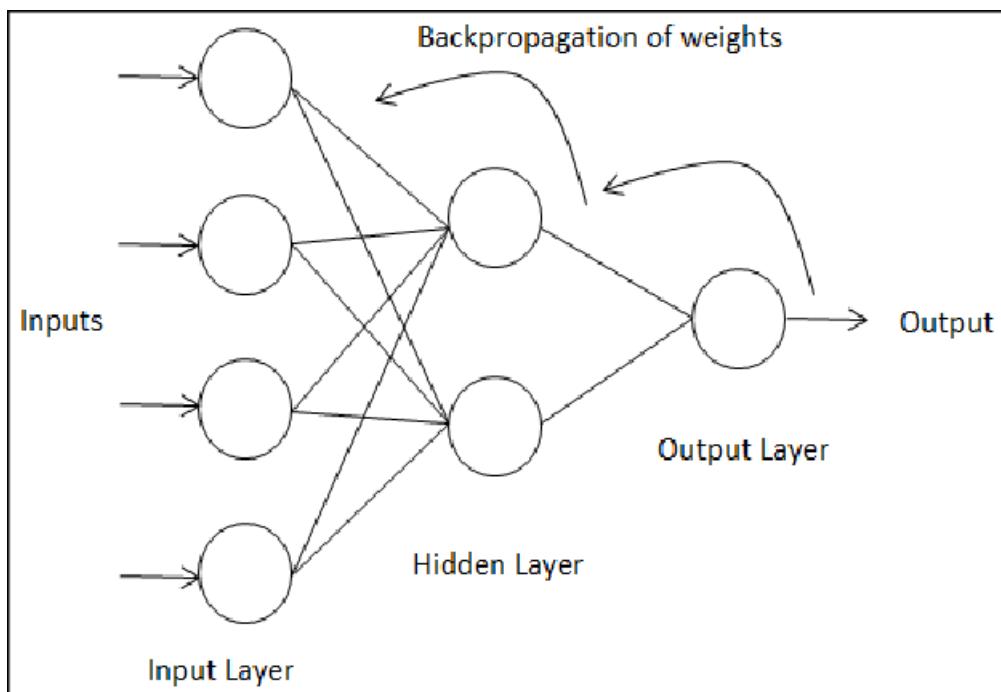
trong đó  $\theta$  đại diện cho tham số của mô hình,  $\nabla_{\theta}$  là đạo hàm của hàm mất mát theo tham số, dấu ‘-’ biểu diễn cho việc điều chỉnh tham số theo hướng ngược chiều của đạo hàm. Việc chọn learning rate cũng mang một ý nghĩa rất quan trọng, nó đại diện cho việc tham số được cập nhật nhiều hay ít. Nếu chọn learning rate quá lớn thì bộ tham số sẽ không thể hội tụ vì tham số khi cập nhật sẽ ở vị trí phía còn lại của cực trị. Nếu chọn learning rate quá nhỏ thì quá trình hội tụ sẽ diễn ra rất lâu. Quá trình được minh họa như ở hình 2.9.

#### 2.6.5 Lan truyền ngược

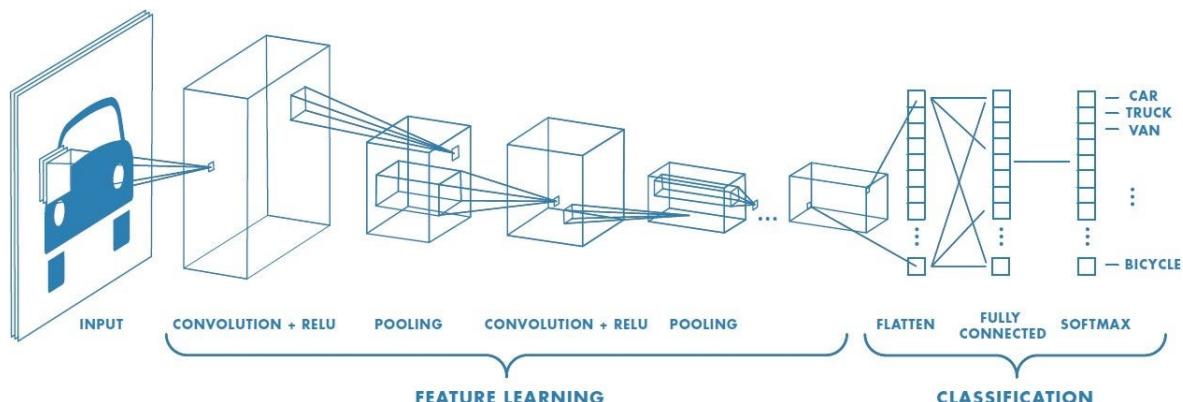
Do đầu vào của lớp sau là đầu ra của lớp trước nó nên quá trình tính toán đầu ra của mô hình là một quá trình chuyển tiếp từ đầu đến cuối. Khi tính toán đạo hàm của hàm mất mát theo từng ma trận trọng số hoặc bias một cách trực tiếp sẽ rất phức tạp vì hàm mất mát không phụ thuộc trực tiếp vào các hệ số. Lan truyền ngược là một thuật toán giúp tính gradient ngược từ lớp cuối cùng đến lớp đầu tiên. Trong quá trình truyền ngược này, chúng ta sẽ áp dụng quy tắc chuỗi (đạo hàm của hàm hợp) để tính đạo hàm của trọng số và bias ở các lớp trước từ các lớp sau. Hình 2.10 minh họa quá trình lan truyền ngược.

#### 2.6.6 Mạng nơ ron tích chập

Mạng nơ ron tích chập (Convolutional Neural Network, CNN) có cấu trúc rất giống với mạng nơ ron nhân tạo đã mô tả ở phần 2.6.1: hình thành từ những nơ ron với các ma trận trọng số và vec tơ bias. Mạng nơ ron tích chập tận dụng các nguyên tắc trong đại số tuyến tính, đặc biệt là phép nhân ma trận để xác định những đặc trưng trong một bức ảnh. Mạng nơ ron tích chập khác biệt với các loại mạng nơ ron nhân tạo khác bởi vì hiệu suất vượt trội khi thao tác trên



Hình 2.10: Minh họa cho quá trình lan truyền ngược. Nguồn ảnh : i2tutorials



Hình 2.11: Cấu trúc của CNN. Nguồn ảnh : saturncloud.

đầu vào là hình ảnh hoặc tín hiệu âm thanh. Thành phần của mạng gồm ba kiểu lớp chính: lớp Convolution (tích chập), lớp Pooling (tổng hợp) và lớp Fully-connected. Minh họa ở hình 2.11.

Ở lớp tích chập, có một thành phần được gọi là bộ phát hiện đặc trưng hay Kernel. Kernel thông thường sẽ có kích thước là  $3 \times 3$ , sẽ được dùng để duyệt qua một vùng trong không gian đầu vào bằng cách thực hiện phép nhân ma trận (tích vô hướng) giữa những điểm ảnh và Kernel. Sau đó, Kernel sẽ dịch chuyển 1 khoảng và quá trình này được lặp lại đến khi Kernel duyệt hết hình ảnh. Lớp Pooling thường được sử dụng ngay sau lớp tích chập để đơn giản hóa thông tin đầu ra nhằm giảm bớt số lượng ron nhưng vẫn giữ được các thuộc tính quan trọng mà lớp tích chập đã lọc. Có 2 loại lớp Pooling chính là Max Pooling và Average Pooling. Cuối cùng, lớp Fully-connected thực hiện phân lớp đầu vào dựa trên những đặc trưng được chiết xuất từ những lớp trước đó.

## 2.7 Bài toán Homogeneous Linear Least Squares

Ta có dạng bài toán:

$$\mathbf{Ax} = \mathbf{0} \quad (2.13)$$

được gọi là bài toán Homogeneous Linear Least Squares (bình phương tuyến tính thuần nhất nhỏ nhất), là một trường hợp cụ thể của  $\mathbf{Ax} = \mathbf{b}$ . Tuy nhiên, phương trình 2.13 không giống với bài toán tổng quát vì ta không thể giải bằng ma trận nghịch đảo hoặc ma trận giả nghịch đảo. Thay vào đó, ta có thể sử dụng Singular Value Decomposition (SVD). Cho ma trận  $\mathbf{A} \in \mathbb{R}^{m \times n}$  bất kì, ta có:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \quad (2.14)$$

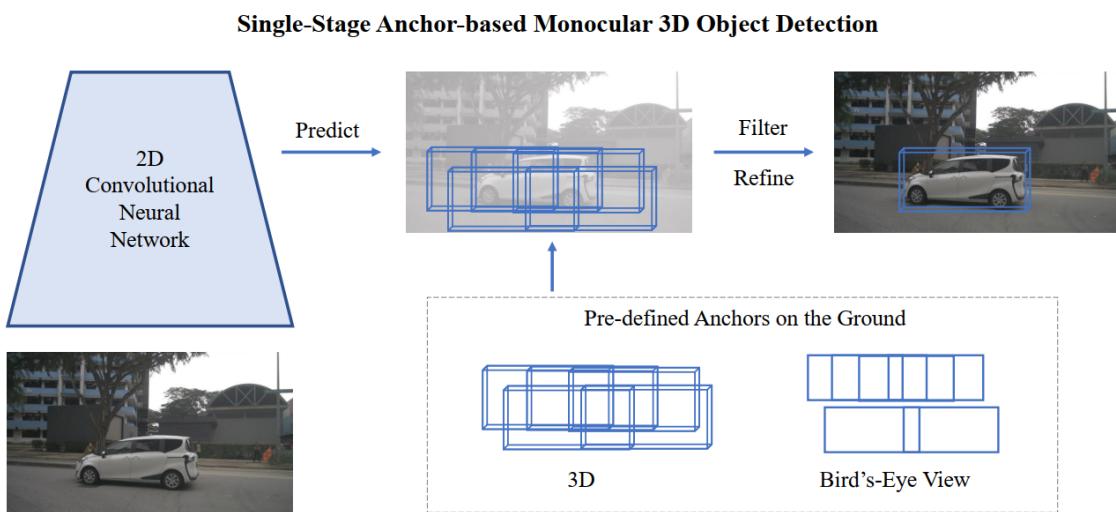
trong đó,  $\mathbf{U} \in \mathbb{R}^{m \times m}$  là một ma trận trực giao với các cột là véc tơ riêng của  $\mathbf{A}$ ,  $\Sigma \in \mathbb{R}^{m \times n}$  là ma trận chéo có các giá trị không âm, được gọi là singular values (các giá trị ở hàng chéo là các trị riêng của  $\mathbf{A}$ ),  $\mathbf{V} \in \mathbb{R}^{n \times n}$  là một ma trận trực giao.

## 2.8 Phân loại các hướng phát triển cho bài toán phát hiện vật thể qua ảnh đơn

### 2.8.1 Phát hiện vật thể 3D qua hình ảnh đơn

Lấy cảm hứng từ những cách tiếp cận trong Phát hiện 2D, một giải pháp đơn giản nhất đối với phát hiện vật thể 3D bằng hình ảnh đơn là trực tiếp hồi quy các thông số của hộp bao quanh 3D từ hình ảnh thông qua một mạng nơ ron tích chập. Những phương pháp trực tiếp hồi quy mượn những thiết kế từ các kiến trúc mạng phát hiện 2D do có sự tương đồng và đồng thời có thể được huấn luyện theo một quy trình đóng kín (end-to-end). Những hướng tiếp cận này có thể được chia thành phương pháp một giai đoạn/hai giai đoạn hoặc phương pháp dựa trên điểm neo (anchor-based)/điểm neo tự do (anchor-free).

#### 2.8.1.1 Phương pháp dựa trên điểm neo



Hình 2.12: Phương pháp dựa trên điểm neo. Nguồn ảnh : [40].

Cách tiếp cận này dựa trên một tập các điểm neo 2D và 3D được đặt tại mỗi điểm ảnh (pixel) và sử dụng mạng nơ ron tích chập 2D để hồi quy các tham số của vật thể từ những điểm neo này. Cách tiếp cận được mô tả trong hình 2.12. Cụ thể hơn, tại mỗi điểm ảnh  $[u, v]$  trên mặt phẳng hình ảnh, một tập các điểm neo 3D  $[w^a, h^a, l^a, \theta^a]_{3D}$ , điểm neo 2D  $[w^a, h^a]_{2D}$  và điểm neo độ sâu  $d^a$  được định nghĩa trước. Hình ảnh được đưa vào mạng tích chập để dự đoán độ dời (offsets) của các thông tin trong hộp bao quanh 2D  $\delta_{2D} = [\delta_x, \delta_y, \delta_w, \delta_h]_{2D}$  và độ dời của các thông tin trong hộp bao quanh 3D  $\delta_{3D} = [\delta_x, \delta_y, \delta_d, \delta_w, \delta_h, \delta_l, \delta_\theta]_{3D}$ . Sau đó lần lượt các thông tin của hộp bao quanh 2D  $b_{2D} = [x, y, w, h]_{2D}$  được tính toán như sau:

$$\begin{aligned}[x, y]_{2D} &= [u, v] + [\delta_x, \delta_y]_{2D} \odot [w^a, h^a]_{2D}, \\ [w, h]_{2D} &= [e^{\delta_w}_{2D}, e^{\delta_h}_{2D}] \odot [w^a, h^a]_{2D},\end{aligned}\quad (2.15)$$

và hộp bao quanh 3D  $b_{3D} = [x, y, z, l, w, h, \theta]_{3D}$  có thể được suy ra từ các điểm neo và  $\delta_{3D}$ :

$$\begin{aligned}[u^c, v^c] &= [u, v] + [\delta_x, \delta_y]_{3D} \odot [w^a, h^a]_{2D}, \\ [w, h, l]_{3D} &= [e^{\delta_w}_{3D}, e^{\delta_h}_{3D}, e^{\delta_l}_{3D}] \odot [w^a, h^a, l^a]_{3D}, \\ d^c &= d^a + \delta_{d3D}, \\ \theta_{3D} &= \theta^a_{3D} + \delta_{\theta_{3D}},\end{aligned}\quad (2.16)$$

trong đó  $[u^c, v^c]$  là tâm vật thể được chiếu trên mặt phẳng hình ảnh. Cuối cùng, hình chiếu của tâm và độ sâu của nó được chuyển thành tâm 3D  $[x, y, z]_{3D}$  của vật thể:

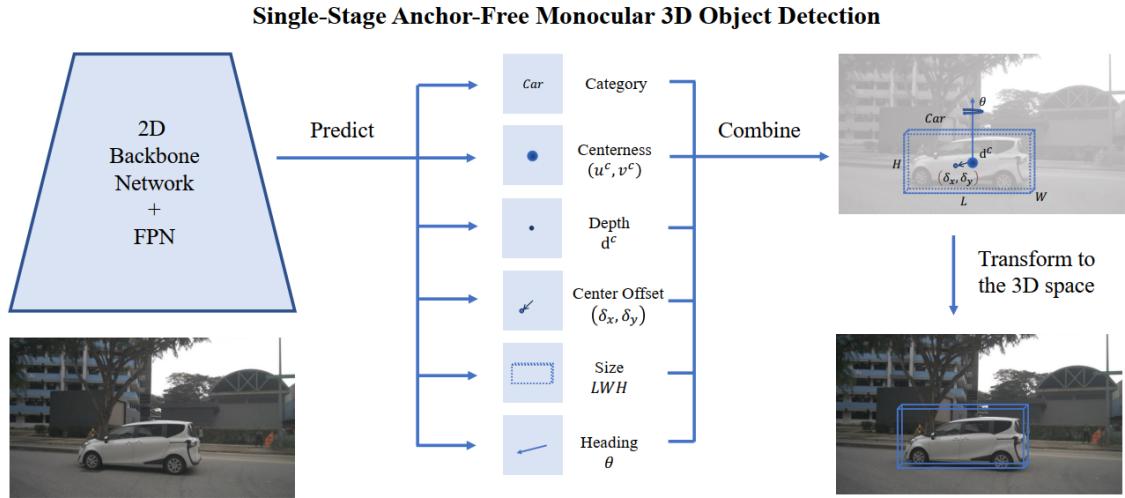
$$d^c \cdot \begin{bmatrix} u^c \\ v^c \\ 1 \end{bmatrix} = KT \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}_{3D}, \quad (2.17)$$

trong đó K và T là ma trận tham số nội và ngoại của máy ảnh, được dùng để chuyển đổi giữa các hệ tọa độ. M3D-RPN [2] là bài báo đầu tiên đề xuất bộ khung phương pháp dựa trên điểm neo này, và sau đó nhiều bài báo nghiên cứu khác đã cố cải thiện bộ khung phương pháp này như mở rộng thành phiên bản phát hiện 3D dựa trên video [3], giới thiệu một hàm non-maximum suppression (Úc chế không tối đa, NMS) mới, thiết kế mô đun Chú ý bất đối xứng (asymmetric attention).

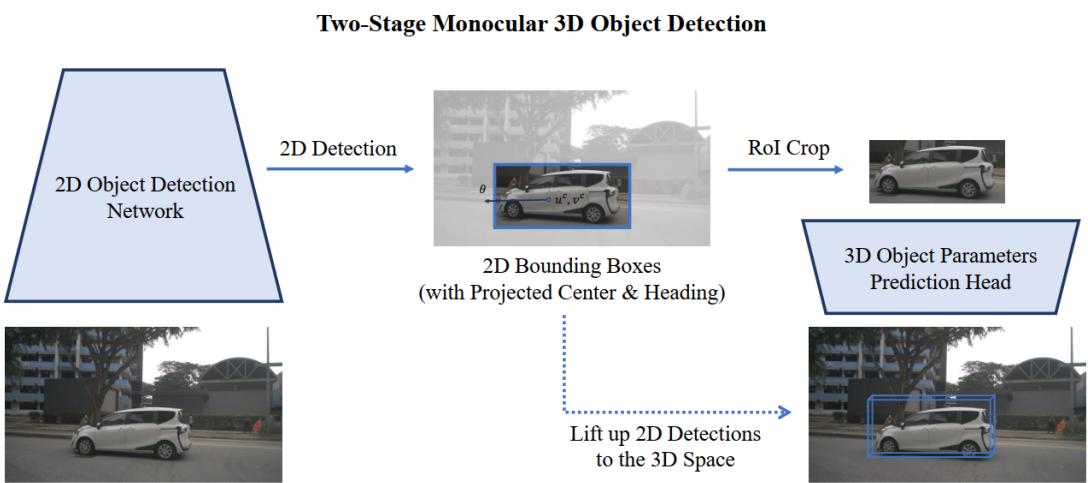
### 2.8.1.2 Phương pháp điểm neo tự do

Cách tiếp cận điểm neo tự do có thể dự đoán đặc điểm của vật thể 3D từ hình ảnh mà không cần sự trợ giúp của điểm neo, được mô tả như hình 2.13. Cụ thể hơn, hình ảnh được đưa vào một mạng nơ ron tích chập 2 chiều và sau đó nhiều đầu ra được áp dụng sau đó để dự đoán các thuộc tính của vật thể một cách riêng biệt. Các đầu ra dự đoán này thông thường sẽ bao gồm một đầu ra phân loại để dự đoán vật thể là đối tượng nào, một đầu ra để dự đoán tâm vật thể  $[u, v]$ , một đầu ra về độ dời (offset) để dự đoán độ dời của tâm  $[\delta_x, \delta_y]$  dựa trên  $[u, v]$ , một đầu ra để dự đoán độ dời của độ sâu  $\delta_d$ , một đầu ra để dự đoán kích thước của vật thể  $[w; h; l]$ , và một đầu ra về hướng để dự đoán góc độ quan sát  $\alpha$  của vật thể. Hình chiếu của tâm  $[u^c, v^c]$  và độ sâu  $d^c$  sau đó được tính toán như sau:

$$\begin{aligned}d^c &= \sigma^{-1} \left( \frac{1}{\delta_d + 1} \right) \\ u^c &= u + \delta_x \\ v^c &= v + \delta_y,\end{aligned}\quad (2.18)$$



Hình 2.13: Phương pháp điểm neo tự do. Nguồn ảnh :[40].



Hình 2.14: Phương pháp tiếp cận hai giai đoạn. Nguồn ảnh : [40].

trong đó  $\sigma$  là hàm sigmoid. Góc xoay  $\theta$  của vật thể có thể suy ra từ góc quan sát  $\alpha$  bằng công thức:

$$\theta = \alpha + \arctan \frac{x}{z}. \quad (2.19)$$

Cuối cùng, tâm 3D của vật thể  $[x, y, z]_{3D}$  được chuyển đổi từ hình chiếu của tâm  $[u^c, v^c]$  và độ sâu  $d^c$  như công thức 2.17. CenterNet được đề xuất năm 2019, là bộ khung (framework) đầu tiên của hướng tiếp cận này do X. Zhou và các đồng nghiệp đề xuất [69]. Có nhiều bài báo sau đó cũng sử dụng và cải thiện bộ khung này như ở những điểm như dùng cách ước tính độ sâu mới [31], [66] [59]; sử dụng kiến trúc tương tự FCOS (kiến trúc dùng để phát hiện vật thể 2D) [58]; hàm IoU mới [38]; điểm chính [25]; mối quan hệ không gian theo cặp [10]; dự đoán thông số nội của máy ảnh [70]; thay đổi góc nhìn [48] [46] [53].

### 2.8.1.3 Phương pháp tiếp cận hai giai đoạn

Cách tiếp cận phát hiện vật thể 3D hai giai đoạn thông thường sẽ được mở rộng từ kiến trúc phát hiện 2D hai giai đoạn, được mô tả như trong hình 2.14. Cụ thể hơn, người ta tận dụng một

công cụ dò 2D trong khâu đầu tiên để tạo ra các hộp bao quanh 2D từ hình ảnh đầu vào. Sau đó trong khâu thứ hai, hộp bao quanh 2D được nâng lên không gian 3D bằng cách dự đoán các thông số vật thể 3D từ 2D ROI (Region of Interest). ROI-10D [39] mở rộng kiến trúc của Faster RCNN [47] với một đầu ra mới để dự đoán thông số của vật thể 3D trong giai đoạn thứ 2. Những mô hình thiết kế tương tự cũng đã được sử dụng trong nhiều dự án với sự cải thiện như là giải quyết vấn đề của hàm mất mát khi phát hiện 2D và 3D[52], dự đoán góc độ hướng tới trong giai đoạn đầu tiên [23], học nhiều thông tin độ sâu chính xác hơn [45, 50, 33].

#### 2.8.1.4 Đánh giá tiềm năng và khó khăn

Phương pháp chỉ dùng hình ảnh hướng tới việc trực tiếp hồi quy thông số của hộp bao quanh 3D từ hình ảnh thông qua một kiến trúc phát hiện vật thể 2D đã được điều chỉnh để phù hợp cho việc phát hiện vật thể 3D. Bởi vì những phương pháp này lấy cảm hứng từ những phương pháp phát hiện vật thể 2D nên nó có thể hưởng lợi từ những tiên tiến của các phương pháp ấy và kiến trúc mạng dùng hình ảnh. Hầu hết những phương pháp có thể được huấn luyện khép kín (end-to-end) mà không cần phải trải qua tiền huấn luyện hoặc hậu xử lý, điều này khiến chúng trở nên khá đơn giản và hiệu quả.

Bởi vì vấn đề khôi phục thông tin 3D không thể giải quyết hoàn toàn nên một thách thức quan trọng của những phương pháp chỉ dùng hình ảnh đó là dự đoán độ sâu  $d^c$  một cách chính xác cho mỗi vật thể 3D. Công trình [59] đã cho thấy, chỉ cần thay đổi sâu dự đoán với độ sâu thực tế sẽ mang lại hơn 20% độ tăng chỉ số đánh giá Average Precision cho ở phân loại xe hơi ở trong tập dữ liệu KITTI 3D, trong khi thay thế các thông số khác chỉ dẫn đến sự tăng nhẹ ở chỉ số này. Quan sát này thể hiện rằng lỗi về độ sâu chiếm phần lớn trong tổng số lỗi và trở thành yếu tố quan trọng nhất cản trở việc phát hiện hình ảnh đơn một cách chính xác.

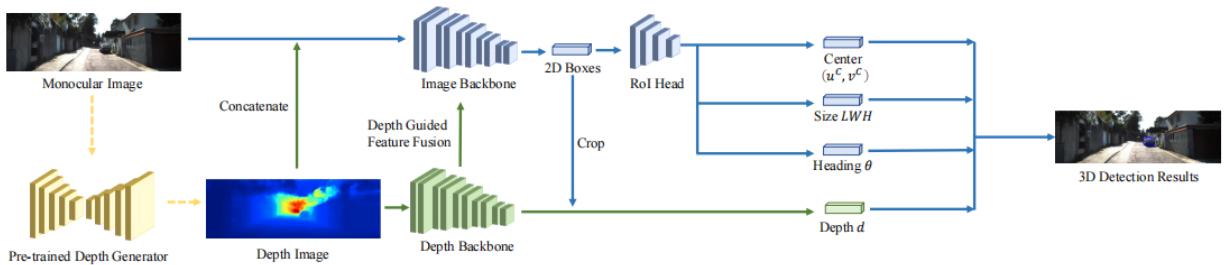
### 2.8.2 Phát hiện đối tượng 3D bằng hình ảnh đơn được hỗ trợ bởi công cụ ước tính độ sâu có sẵn

Hiệu suất của các phương pháp phát hiện đối tượng 3D dựa trên hình ảnh phụ thuộc rất nhiều vào khả năng ước tính khoảng cách chính xác của các đối tượng. Để đạt được kết quả phát hiện chính xác hơn, nhiều bài báo sử dụng cách huấn luyện trước một mạng ước tính độ sâu hỗ trợ. Cụ thể, hình ảnh đơn trước tiên được chuyển qua một công cụ ước tính độ sâu được huấn luyện trước, ví dụ: MonoDepth [14] hoặc DORN [12], để tạo ra một hình ảnh độ sâu. Sau đó, có hai loại phương pháp chính xử lý ảnh độ sâu và ảnh đơn: **phương pháp dựa trên hình ảnh độ sâu** và **phương pháp dựa trên Pseudo-LiDAR**.

#### 2.8.2.1 Phương pháp dựa trên hình ảnh độ sâu

Các phương pháp dựa trên hình ảnh độ sâu kết hợp hình ảnh và bản đồ độ sâu với mạng nơ-ron chuyên biệt để tạo ra các đặc trưng nhận biết chiều sâu có thể giúp nâng cao hiệu suất phát hiện. Các phương pháp tiếp cận dựa trên hình ảnh độ sâu thường tận dụng hai mạng backbone lặp lượt cho hình ảnh RGB và độ sâu. Từ đó, phương pháp này có được các đặc trưng nhận biết chiều sâu hình ảnh bằng cách kết hợp thông tin từ hai mạng backbone với các toán tử chuyên biệt.

Với các đặc trưng nhận biết độ sâu này, phương pháp có thể học được các hộp bao quanh 3D chính xác hơn và có thể được tinh chỉnh thêm với các hình ảnh độ sâu. MultiFusion [64] là công trình tiên phong giới thiệu bộ khung phát hiện dựa trên hình ảnh độ sâu. Các bài báo sau áp dụng các mô hình thiết kế tương tự với những cải tiến về kiến trúc mạng, toán tử, và chiến



Hình 2.15: Phương pháp dựa trên hình ảnh độ sâu. Nguồn ảnh : [40].

lược huấn luyện, ví dụ: phác hợp hướng dẫn theo chiều sâu D4LCN [11], truyền thông điệp có điều kiện độ sâu DDMP[56] ...

### 2.8.2.2 Phương pháp dựa trên Pseudo-LiDAR

#### Đám mây điểm

Các đối tượng có các vị trí khác nhau trong thế giới 3D có thể có cùng tọa độ trong mặt phẳng hình ảnh, điều này gây khó khăn cho mạng để ước tính kết quả cuối cùng.

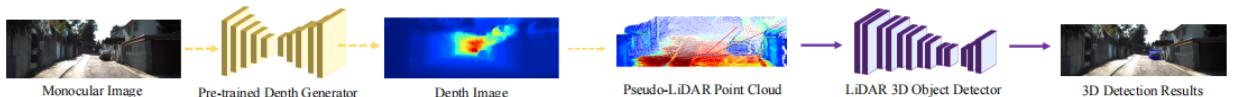
Các lợi ích của việc chuyển đổi bản đồ độ sâu thành đám mây điểm có thể được liệt kê như sau:

- Dữ liệu đám mây điểm hiển thị thông tin không gian một cách rõ ràng, giúp mạng dễ dàng học ánh xạ phi tuyến tính từ đầu vào đến đầu ra.
- Mạng có thể học được các tính năng phong phú hơn vì một số cấu trúc không gian cụ thể chỉ tồn tại trong không gian 3D.
- Tiên bộ đáng kể gần đây của học sâu trên các đám mây điểm cung cấp một viền gạch xây dựng vững chắc, chúng ta có thể ước tính kết quả phát hiện 3D theo cách hiệu quả và hiệu quả hơn.
- Sẽ hiệu quả hơn khi suy ra các hộp giới hạn 3D từ không gian cảnh 3D được tạo ra (tức là không gian X, Y, Z) so với mặt phẳng hình ảnh (tức là mặt phẳng hình ảnh R, G, B).

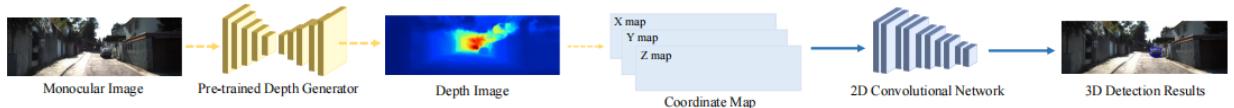
#### Pseudo-LiDAR

Tín hiệu 3D được lấy từ LiDAR chính xác nhưng tốn nhiều chi phí, trong khi các cách tiếp cận dựa trên hình ảnh tiết kiệm hơn nhưng độ ước tính độ sâu khá thấp, đây là động lực để phương pháp Pseudo-LiDAR ra đời. Các phương pháp dựa trên Pseudo-LiDAR biến đổi hình ảnh độ sâu thành một đám mây điểm LiDAR giả, và sau đó các công cụ phát hiện 3D dựa trên LiDAR có thể được sử dụng để phát hiện các hình ảnh 3D từ đám mây điểm. Tác giả của Pseudo-LiDAR lập luận rằng không phải chất lượng của dữ liệu mà là cách thể hiện của nó mới chiếm phần lớn sự khác biệt. Xem xét hoạt động bên trong của mạng nơ-ron tích tụ, tác giả đề xuất chuyển đổi bản đồ độ sâu dựa trên hình ảnh thành biểu diễn Pseudo-LiDAR, về cơ bản là bắt chước tín hiệu LiDAR, nó là một đám mây điểm 3D nhưng được trích xuất từ hình ảnh độ sâu thay vì cảm biến LiDAR thực. Với cách biểu diễn này, chúng ta có thể áp dụng các thuật toán phát hiện dựa trên LiDAR hiện có khác nhau. Nhiều bài báo đã cải thiện khuôn khổ phát hiện Pseudo-LiDAR, bao gồm tăng cường đám mây điểm giả với thông tin màu AM3D [37], giới thiệu phân đoạn phiên bản [62], thiết kế lược đồ biến đổi tọa độ tiến bộ [57], cải thiện ước tính độ sâu pixel với dự đoán tiền cảnh và hậu cảnh riêng biệt [60] và thiết kế cảm biến vật lý mới [6].

PatchNet [35] thách thức ý tưởng thông thường về việc tận dụng biểu diễn Pseudo-LiDAR  $P \in R^{HW \times 3}$  để phát hiện đối tượng 3D bằng hình ảnh đơn. Họ cho rằng khả năng của biểu



Hình 2.16: Phát hiện đối tượng 3D bằng hình ảnh đơn dựa trên Pseudo-LiDAR. Nguồn ảnh : [40].



Hình 2.17: Mô hình mạng PatchNet. Nguồn ảnh : [40].

biểu diễn Pseudo-LiDAR đến từ phép biến đổi tọa độ thay vì biểu diễn đám mây điểm. Do đó, bản đồ tọa độ  $M \in R^{H \times W \times 3}$  trong đó mỗi điểm ảnh mã hóa một tọa độ 3D có thể đạt được kết quả phát hiện có thể so sánh được với biểu diễn đám mây điểm Pseudo-LiDAR. Quan sát này cho phép họ áp dụng trực tiếp mạng nơ ron 2D trên bản đồ tọa độ để dự đoán các đối tượng 3D trên các đám mây điểm, loại bỏ nhu cầu sử dụng các phương pháp phát hiện dựa trên LiDAR vì tồn thời gian.

### 2.8.2.3 Đánh giá tiềm năng và khó khăn

Các phương pháp sử dụng thêm sự hỗ trợ về độ sâu ước tính độ sâu chính xác hơn bằng cách tận dụng mạng dự đoán độ sâu được huấn luyện trước. Cả biểu diễn hình ảnh độ sâu và Pseudo-LiDAR đều có thể tăng đáng kể hiệu suất phát hiện vật thể 3D trong hình ảnh đơn. Tuy nhiên, so với các phương pháp dùng hình ảnh chỉ yêu cầu chú thích về hộp 3D, thì việc huấn luyện trước mạng dự đoán độ sâu yêu cầu thêm chú thích độ sâu theo điểm ảnh, điều này tốn kém và cần trả quá trình huấn luyện từ đầu đến cuối của toàn bộ khung. Hơn nữa, các mạng ước tính độ sâu được huấn luyện trước có khả năng khai quật hóa kém. Vẫn còn một khoảng cách miền không đáng kể giữa nguồn được tận dụng chính để huấn luyện trước chuyên sâu và miền mục tiêu để phát hiện bằng ảnh đơn. Với thực tế là các tình huống lái xe thường rất đa dạng và phức tạp, các mạng chuyên sâu huấn luyện trước một miền bị hạn chế có thể không hoạt động tốt trong các ứng dụng trong thế giới thực.

# Chương 3

## Tập dữ liệu và chỉ số đánh giá

### 3.1 Tập dữ liệu

Dữ liệu là một phần không thể thiếu trong các mô hình Học máy và sự khả dụng của một tập dữ liệu quy mô lớn là một điều rất cần thiết cho sự thành công của một kĩ thuật học sâu hướng dữ liệu. Về việc phát hiện vật thể 3D bằng cách chỉ dùng hình ảnh trong bối cảnh xe tự lái, các đặc điểm chính của các tập dữ liệu công khai hiện có được tổng hợp trong bảng 3.1. Trong số những tập dữ liệu được nêu, bộ KITTI 3D [13], nuScenes [4] và Waymo Open [54] là những tập dữ liệu thường được sử dụng nhất và đã giúp thúc đẩy sự phát triển đáng kể trong mảng Phát hiện 3D. Vì vậy ở những phần tiếp theo, chúng em xin được tập trung về phần thông tin của ba tập dữ liệu này.

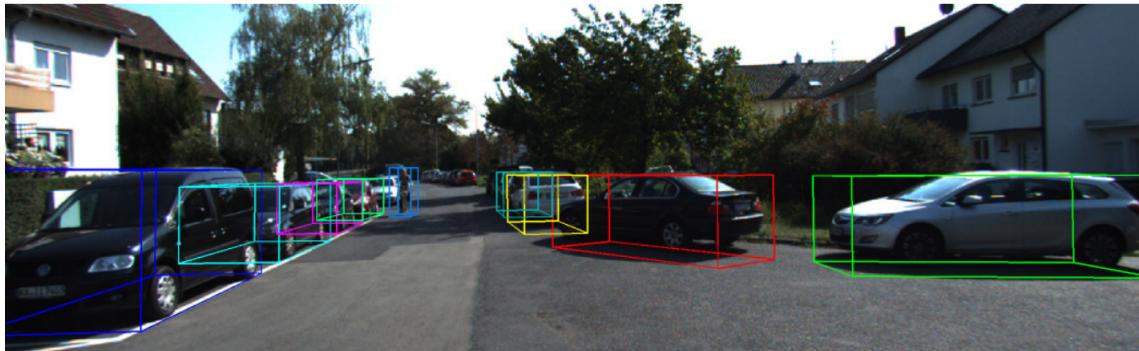
#### 3.1.1 KITTI 3D

KITTI 3D là bộ dữ liệu ra mắt vào năm 2012 và vì các bộ dữ liệu khác xuất hiện khá trễ (hầu hết là xuất hiện từ năm 2018) nên KITTI 3D được xem như là bộ dữ liệu duy nhất hỗ trợ cho sự phát triển của các công cụ phát hiện 3D chỉ dùng hình ảnh trong thập kỷ trước. KITTI 3D cung cấp những hình ảnh góc nhìn phía trước với độ phân giải  $1280 \times 384$  pixels. Về kích cỡ, KITTI 3D là tập dữ liệu nhỏ nhất trong ba tập dữ liệu phổ biến được nêu ở trên. KITTI 3D cung cấp 7481 hình ảnh huấn luyện và 7518 hình ảnh cho việc thử nghiệm. Hầu hết các bài báo chỉ sử dụng tập dữ liệu KITTI 3D với sự tập trung vào phân loại Xe hơi để đánh giá mô hình của mình, chỉ trừ một số công trình nghiên cứu đánh giá hiệu suất hoạt động dựa trên tập nuScenes

Tập dữ liệu	Năm	Kích thước				Độ đa dạng	
		Huấn luyện	Kiểm thử	Thử nghiệm	Hộp*	Lớp*	Đêm/Mưa
KITTI 3D[13]	2012	$7,418 \times 1$	-	$7,518 \times 1$	200K	8(3)	Không/Không
Argoverse[7]	2019	$39,384 \times 7$	$15,062 \times 7$	$12,507 \times 7$	993K	15	Có/Có
Lyft L5[20]	2019	$22,690 \times 6$	-	$27,468 \times 6$	1.3M	9	Không/Không
H3D[43]	2019	$8,873 \times 3$	$5,170 \times 3$	$13,678 \times 3$	1.1M	8	Không/Không
nuScenes[4]	2019	$28,130 \times 6$	$6,019 \times 6$	$6,008 \times 6$	1.4M	23(10)	Có/Có
Waymo Open[54]	2019	$122,200 \times 5$	$30,407 \times 5$	$40,077 \times 5$	12M	4(3)	Có/Có

Bảng 3.1: Một số tập dữ liệu được dùng trong Phát hiện vật thể 3D bằng hình ảnh trong ngữ cảnh xe tự hành. \*: K viết tắt cho nghìn, M viết tắt cho triệu. \*: trong ngoặc là số lượng lớp được dùng trong tiêu chuẩn đánh giá.

hoặc Waymo Open. Tuy nhiên, như đã nhắc ở trên, trong tương lai, việc đánh giá mô hình trong những tập dữ liệu quy mô lớn là rất cần thiết để đánh giá được độ hiệu quả của thuật toán. Dữ liệu của tập KITTI 3D được ghi lại ở Đức vào ban ngày trong điều kiện thời tiết tốt và chủ yếu đánh giá các vật thể từ ba phân loại chính gồm: Xe hơi, người đi bộ và người đi xe đạp (Car, Pedestrian, Cyclist). Các phân loại này được chia thành ba mức độ khó dựa trên độ cao của hình hộp 2D bao quanh, độ che lấp và mức độ bị cắt của vật thể. Tập KITTI chứa các thông tin tương ứng như nhãn và thông số máy ảnh tương ứng với hình ảnh đã chụp. Nhãn của tập KITTI gồm



Hình 3.1: Hình ảnh từ tập KITTI 3D. Nguồn ảnh : [13].

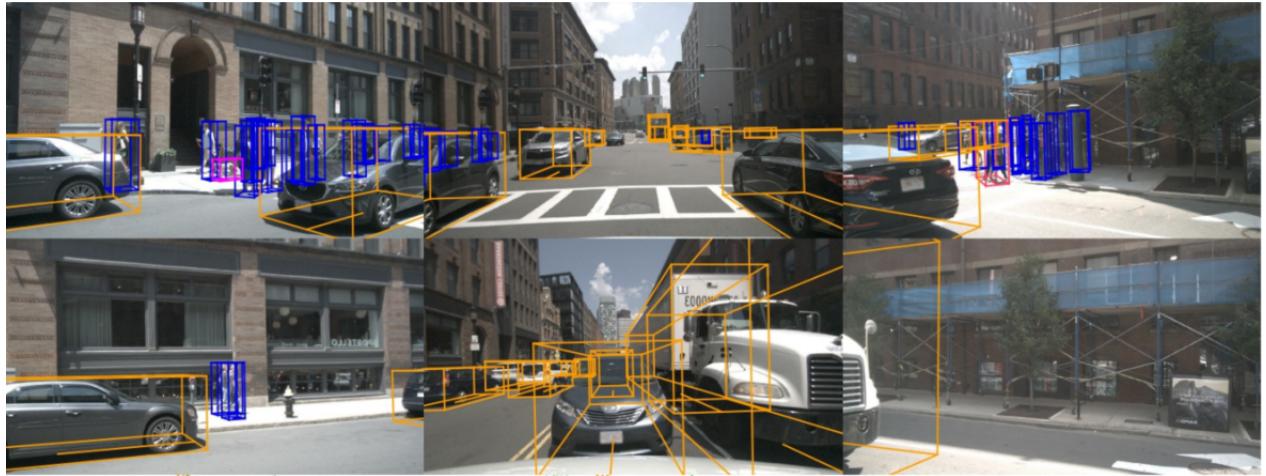
các thông tin như sau:

- Lớp: mô tả lớp của đối tượng, gồm ba lớp chính trong tiêu chuẩn đánh giá là *Xe hơi*, *Người đi xe đạp*, *Người đi bộ* và một số lớp khác. Ngoài ra, các đối tượng ở quá xa hoặc bị che khuất nhiều trong khung hình cũng được đánh dấu với nhãn riêng là 'DontCare'.
- Mức độ cắt: Giá trị trong đoạn  $[0,1]$ , thể hiện mức độ bị cắt của đối tượng khi nằm ở rìa bức ảnh. Các đối tượng có nhãn là 'DontCare' mang giá trị là -1.
- Độ che khuất: thể hiện mức độ bị che khuất với các giá trị nguyên là 0 (nhìn thấy hoàn toàn), 1 (bị che khuất một phần), 2 (bị che khuất phần lớn), 4 (không xác định).
- Alpha: góc quan sát của đối tượng, có giá trị  $[-\pi, \pi]$ .
- Hộp bao quanh 2D: gồm tọa độ của 4 điểm của hộp bao quanh 2D trong hệ tọa độ điểm ảnh.
- Kích thước: kích thước 3D của đối tượng, gồm chiều dài, chiều rộng và chiều cao.
- Vị trí 3D: vị trí 3D x,y,z của đối tượng trong tọa độ hệ tọa độ máy ảnh.
- Góc quay: góc quay quanh trục Y trong hệ tọa độ máy ảnh, giá trị  $[-\pi, \pi]$
- Điểm: thể hiện độ tin cậy trong dự đoán, càng cao thì đối tượng càng dễ dự đoán.

### 3.1.2 nuScenes

Xuất hiện vào năm 2019, sau bộ KITTI 3D, tuy nhiên nuScenes lại trở thành một trong những tập dữ liệu được sử dụng phổ biến nhất. Trong tập nuScenes, sáu camera được sử dụng để tạo ra góc nhìn 360 độ với độ phân giải 1600 x 900 pixels. Với khoảng 40 nghìn khung hình, nuScene có quy mô lớn hơn nhiều so với tập KITTI 3D với mỗi khung hình là một ảnh toàn cảnh được tạo ra bởi nhiều camera. Cụ thể hơn, nuScenes cung cấp 28 130 khung hình cho việc huấn luyện, 6 019 và 6 008 khung hình lần lượt cho việc kiểm thử và thử nghiệm (6 hình ảnh cho mỗi khung hình). Tập nuScenes gồm 1000 phân đoạn, mỗi phân đoạn gồm 20 giây được ghi lại ở thành phố Boston và Singapore. Khác với KITTI 3D, những phân đoạn này được ghi lại ở các thời điểm khác nhau của ngày (bao gồm cả ban đêm) và trong các điều kiện thời tiết khác nhau (bao gồm

cả mưa). Với nuScenes, có mười loại vật thể chính cho việc Phát hiện 3D. Đồng thời tập dữ liệu này cũng chú thích nhãn về đặc điểm cho từng loại, ví dụ như xe hơi đang di chuyển hay đang đậu lại hoặc xe đạp có người lái hay không. Những đặc điểm ấy có thể được xem như nhãn của một lớp chi tiết (fine-grained class labels) và độ chính xác của việc nhận diện đặc điểm cũng được xem xét trong tiêu chuẩn của nuScenes.



Hình 3.2: Hình ảnh từ 6 camera tạo nên khung hình toàn cảnh trong tập nuScenes. Nguồn ảnh : [4].

### 3.1.3 Waymo Open

Tương tự nuScenes, Waymo Open cũng xuất hiện vào năm 2019 và trở thành bộ dữ liệu giúp ích rất nhiều cho sự phát triển của mảng Phát hiện 3D. Tập Waymo Open cũng ghi lại góc nhìn 360 độ bằng cách sử dụng năm camera đồng bộ với độ phân giải là 1920 x 1280 pixels. Sử dụng nhiều camera để tạo ra ảnh toàn cảnh như nuScenes nhưng Waymo Open lại có kích thước lên đến khoảng 200 nghìn khung hình. Trong đó gồm 122,200 khung hình cho việc huấn luyện, 30,407 khung hình cho việc kiểm thử và 40,077 khung hình cho việc thử nghiệm với năm hình ảnh ở mỗi khung hình. Điều này khiến Waymo Open trở thành tập dữ liệu có quy mô lớn nhất trong mảng Phát hiện 3D. Tập Waymo Open chứa 1,150 phân đoạn, được chụp ở các thành phố ở Hoa Kỳ như Phoenix, Mountain View và San Francisco dưới nhiều điều kiện thời tiết khác nhau, bao gồm ban đêm và những ngày mưa. Tương tự như KITTI 3D, Waymo Open cũng định nghĩa hai mức độ khó cho việc Phát hiện 3D dựa vào số lượng điểm LiDAR chứa trong mỗi hộp 3D bao quanh. Các vật thể được quan tâm chính trong bộ dữ liệu này bao gồm các phương tiện giao thông, người đi bộ và người đạp xe đạp.

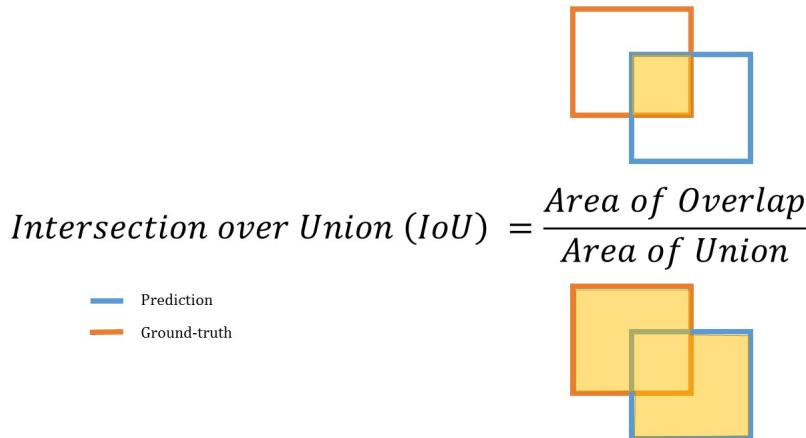
## 3.2 Chỉ số đánh giá

Tương tự như trong mảng Phát hiện vật thể 2D, Average Precision (Độ chính xác trung bình, AP) là chỉ số chính dùng để đánh giá được sử dụng trong mảng Phát hiện vật thể 3D. Từ định nghĩa ban đầu, mỗi tập dữ liệu đều áp dụng những sự điều chỉnh riêng tạo nên những chỉ số đánh giá cụ thể của từng tập. Độ chính xác trung bình nguyên bản sẽ được nói ở phần 3.2.1, còn các biến thể khác được sử dụng trong các tập dữ liệu như KITTI 3D, nuScenes và Waymo Open sẽ được miêu tả trong phần 3.2.2

### 3.2.1 Chỉ số chính xác trung bình

Để tính toán được chỉ số chính xác trung bình, các dự đoán trước hết được so sánh với các kết quả thực tế tương ứng theo một thước đo riêng. Điều này sẽ giúp đánh giá được độ tốt khi biết được các dự đoán có tốt đú hay không. Thước đo được sử dụng nhiều nhất trong mảng Phát hiện vật thể 3D chính là Intersection over Union (IoU)(hình 3.3), được định nghĩa như sau, giữa kết quả thực tế A và hộp bao quanh 3D B:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3.1)$$



Hình 3.3: Hình minh họa cho công thức IoU.

Thước đo IoU sẽ được sử dụng để quyết định một đánh giá là True Positive hay False Positive bằng cách so sánh nó với một ngưỡng nhất định. Giải thích của một số thuật ngữ được sử dụng như sau:

- *True Positive*: Mô hình dự đoán kết quả cho IoU vượt ngưỡng (hộp bao quanh chính xác) và phân loại đúng.
- *False Positive*: Khi mô hình đưa ra dự đoán cho IoU không vượt ngưỡng quy định bất kể class được dự đoán là gì.
- *False Negatives*: Khi thực tế là có vật thể nhưng mô hình lại không đưa ra dự đoán.
- *Precision*: Độ đo chính xác cho những dự đoán của mô hình đưa ra, thể hiện những dự đoán được đưa ra tốt như thế nào.

$$Precision = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}} \quad (3.2)$$

- *Recall*: Độ đo chính xác của kết quả mô hình đưa ra so với tổng thể bức ảnh, thể hiện mô hình dự đoán những vật thể trong hình tốt như thế nào.

$$Recall = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}} \quad (3.3)$$

Precision có thể được coi như một hàm với tham số là recall  $p(r)$ . Để giảm sự nhập nhằng trong đường cong precision-recall, giá trị interpolated precision (giá trị chính xác nội suy) sẽ được

dùng để tính Average Precision (Độ chính xác trung bình):

$$AveragePrecision = \frac{1}{|\mathbb{R}|} \sum_{r \in \mathbb{R}} p_{interp}(r) \quad (3.4)$$

trong đó  $\mathbb{R}$  là tập các vị trí recall đã được xác định trước và  $p_{interp}$  là hàm nội suy được định nghĩa như sau:

$$p_{interp}(r) = \max_{r': r' \geq r} p(r') \quad (3.5)$$

nghĩa là thay vì lấy trung bình của các giá trị precision quan sát tại điểm recall  $r$ , giá trị precision lớn nhất tại giá trị recall lớn hơn hoặc bằng  $r$  sẽ được lấy thay thế.

### 3.2.2 Chỉ số đánh giá của các tập dữ liệu

#### 3.2.2.1 KITTI 3D

Tập KITTI 3D sử dụng Average Precision là chỉ số đánh giá chính và có thực hiện một vài sửa đổi. Sửa đổi đầu tiên là tính toán của IoU sẽ được thực hiện trong không gian 3D. Bên cạnh đó, KITTI 3D sử dụng đề xuất của Simonelli và các đồng nghiệp [52] và thay thế  $\mathbb{R}_{11} = \{0, 1/10, 2/10, 3/10, \dots, 9/10, 1\}$  trong 3.4 với  $\mathbb{R}_{40} = \{1/40, 2/40, 3/40, \dots, 1\}$ , nghĩa là lấy mẫu với tần suất dày đặc hơn và bỏ đi giá trị recall tại 0.

Không những thế, KITTI 3D cũng đồng thời đề xuất một chỉ số đánh giá mới, Average Orientation Similarity (Độ tương tự định hướng trung bình, AOS), để đánh giá độ chính xác của việc ước tính hướng của vật thể. Công thức của AOS như sau:

$$AOS = \frac{1}{R} \sum_{r \in R} \max_{r': r' \geq r} s(r') \quad (3.6)$$

Độ tương tự định hướng  $s(r) \in [0, 1]$  với recall  $r$  là một biến thể được chuẩn hóa của Độ tương tự cosine:

$$s(r) = \frac{1}{|D(r)|} \sum_{i \in D(r)} \frac{1 + \cos \Delta_\theta^{(i)}}{2} \delta_i, \quad (3.7)$$

trong đó  $D(r)$  thể hiện cho tập hợp các kết quả dự đoán vật thể tại tỉ lệ recall  $r$  và  $\Delta_\theta^{(i)}$  là sự khác biệt về góc giữa hướng được dự đoán và hướng thực tế trong dự đoán  $i$ . Để xử phạt việc đưa ra nhiều dự đoán cho một vật thể, KITTI 3D buộc  $\delta_i = 1$  nếu dự đoán  $i$  đã được gán cho một hộp bao quanh thực tế và  $\delta_i = 0$  nếu nó chưa được gán. Lưu ý rằng tất cả các chỉ số AP được tính toán độc lập cho từng độ khó và loại vật thể.

#### 3.2.2.2 Waymo Open

Waymo Open cũng sử dụng chỉ số AP với sự điều chỉnh nhỏ: thay thế  $\mathbb{R}_{11}$  trong 3.4 với  $\mathbb{R}_{21} = \{0, 1/20, 2/20, \dots, 1\}$ . Tiếp đến, họ xem xét và thấy rằng việc dự đoán góc hướng đến của vật thể một cách chính xác rất quan trọng trong mảng xe tự lái và chỉ số AP thì không có khái niệm về góc hướng đến nên đã đề xuất Average Precision weighted by Heading (Average Precision có trọng số góc hướng đến, APH) là chỉ số đánh giá chính của tập. Cụ thể hơn, APH kết hợp thông tin của góc hướng đến vào trong tính toán precision. Mỗi True Positive có trọng số bằng độ chính xác của góc hướng tới được định nghĩa bằng  $\min(|\theta - \theta^*|, 2\pi - |\theta - \theta^*|)/\pi$ , trong đó  $\theta$  và  $\theta^*$  lần lượt là góc hướng tới được dự đoán và góc hướng tới thực tế với đơn vị là radians trong đoạn  $[-\pi, \pi]$ . Lưu ý rằng APH đánh giá cả hiệu suất của Phát hiện 3D lẫn việc ước tính hướng, trong khi AOS ở 3.6 chỉ được thiết kế để dành cho việc ước tính hướng.

### 3.2.2.3 nuScenes

Tập nuScenes đề xuất một chỉ số AP mới. Cụ thể hơn, nó sử dụng khoảng cách tâm 2D trên mặt đất để xem xét việc trùng khớp giữa dự đoán và thực tế với một ngưỡng khoảng cách nhất định  $d$  (ví dụ 2m), thay vì sử dụng chỉ số IoU ở 3.1. Bên cạnh đó, nuScenes tính toán AP dưới dạng vùng chuẩn hóa dưới đường cong precision-curve với recall và precision trên 10%. Sau cùng, nuScenes tính mean Average Precision (Trung bình độ chính xác trung bình, mAP) qua từng ngưỡng trùng khớp  $\mathbb{D} = \{0.5, 1, 2, 4\}$  mét và tập hợp các lớp  $\mathbb{C}$ :

$$mAP = \frac{1}{|\mathbb{C}| |\mathbb{D}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} AP_{c,d} \quad (3.8)$$

Tuy nhiên, chỉ số này chỉ xem xét đến vấn đề xác định vật thể, bỏ qua hiệu ứng của các khía cạnh khác như chiều và hướng. Để khắc phục việc này, nuScenes đã đề xuất một tập các chỉ số True Positive được thiết kế để đo lường các sai số dự đoán một cách riêng biệt bằng cách sử dụng tất cả True Positive (được xác định theo khoảng cách tâm  $d = 2m$  trong khi xét trùng khớp). Tất cả 5 chỉ số True Positive được thiết kế là đơn vị dương, được xác định như sau:

- Average Translation Error (Lỗi chuyển đổi trung bình): là khoảng cách Euclid cho tâm của vật thể trên mặt phẳng 2D (đơn vị: mét).
- Average Scale Error (Lỗi quy mô trung bình): là lỗi 3D IoU ( $1 - IoU$ ) sau khi căn chỉnh hướng và chuyển đổi.
- Average Orientation Error (Lỗi định hướng trung bình): là khác biệt về góc xoay nhỏ nhất giữa dự đoán và thực tế (đơn vị: radians).
- Average Velocity Error (Lỗi vận tốc trung bình): là sai số vận tốc tuyệt đối dưới dạng chuẩn L2 của sự khác biệt vận tốc trong 2D. (m/s)
- Average Attribute Error (Lỗi đặc điểm trung bình): được định nghĩa là 1 trừ độ chính xác phân loại thuộc tính ( $1 - acc$ ).

# Chương 4

## Các nghiên cứu liên quan

### 4.1 Mô hình đề xuất

Mục tiêu lựa chọn mô hình cơ sở của nhóm là lựa chọn một thiết kế không quá lỗi thời, độ chính xác tương đối, độ trễ không cao nhằm phù hợp với xu hướng thực tế, góp phần cho việc phát triển những hướng đi có kết quả hiện đại, tốt nhất. Những điều kiện này sẽ làm một nền tảng tốt để thành quả nghiên cứu có thể đáp ứng được mục tiêu đề ra ban đầu của nhóm. Ở mảng Phát hiện vật thể 3D trong hình ảnh đơn, thời gian gần đây liên tục có nhiều bài báo được đề xuất để giải quyết bài toán này và các chỉ số đo cũng liên tục được cải thiện được so với trước đây. Dưới đây là một vài bài báo nghiên cứu được nhóm chọn lọc và cho là phù hợp với mục tiêu đặt ra.

#### 4.1.1 Objects are Different: Flexible Monocular 3D Object Detection

MonoFlex[66] được đề xuất vào năm 2021 bởi Yunpeng Zhang và các cộng sự . Hầu hết các phương pháp trước MonoFlex đều sử dụng chung một cách tiếp cận cho tất cả các vật thể mặc dù phân phối thông tin của chúng rất đa dạng. Điều này dẫn đến việc bỏ qua sự khác nhau giữa các vật thể khi chỉ xem xét phương sai quy mô chung, làm cho mô hình phải chịu ảnh hưởng bởi việc dự đoán những vật thể nằm ngoài phân phối ấy và kết quả là hiệu suất mô hình sẽ bị giảm. Những vật thể bị che khuất, bị cắt cũng bị ảnh hưởng bởi sự giới hạn này khi chúng rất khó để phát hiện, tuy nhiên loại vật thể này lại rất quan trọng đối với sự an toàn của xe tự hành. Để giải quyết vấn đề này, tác giả đã đề xuất một framework (bộ khung) hoàn toàn tách các vật thể bị cắt thành một phần riêng để xử lý và kết hợp nhiều cách tiếp cận cho việc ước tính độ sâu của vật thể. Các đề xuất ấy được xem xét về việc xây dựng tính linh hoạt lần lượt từ hai khía cạnh là hình chiếu của tâm 3D và độ sâu của vật thể.

##### Tách các biểu diễn của vật thể

Để xác định được hình chiếu của tâm 3D, tác giả đề xuất chia vật thể thành hai kiểu là bên trong và bên ngoài dựa trên việc hình chiếu của tâm nằm trong hay nằm ngoài tâm ảnh. Sau đó, họ biểu diễn những vật thể thuộc loại bên trong bằng chính hình chiếu của tâm, còn vật thể thuộc loại bên ngoài được biểu diễn bằng cách lựa chọn những điểm ngay rìa của bức ảnh một cách hợp lí để cả hai nhóm vật thể đều được xử lý lần lượt bởi vùng trong và ngoài của bản đồ đặc trưng (feature map).

##### Độ sâu của vật thể

Để ước tính độ sâu của vật thể, tác giả đề xuất việc kết hợp nhiều bộ ước lượng độ sâu khác nhau với ước lượng không chắc chắn (uncertainty estimation). Việc mô hình sự không chắc chắn của các độ sâu ước tính từ nhiều bộ ước lượng sẽ được dùng để định lượng sự đóng góp của



Hình 4.1: MonoFlex giải quyết vấn đề dự đoán những vật thể nằm ngoài phân phối. Ảnh trên: các phương pháp trước đó. Ảnh dưới: MonoFlex. Nguồn ảnh :[66].

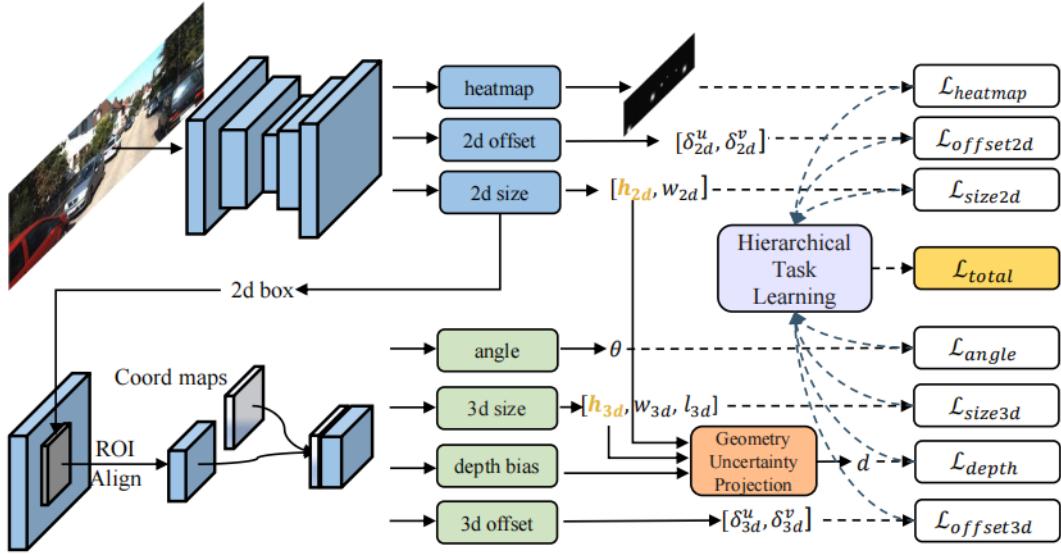
chúng vào kết quả dự đoán kết hợp cuối cùng. Bộ ước lượng sẽ gồm hồi quy trực tiếp và giải pháp hình học từ những keypoint (điểm chính). Tác giả thấy rằng việc tính toán độ sâu từ những keypoint thông thường sẽ là một vấn đề có nhiều ràng buộc, trong khi lấy trung bình của kết quả từ nhiều keypoint có thể nhạy cảm với sự bị che khuất của các keypoint. Vì thế, MonoFlex chia các keypoint thành  $M$  nhóm, mỗi nhóm sẽ đủ để giải quyết độ sâu. Để kết hợp  $M$  bộ ước lượng dựa trên keypoint, tác giả mô hình hóa sự không chắc chắn của chúng và hình thành kết quả ước lượng cuối cùng dưới dạng trung bình sự không chắc chắn có trọng số (uncertainty-weighted average). Để xuất kết hợp này cho phép mô hình chọn lựa linh hoạt các bộ ước lượng phù hợp hơn để dự đoán tốt và chính xác.

#### 4.1.2 Geometry Uncertainty Projection Network for Monocular 3D Object Detection

Công trình nghiên cứu thuộc về Yan Lu và các cộng sự, được đề xuất năm 2021. [33]. GUPNet là một công trình thuộc kiểu phương pháp hai giai đoạn và được phát triển theo hướng tận dụng phép chiếu hình học để giúp cho việc suy luận độ sâu. Độ sâu của vật thể sẽ không được trực tiếp hồi quy từ đầu ra của mạng nơ ron nhân tạo mà sẽ được tính thông qua mối quan hệ về giữa chiều cao và độ sâu của vật thể. Cụ thể hơn, sau khi chúng ta có chiều cao 2D của vật thể trong mặt phẳng hình ảnh, chiều cao 3D của vật thể trong không gian 3D, ta có thể suy ra được độ sâu của vật thể  $z$  bằng công thức:

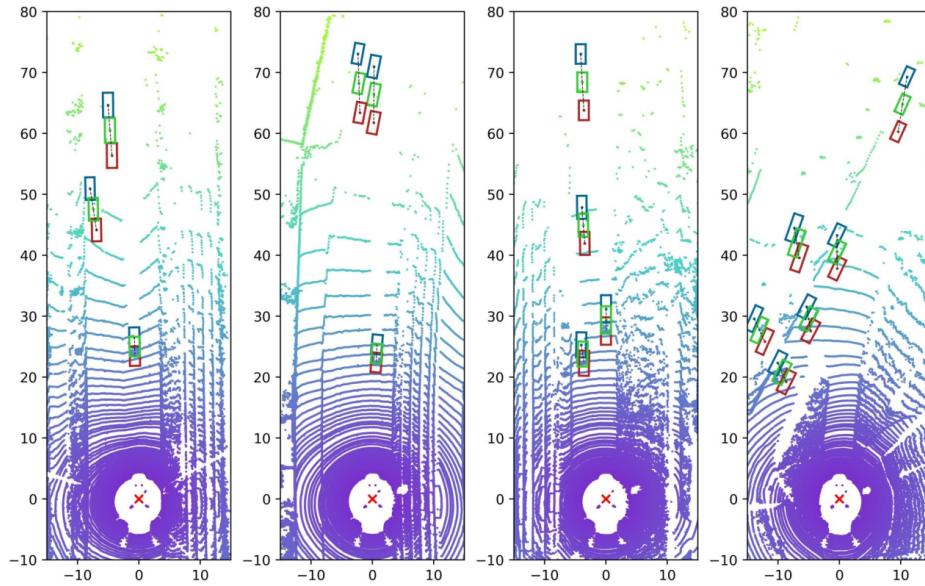
$$z = \frac{f \cdot h_{2D}}{h_{3D}} \quad (4.1)$$

với  $f$  là tiêu cự của máy ảnh. GUPNet đã đưa ra rằng việc suy luận độ sâu phụ thuộc vào chiều cao sẽ bị ảnh hưởng rất nhiều nếu việc ước tính chiều cao không chính xác, khi ấy lỗi của độ sâu cũng sẽ bị khuếch đại lên, được minh họa ở hình 4.3. Tuy nhiên, lỗi của việc ước tính độ cao là không thể tránh khỏi. Đối với độ cao 2D, do dựa trên nền tảng của các công cụ phát hiện vật



Hình 4.2: Thiết kế của mạng GUPNet. Nguồn ảnh: [33].

thể 2D có sự phát triển tiên tiến nên việc ước tính độ cao 2D mang lại độ chính xác tương đối cao. Lỗi trong việc ước tính chiều cao chủ yếu nằm ở độ cao 3D khi các thông tin 3D không thể được ước tính chính xác khi chỉ dùng ảnh 2D. Việc khuếch đại lỗi phía trên làm cho đầu ra của các phương pháp dựa trên phép chiếu này trở nên khó kiểm soát, đồng thời ảnh hưởng nghiêm trọng đến cả độ tin cậy của khâu suy luận và độ hiệu quả của khâu huấn luyện. Vì thế, bài báo cố gắng xử lý việc lỗi của độ sâu được suy ra từ lỗi của việc ước tính độ cao 3D.



Hình 4.3: Minh họa việc sai sót về độ cao dẫn đến sai sót của độ sâu bị khuếch đại. Hộp màu xanh lá là đầu ra gốc, còn hộp màu xanh dương và màu đỏ lần lượt được dời bằng cách +0.1m và -0.1m trong độ cao 3D. Nguồn ảnh : [33].

## 4.2 Các hướng phát triển mới

Sau khi đã lựa chọn một mô hình cơ sở phù hợp, việc tiếp theo là khảo sát các hướng phát triển mới có thể áp dụng. Một số vấn đề có thể khai thác như sau:

- Việc kết hợp với các công cụ ước tính độ sâu phụ trợ vẫn chưa được khai thác nhiều.
- Việc tăng cường dữ liệu đầu vào vẫn chưa được tận dụng nhiều bởi việc tăng cường dữ liệu trong khung phát hiện 3D bằng hình ảnh đơn hiện tại gây ra vi phạm các ràng buộc hình học.
- Việc huấn luyện vẫn còn độc lập, chưa tận dụng hết được các ràng buộc không gian.

Chúng ta sẽ đề cập đến việc giải quyết những vấn đề nêu trên ở các phần dưới.

### 4.2.1 Tích hợp mạng ước tính độ sâu phụ trợ

Bên cạnh sự phát triển của của phát hiện đối tượng chỉ bằng một hình ảnh thì các phương pháp phát hiện đối tượng với sự trợ giúp của các công cụ ước tính độ sâu có sẵn cũng phát triển không kém. Các phương pháp sử dụng ước tính độ sâu bằng công cụ có sẵn được chứng minh là đạt được hiệu suất tổng thể tốt hơn. Tuy nhiên hầu hết mạng hỗ trợ độ sâu và các phương pháp phát hiện vẫn hoàn toàn độc lập, riêng biệt. DD3D [42] đã đề xuất nhúng mạng phát hiện 3D và ước tính độ sâu thành một mạng đa tác vụ duy nhất. Họ đã chứng minh rằng, khi hai nhiệm vụ này được huấn luyện cùng nhau và mang lại lợi ích cho nhau, thì hiệu suất phát hiện 3D thậm chí còn tăng lên nhiều hơn.

### 4.2.2 Tăng cường dữ liệu đầu vào

Các phương pháp hiện tại không định vị được các đối tượng một cách nhất quán khi xảy ra các thay đổi hình học khác nhau. Việc phục hồi độ sâu dưới sự thay đổi về kích thước và vị trí rõ ràng của đối tượng không ổn định.

Tăng cường dữ liệu là một kỹ thuật hiệu quả để tăng cường hiệu suất phát hiện đối tượng. Cả dựa trên hình học (ví dụ: tỷ lệ ngẫu nhiên, cắt xén ngẫu nhiên, v.v.) và các kỹ thuật tăng cường dựa trên màu sắc (ví dụ: biến dạng màu) đã được áp dụng rộng rãi trong các mô hình phát hiện 2D. Mặc dù các phương pháp tăng cường dữ liệu tích cực này có mang lại mức tăng ấn tượng cho cả trường hợp 2D hoặc một số biểu diễn dữ liệu 3D cụ thể, tuy nhiên, chúng hầu như không được tận dụng trong các khung phát hiện 3D bằng hình ảnh đơn hiện tại do vi phạm các ràng buộc hình học, trong đó lật ảnh và biến dạng màu là hai phương pháp duy nhất được sử dụng trong lĩnh vực này trong một thời gian dài.

Để giải quyết vấn đề trên, [28] đề xuất bốn kỹ thuật tăng cường nhận biết hình học ở cấp độ hình ảnh và cấp độ phiên bản nhằm tạo thêm dữ liệu huấn luyện với việc bảo toàn các thuộc tính hình học:

- Sửa đổi một số phương pháp tăng cường dữ liệu thường được sử dụng cho hình ảnh 2D để chúng có thể duy trì tính nhất quán hình học trong không gian 3D.
- Đề xuất một phương pháp nhiều hình ảnh dành riêng cho 3D sử dụng chuyển động của máy ảnh.

Với dữ liệu huấn luyện đa dạng hơn, các phương pháp tăng cường mang lại sự cải thiện nhất quán so với các phương pháp hiện đại trên bộ dữ liệu KITTI và nuScenes. Ngoại trừ các nhiễu loạn hình ảnh đơn giản, các kỹ thuật tăng cường phức tạp đã xuất hiện trong phát hiện đối tượng 2D và hiểu cảnh 3D để cải thiện độ bền của mô hình (ví dụ: trộn, tổng hợp chế độ xem mới,

mô phỏng thành thực, ví dụ đối nghịch, v.v.). Mặt khác, tính năng phát hiện đối tượng 3D bằng hình ảnh đơn cũng có các vấn đề về độ bền của nó (ví dụ: nhiễu loạn của cao độ máy ảnh và góc cuộn, tắc v.v.), điều này có thể được giảm bớt bằng các phương pháp tăng cường dữ liệu tùy chỉnh.

#### 4.2.3 Giới thiệu thêm các ràng buộc mới trong quá trình huấn luyện

Hầu hết các phương pháp hiện tại coi mỗi đối tượng 3D trong cảnh là một mẫu huấn luyện độc lập, trong khi bỏ qua các mối quan hệ hình học vốn có của chúng, do đó chắc chắn dẫn đến việc thiếu các ràng buộc về không gian tận dụng. [15] đề xuất một phương pháp mới xem xét tất cả các đối tượng và khám phá các mối quan hệ lẫn nhau của chúng để giúp ước tính tốt hơn các hộp 3D.

Một hàm mất mát khả vi, được gọi **Homography Loss**, được đề xuất để đạt được mục tiêu, khai thác cả thông tin 2D và 3D, nhằm mục đích cân bằng các mối quan hệ vị trí giữa các đối tượng khác nhau bằng các ràng buộc toàn cục, để có được dự đoán chính xác hơn hộp 3D .

# Chương 5

## Mô hình cơ sở

Sau khi đánh giá kĩ càng và thử nghiệm sơ bộ, nhóm đã quyết định chọn mô hình **GUP-Net**[33] làm mô hình cơ sở của mình:

- Lý do chủ chốt nhất là qua quá trình mà nhóm khảo sát, việc GUPNet[33] được sử dụng làm thiết kế đề xuất để cải thiện không nhiều, ngược lại nghiên cứu như MonoFlex[66] thì lại xuất hiện khá nhiều trong các bài báo nghiên cứu cải thiện gần đây. Đây là một tiền đề khá tốt để chọn GUPNet vì sẽ không trùng lặp, xung đột với các nghiên cứu đã có, đồng thời việc đưa ra một thử nghiệm mới sẽ giúp góp phần vào công cuộc phát triển của mảng Phát hiện vật thể 3D với việc sử dụng hình ảnh đơn.
- Cách hiện thực và phân bố mã nguồn GUPNet[33] rõ ràng, làm cho nhóm dễ tiếp cận.
- GUPNet là một thiết kế phù hợp với mục tiêu của nhóm đề ra vì không quá lỗi thời và đồng thời độ trễ không cao.

### 5.1 Tiền xử lý ảnh và tăng cường dữ liệu

Trong quá trình huấn luyện, các ảnh đầu vào được đưa về kích thước  $384 \times 1280$  điểm ảnh (chiều cao  $\times$  chiều dài). Trước tiên, thông qua sự ngẫu nhiên, bức ảnh sẽ được xác định xem sẽ thực hiện những phép biến đổi tăng cường nào, từ đó tính toán các thông số của phép biến đổi hoặc trực tiếp biến đổi ảnh. Sau đó, các thông số này sẽ được tổng hợp trong một ma trận biến đổi Affine. Bằng cách áp dụng ma trận này, ảnh sẽ được đưa về dạng kích thước chuẩn nói trên, đồng thời cũng thực hiện các phép biến đổi theo các thông số đã tính toán. Tiếp theo bức ảnh sẽ được chuẩn hóa với giá trị trung bình là  $[0.485, 0.456, 0.406]$  và độ lệch chuẩn là  $[0.229, 0.224, 0.225]$ . Cuối cùng, các nhãn của ảnh cũng được thay đổi để phù hợp với ảnh mới. Các kĩ thuật tăng cường dữ liệu được mô tả như sau:

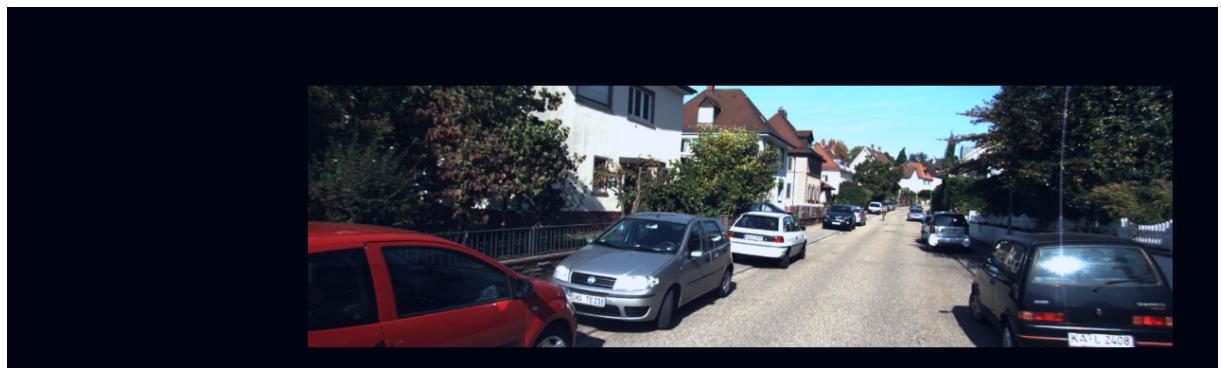
- **Scale và dời ngẫu nhiên:** Các bức ảnh sẽ được ngẫu nhiên scale theo kích thước ảnh gốc  $(H, W)$  và dời tâm của bức ảnh theo cả chiều trực x và y. Ảnh sau khi scale sẽ có kích thước trong khoảng  $(H \times 0.6, W \times 0.6)$  đến  $(H \times 1.4, W \times 1.4)$ . Tọa độ tâm bức ảnh mới cũng được dời riêng biệt theo chiều dọc và chiều ngang lần lượt một khoảng  $(-H \times 0.2, H \times 0.2)$  và  $(-W \times 0.2, W \times 0.2)$ . Như đã nhắc đến ở trên, các thông số mới sẽ được tổng hợp trong một phép biến đổi Affin để giữ thông tin của ảnh gốc. Kết quả của phép biến đổi được thể hiện trong hình 5.1c
- **Lật ảnh ngẫu nhiên :** thực hiện lật ảnh chiều ngang một cách ngẫu nhiên. Việc tăng cường dữ liệu này sẽ được thực hiện trực tiếp chứ không được tổng hợp trong ma trận Affine. Việc lật ảnh được minh họa ở hình 5.1d.



(a) Ảnh gốc.



(b) Ảnh sau khi được chuẩn hóa.



(c) Ảnh sau khi được thực hiện scale và dời tâm.



(d) Ảnh sau khi được thực hiện lật ngược

Hình 5.1: Các kiểu tăng cường dữ liệu.

## 5.2 Hướng tiếp cận

### 5.2.1 Sử dụng phân phối thay cho giá trị rời rạc

Sự khuếch đại lỗi do việc tận dụng phép chiếu để từ độ cao suy ra độ sâu sẽ khiến đầu ra khó kiểm soát. Điều này do mô hình không thể dự đoán độ không chắc chắn (Uncertainty) hoặc độ tin cậy ổn định. GUPNet đề xuất sử dụng một mô đun để suy ra độ sâu dựa trên hình thức phân phối thay vì là một giá trị rời rạc. Phân phối của độ sâu sẽ được suy ra từ một phân phối độ cao 3D được ước tính. Những đặc điểm thống kê của việc ước tính độ cao 3D sẽ được phản ánh trong phân phối độ sâu đầu ra, dẫn đến đưa ra sự không chắc chắn chính xác hơn. Cụ thể hơn, bài báo giả sử dự đoán của độ cao 3D cho mỗi vật thể là một phân phối Laplace  $La(\mu_h, \lambda_h)$  dựa theo hàm mật độ xác suất Laplace của một biến ngẫu nhiên  $X$   $La(\mu, \lambda)$ :

$$f_X(x) = \frac{1}{2\lambda} e^{(-\frac{|x-\mu|}{\lambda})} \quad (5.1)$$

Trong đó  $\mu$  và  $\lambda$  là tham số của phân phối Laplace. Từ đó, có thể suy ra được hàm mất mát của độ cao 3D:

$$\mathcal{L}_{h3d} = \frac{\sqrt{2}}{\sigma_h} |\mu_h - h_{3d}^{gt}| + \log(\sigma_h) \quad (5.2)$$

Các tham số  $\mu_h$  và  $\lambda_h$  được dự đoán trực tiếp từ mô hình, với  $\mu_h$  là giá trị hồi quy của độ cao 3D và  $\sigma_h$  là độ không chắc chắn của việc suy luận này. Độ không chắc chắn  $\sigma_h$  có giá trị tương ứng độ lệch chuẩn của phân phối. Khi ta thực hiện tối ưu hàm mất mát này, chênh lệch giữa độ cao 3D được hồi quy  $\mu_h$  và độ cao thực tế  $h_{3d}^{gt}$  cũng sẽ trở nên nhỏ đi. Đồng thời, những mẫu ở mức độ khó hoặc có nhãn bị nhiễu thông thường sẽ dẫn đến độ lệch chuẩn của phân phối lớn, nên phải chịu độ không chắc chắn  $\sigma_h$  lớn, từ đó thể hiện độ tin cậy của dự đoán thấp. Lưu ý là hàm mất mát ở trên có thể âm nếu dự đoán đủ tốt. Khi đó chênh lệch giữa  $\mu_h$  và  $h_{3d}^{gt}$  đủ nhỏ, đồng thời độ không chắc chắn cũng sẽ nhỏ, dẫn đến  $\log(\sigma_h)$  sẽ có giá trị âm. Dựa trên phân phối độ cao 3D đã học, khi áp dụng vào công thức 4.1, phân phối độ sâu của đầu ra phép chiếu  $La(\mu_p, \lambda_p)$  được thể hiện như sau:

$$\begin{aligned} d_p &= \frac{f \cdot h_{3d}}{h_{2d}} = \frac{f \cdot (\lambda_h \cdot X + \mu_h)}{h_{2d}} \\ &= \frac{f \cdot \lambda_h}{h_{2d}} \cdot X + \frac{f \cdot \mu_h}{h_{2d}}, \end{aligned} \quad (5.3)$$

trong đó  $X$  là phân phối Laplace chuẩn  $La(0, 1)$ . Ở trong trường hợp này, giá trị trung bình và độ lệch chuẩn của độ sâu  $d_p$  lần lượt là  $\frac{f \cdot \mu_h}{h_{2d}}$  và  $\frac{f \cdot \sigma_h}{h_{2d}}$ . Để dự đoán được độ sâu tốt hơn, tác giả đã thêm một bias được học để điều chỉnh kết quả ban đầu của phép chiếu. Tác giả cũng giả sử rằng bias này là một phân phối Laplace  $La(\mu_b, \lambda_b)$  và độc lập với độ sâu. Từ đó, phân phối độ sâu sau cùng được cấu trúc như sau:

$$\begin{aligned} d &= La(\mu_p, \sigma_p) + La(\mu_b, \sigma_b), \\ \mu_d &= \mu_p + \mu_b, \\ \sigma_d &= \sqrt{(\sigma_p)^2 + (\sigma_b)^2} \end{aligned} \quad (5.4)$$

Độ không chắc chắn  $\sigma_d$  sau cùng được gọi là Geometry based Uncertainty (GeU - Độ không chắc chắn dựa trên hình học). Độ không chắc chắn  $\sigma_d$  này sẽ phản ánh cả  $\mu_p$  và  $\mu_b$ . Với công

thức trên, một lượng nhỏ không chắc chắn của  $h_3d$  cũng sẽ được phản ánh trong  $\sigma_d$ . Để tối ưu phân phối độ sâu, tác giả áp dụng hàm mất mát hồi quy không chắc chắn:

$$\mathcal{L}_{depth} = \frac{\sqrt{2}}{\sigma_d} |\mu_d - d^{gt}| + \log(\sigma_d) \quad (5.5)$$

Ở đây tác giả giả sử phân phối độ sâu thuộc phân phối Laplace để đơn giản hóa. Hàm mất mát tổng quát sẽ làm cho kết quả của phép chiếu gần với độ sâu thực té  $d^{gt}$  và gradient cũng sẽ ảnh hưởng đồng thời đến bias độ sâu, độ cao 2D và độ cao 3D. Bên cạnh đó, uncertainty của độ cao 3D và bias độ sâu cũng sẽ được huấn luyện trong quá trình tối ưu. Trong khâu suy luận, độ tin cậy của độ sâu mang tính chủ chốt đối với sự ứng dụng trong thế giới thực. Một hệ thống suy luận đáng tin thì nên trả về điểm tin cậy cao đối với một dự đoán tốt và ngược lại, nên trả về điểm thấp cho những kết quả không tốt. Bởi vì GeU có khả năng thể hiện được uncertainty của độ sâu, nên có thể ánh xạ nó với một giá trị trong khoảng  $(0, 1)$  bằng một hàm mũ để thể hiện Uncertainty-Confidence (độ tin cậy của uncertainty):

$$p_{depth} = \exp(-\sigma_d) \quad (5.6)$$

Vì thể hiện độ tin cậy chính xác hơn cho mỗi phép chiếu độ sâu nên tác giả sử đã sử dụng độ tin cậy này dưới dạng điểm số của hộp bao quanh 3D trong khâu kiểm thử. Điểm tin cậy cuối cùng có thể tính toán như sau:

$$p_{3d} = p_{3d|2d} \cdot p_{2d} = p_{depth} \cdot p_{2d} \quad (5.7)$$

Con số này thể hiện độ tin cậy trong khâu phát hiện 2D và cả độ tin cậy của việc suy luận độ sâu nên mang lại độ đáng tin tốt hơn.

### 5.2.2 Cơ chế học phân tầng

Tại đầu khâu huấn luyện, ước tính độ cao 2D/3D sẽ có xu hướng bị nhiễu và lỗi sẽ bị phóng đại, dẫn đến ước tính độ sâu quá mức. Hậu quả là, quá trình huấn luyện của mạng sẽ bị dẫn dắt sai, làm cho hiệu suất cuối cùng bị suy thoái. GUPNet đề xuất một chiến lược học phân tầng (Hierarchical Task Learning - HTL), hướng tới việc đảm bảo các nhiệm vụ chỉ được huấn luyện khi các công việc trước đó (ví dụ ước tính độ cao 3D là một trong những nhiệm vụ trước của ước tính độ sâu) đã được huấn luyện tốt. Chiến lược này kiểm soát trọng số cho mỗi nhiệm vụ ở mỗi epoch, điều này có thể cải thiện đáng kể độ ổn định của việc huấn luyện, từ đó nâng cao được hiệu suất cuối cùng.

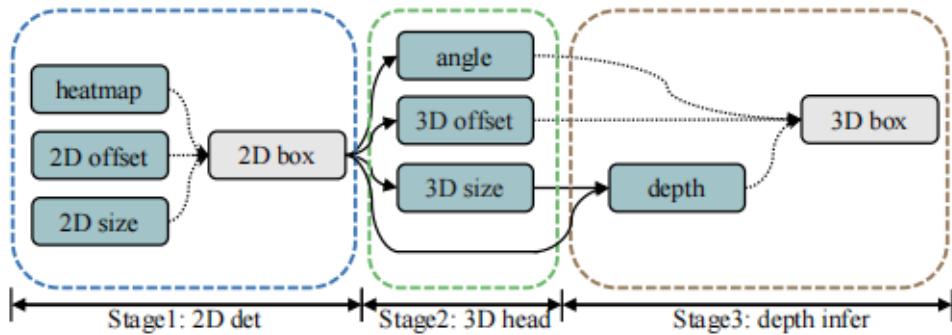
Với HTL, hàm mất mát tổng thể có thể được định nghĩa như sau :

$$\mathcal{L}_{total} = \sum_{j \in \mathcal{T}} w_i(t) \cdot \mathcal{L}_i \quad (5.8)$$

Trong đó  $\mathcal{T}$  là tập nhiệm vụ,  $t$  là chỉ số của epoch hiện tại,  $\mathcal{L}_i$  hàm mất mát nhiệm vụ thứ  $i$  và  $w_i(t)$  là trọng số của hàm mất mát nhiệm vụ thứ  $i$  tại epoch thứ  $t$ . HTL đảm bảo rằng là mỗi nhiệm vụ nên bắt đầu huấn luyện khi nhiệm vụ trước nó được huấn luyện đủ tốt. Tác giả chia các nhiệm vụ thành các giai đoạn khác nhau như trong hình 5.2.

Để huấn luyện từng nhiệm vụ, HTL đặt mục tiêu tăng dần  $w_i(t)$  từ  $0 \rightarrow 1$  khi quá trình huấn luyện tiến triển,  $w_i(t)$  được thiết kế dưới dạng hàm thời gian :

$$w_i(t) = \frac{t}{T}^{1-a_i(t)}, a_i(t) \in [0, 1], \quad (5.9)$$



Hình 5.2: Hệ thống phân cấp nhiệm vụ của GUP Net. Nguồn ảnh: [33].

Trong đó  $T$  là tổng số epoch huấn luyện,  $\frac{t}{T}$  có thể tự điều chỉnh theo thang thời gian.  $a_i(t)$  là một tham số điều chỉnh tại epoch  $t$ , tương ứng với các nhiệm vụ trước của nhiệm vụ  $i$ . Nếu các nhiệm vụ trước được huấn luyện tốt  $a_i(t)$  được kì vọng là sẽ lớn. Hình 5.3 cho thấy rằng  $a_i(t)$  càng lớn thì  $w_i(t)$  tăng càng nhanh,  $a_i(t)$  được định nghĩa như sau :

$$a_i(t) = \prod_{j \in P_i} ls_j \quad (5.10)$$

Trong đó  $P_i$  là tập nhiệm vụ trước nhiệm vụ thứ  $i$ ,  $ls_j$  là chỉ số phản ánh tình hình học tập của nhiệm vụ thứ  $j$ , có giá trị trong khoảng  $[0,1]$ . Khi các tất cả các nhiệm vụ trước nhiệm vụ  $i$  được  $ls$  cao (đào tạo tốt) thì dĩ nhiên  $a_i(t)$  cũng sẽ cao .

$$ls_j(t) = \frac{\mathcal{DF}_j(K) - \mathcal{DF}_j(t)}{\mathcal{DF}_j(K)} \quad (5.11)$$

$$\mathcal{DF}_j(t) = \frac{1}{K} \sum_{\hat{t}=t-K}^{t-1} |\mathcal{L}'_j(\hat{t})| \quad (5.12)$$

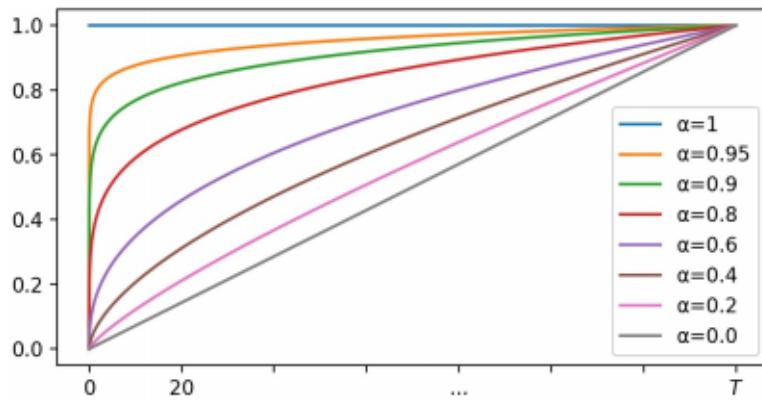
Trong đó  $\mathcal{L}'_j(\hat{t})$  là đạo hàm của  $\mathcal{L}_j(\cdot)$  tại epoch thứ  $\hat{t}$ , đạo hàm này có thể chỉ ra xu hướng thay đổi cục bộ của hàm mất mát.  $\mathcal{DF}_j(t)$  tính toán trung bình của các giá trị đạo hàm trong  $K$  epoch trước epoch thứ  $t$  để phản ánh xu hướng thay đổi trung bình. Vì vậy, công thức  $ls_j$  có nghĩa là so sánh sự khác biệt giữa xu hướng hiện tại  $\mathcal{DF}_j(t)$  và xu hướng của  $K$  epoch đầu tiên khi bắt đầu huấn luyện  $\mathcal{DF}_j(K)$  cho nhiệm vụ thứ  $j$ . Nếu xu hướng mất mát hiện tại tương tự như xu hướng bắt đầu,  $ls_j$  sẽ cho một giá trị nhỏ, điều đó có nghĩa là nhiệm vụ này đã không được huấn luyện tốt. Ngược lại, nếu một tác vụ có xu hướng hội tụ thì  $ls_j$  sẽ gần bằng 1, nghĩa là trạng thái học của nhiệm vụ này được thỏa mãn.

Dựa trên thiết kế tổng thể, trọng số giảm của mỗi thuật ngữ có thể phản ánh linh hoạt tình hình học tập của các nhiệm vụ trước đó, điều này có thể giúp quá trình huấn luyện ổn định hơn.

### 5.3 Thử nghiệm lại

Một số môi trường lớn mà tác giả sử dụng để hiện thực mô hình như sau:

- Python phiên bản 3.6
- Pytorch phiên bản 1.1



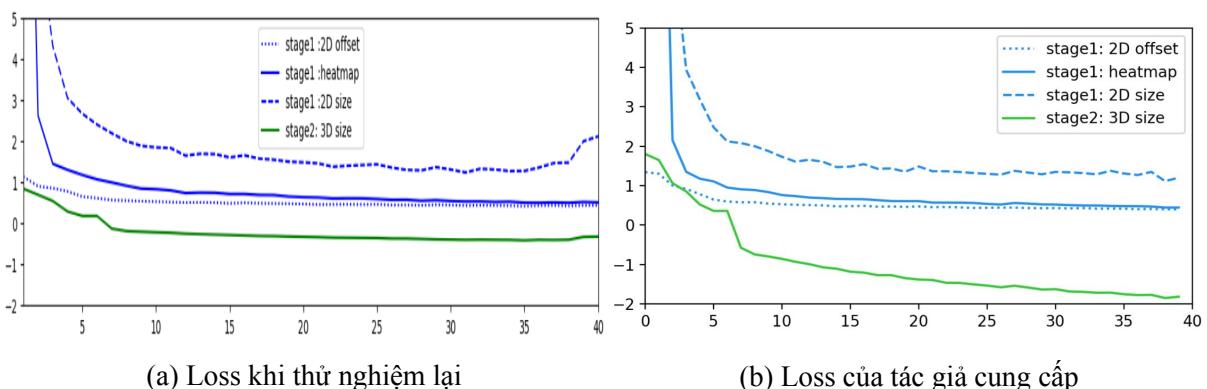
Hình 5.3: Hàm lập lịch thời gian đa thức với tham số điều chỉnh. Trục tung là giá trị của  $w_i(t)$  và trục hoành là chỉ số epoch t. Nguồn ảnh : [33].

	Tác giả	Nhóm
Batch size	32	16
Learning rate	0.00125	0.000625

Bảng 5.1: Một số cài đặt khác với tác giả

- Cuda phiên bản 9.0
- Ngoài ra còn một số thư viện phụ trợ như scikit-image, opencv, tqdm, pyyaml, matplotlib

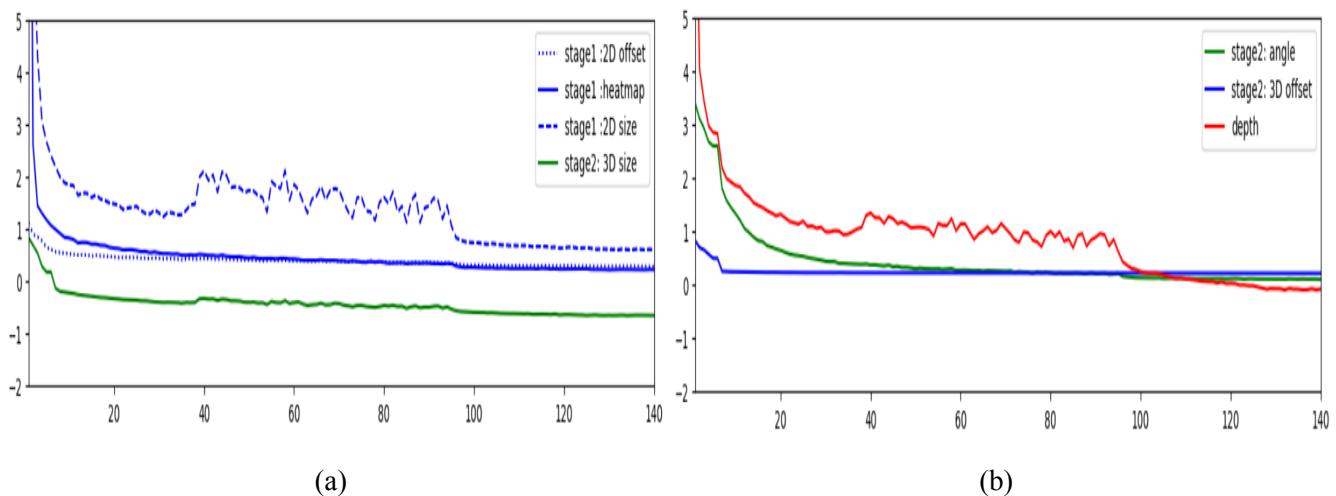
Nhóm chúng em đã thử nghiệm lại mô hình với những thiết lập tương đồng của tác giả (trừ một số các thành phần được liệt kê trong bảng 5.1) trên 2 GPU V100 với RAM đồ họa là 32 GB ở mỗi GPU. Việc huấn luyện总共 khoảng 9 tiếng để chạy hết 140 epoch và lượng GPU RAM phải sử dụng trong lúc huấn luyện là khoảng 11 GB cho mỗi GPU. Do phần hiện thực mô hình của tác giả sử dụng một số hàm không hỗ trợ cho việc thử nghiệm lại (ví dụ như ROI Align) nên tổng quát phần hiện thực không thể xóa bỏ hoàn toàn sự ngẫu nhiên trong lúc huấn luyện, nên kết quả sẽ chấp nhận sai lệch so với tác giả cung cấp.



Hình 5.4: So sánh về loss của 40 epoch đầu tiên.

Quá trình thử nghiệm lại của nhóm gồm hai phiên bản :

- Huấn luyện mô hình trên cả ba lớp (Car, Pedestrian, Cyclist).
- Huấn luyện mô hình chỉ với mỗi lớp Car.



Hình 5.5: Loss của 140 epoch. a) Loss của khâu 1 và khâu 2. b) Loss của khâu 2 và khâu 3

Nhóm chúng em đã thực hiện thử nghiệm khá nhiều lần và chọn ra các phiên bản tốt nhất để tiếp tục phát triển hướng đi mới. Bản thân tác giả cũng đã huấn luyện rất nhiều lần và chọn kết quả tốt nhất để đưa lên bài nghiên cứu. Nhóm chúng em cũng vẽ biểu đồ để có thể so sánh loss với tác giả tại 40 epoch đầu tiên. Chi tiết được thể hiện ở hình 5.4.

Về phần loss khi thử nghiệm lại, các chỉ số không tốt bằng khi so với số liệu tác giả cung cấp. Loss của kích thước 3D khi bắt đầu thấp hơn và sau cùng không tiếp tục giảm sâu được, đồng thời loss của kích thước 2D của vật thể không được ổn định từ epoch 30 trở đi. Những điều này được thể hiện rõ hơn khi thực hiện minh họa trên cả 140 epoch ở hình 5.5a. Ta có thể thấy rõ được sự bất ổn định của kích thước 2D trong khoảng từ epoch 35 đến epoch 100, từ đó dẫn đến sự bất ổn định của độ sâu do có sự tương quan như đã đề cập ở công thức 4.1.

Qua quá trình lân lượt đánh giá các chỉ số trong khi huấn luyện ở cả hai phiên bản mô hình cơ sở mà nhóm đã thử nghiệm lại, các kết quả tốt nhất của hai phiên bản sẽ được sử dụng để so sánh với kết quả sau khi áp dụng giải pháp cải thiện ở chương sau. Về các chỉ số đánh giá từ kết quả không tốt bằng chỉ số của tác giả đưa ra, điều này có lẽ do thời gian của việc thử nghiệm chưa đủ, vì bản thân tác giả khi thử nghiệm lại cũng ra những kết quả không tốt bằng. Kết quả chi tiết được mô tả ở các bảng 5.2 và 5.3.

Method	$Car AP_{3D R_{40}}$			$Pedestrian AP_{3D R_{40}}$			$Cyclist AP_{3D R_{40}}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
GUPNet[33]	<b>23.17</b>	<b>16.23</b>	<b>13.56</b>	<b>11.47</b>	8.17	6.74	9.48	5.00	4.13
GUPNet(reproduce)	21.25	15.46	13.04	11.42	<b>8.73</b>	<b>7.01</b>	<b>10.78</b>	<b>5.42</b>	<b>4.46</b>

Bảng 5.2: Hiệu suất phát hiện đối tượng 3D trên tập Validation KITTI khi huấn luyện với ba lớp. Số liệu của GUPNet được lấy từ checkpoint đã được công bố trên trang Github<sup>1</sup> của tác giả. Các số liệu tốt nhất được in đậm.

<sup>1</sup><https://github.com/SuperMHP/GUPNet>

Method	$AP_{3D R_{40}}$			$AP_{BEV R_{40}}$		
	Easy	Mod.	Hard	Easy	Mod.	Hard
GUPNet[33]	22.76	<b>16.46</b>	13.72	<b>31.07</b>	<b>22.94</b>	<b>19.75</b>
GUPNet(reproduce)	<b>23.29</b>	16.37	<b>14.54</b>	30.30	22.54	19.60

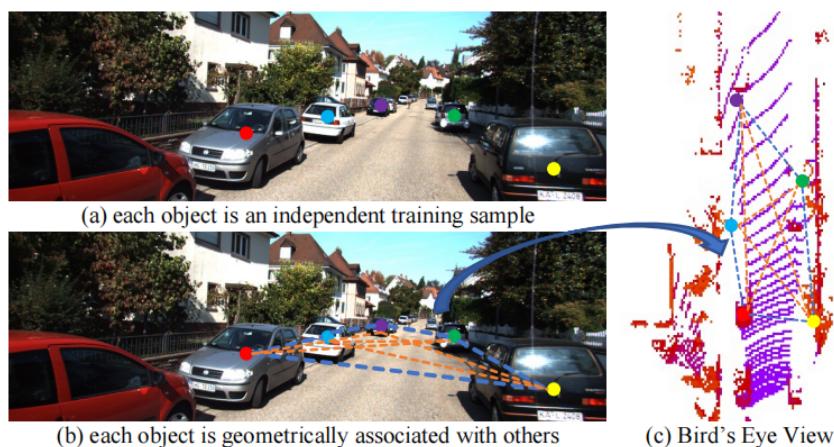
Bảng 5.3: Hiệu suất phát hiện đối tượng 3D của lớp Car trên tập Validation KITTI khi chỉ huấn luyện với lớp Car. Số liệu của GUPNet được lấy từ GUPNet[33]. Các số liệu tốt nhất được in đậm.

# Chương 6

## Giải pháp phát triển

Hầu hết các phương pháp hiện nay (trong đó có cả mô hình cơ sở) coi mỗi đối tượng 3D trong hình ảnh là một đối tượng được huấn luyện độc lập, tức là trong hàm mắt mèo, ta chỉ xem xét sai lệch giữa dự đoán của một đối tượng so với nhãn của chúng, trong khi bỏ qua các quan hệ liên kết với nhau vốn có. Do đó chắc chắn dẫn đến việc hạn chế về tận dụng những ràng buộc về hình học trong không gian. Có hai quan sát chính về việc này: đầu tiên là những thông tin trong khâu phát hiện 2D có thể được sử dụng dẫn dắt để ràng buộc và giám sát quá trình huấn luyện ở khâu Phát hiện 3D; thứ hai, vị trí của một vật thể sẽ bị ảnh hưởng toàn cục bởi những vật thể xung quanh khác trong hình, minh họa ở hình 6.1. Để giải quyết điều này, một hàm mắt mèo khả vi, gọi là Homography được đề xuất, để thực hiện việc chuyển đổi từ không gian hình ảnh 2D sang không gian Bird Eye's View 3D và đồng thời ràng buộc mối quan hệ hình học toàn cục của tất cả các vật thể.

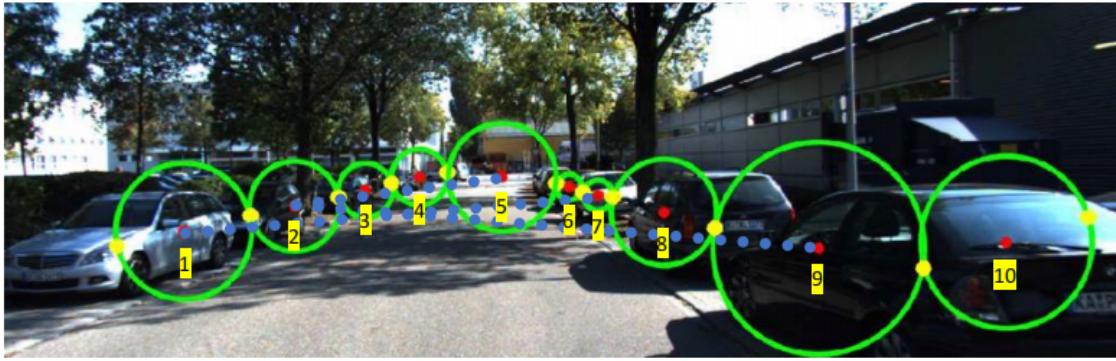
Nhờ thiết kế ngắn gọn, hàm mắt mèo Homography được giới thiệu có thể được thêm vào bất kỳ công cụ phát hiện vật thể 3D bằng hình ảnh đơn nào, đồng thời tăng đáng kể hiệu suất so với mức cơ bản của chúng. Đồng thời việc áp dụng hàm mắt mèo cũng không ảnh hưởng đến độ trễ khi chạy, là một lý do để chọn giải pháp này. Hiệu suất của hai mô hình ImVoxelNet[49] và MonoFlex[66] được thể hiện trong bảng 6.1.



Hình 6.1: Mô tả ràng buộc hình học của homography. (a) Hầu hết các phương pháp hiện tại coi mỗi đối tượng là một mẫu huấn luyện duy nhất, (b) mắt mèo Homography thiết lập kết nối giữa các đối tượng và áp dụng phát hiện 2D làm hướng dẫn để giúp hạn chế định vị 3D trong (c) chế độ Bird Eye's View. Nguồn ảnh: [15].

Method	Extra data	$AP_{3D R_{40}}$			$AP_{BEV R_{40}}$			Time(s)
		Easy	Moderate	Hard	Easy	Moderate	Hard	
Mono-PLiDar[63]	Depth	10.76	7.50	6.10	21.27	13.92	11.25	0.10
PatchNet[35]	Depth	15.68	11.12	10.17	22.97	16.86	14.97	0.40
D4LCN[11]	Depth	16.65	11.72	9.51	22.51	16.03	12.55	0.20
MonoRUN[8]	Depth	19.95	12.30	10.58	27.94	17.34	15.24	0.07
Kinematic3D[3]	Temporal	19.07	12.72	9.17	26.69	17.52	13.10	0.12
Aug3D-RPN[16]	Depth	17.82	12.99	9.78	26.00	17.89	14.18	0.08
DFR-Net[17]	Depth	19.40	13.63	10.35	28.17	19.17	14.84	0.18
CaDDN[46]	LiDAR	19.17	13.41	11.46	27.94	18.91	17.19	0.63
MonoEF[70]	Depth	21.29	13.87	11.71	29.03	19.70	17.26	0.03
Autoshape[32]	Shape	22.47	14.17	11.36	30.06	20.08	15.59	0.04
M3D-RPN[2]	-	14.76	9.71	7.42	21.02	13.67	10.23	0.16
SMOKE[31]	-	14.03	9.76	7.84	20.83	14.49	12.75	0.03
MonoPair[10]	-	13.04	9.99	8.65	19.28	14.83	12.89	0.06
RTM3D[25]	-	14.41	10.34	8.77	19.17	14.20	11.99	0.05
PGD-FSCOS3D[58]	-	19.05	11.75	9.39	26.89	16.51	13.49	0.03
M3DSSD[34]	-	17.51	11.46	8.98	24.15	15.93	12.11	0.16
MonoDLE[38]	-	17.23	12.26	10.29	24.79	18.89	16.00	0.04
MonoRCNN[51]	-	18.36	12.65	10.03	25.48	18.11	14.10	0.07
ImVoxelNet[49]	-	17.15	10.97	9.15	25.19	16.37	13.58	0.20
ImVoxelNet(+homo)	-	20.10	12.99	10.50	29.18	19.25	16.21	0.20
MonoFlex[66]	-	19.94	13.89	12.07	28.23	19.75	16.89	0.03
MonoFlex(+homo)	-	<b>21.75</b>	<b>14.94</b>	<b>13.07</b>	<b>29.60</b>	<b>20.68</b>	<b>17.81</b>	<b>0.03</b>

Bảng 6.1: Hiệu suất của một vài phương pháp trong việc phát hiện vật thể 3D đối với phân loại Xe trên tập test của KITTI. Chỉ số cao nhất được tô đậm (chỉ so sánh với các phương pháp sử dụng hình ảnh đơn). 'Extra Data' liệt kê những phương pháp sử dụng thông tin thêm, trong đó 'Depth' là sử dụng một số dự đoán.



Hình 6.2: Vị trí của đối tượng mục tiêu bị ảnh hưởng toàn cục bởi các đối tượng khác. Khi một đối tượng riêng lẻ chỉ kết nối cục bộ với đối tượng gần nhất của nó, như đề xuất của [10] thì mối quan hệ theo cặp này không đủ để mã hóa mối quan hệ không gian của các vật thể. Thay vào đó, chúng ta xem xét đến việc ảnh hưởng toàn cục. Ví dụ như, vị trí của Ô tô 2 không chỉ bị ảnh hưởng bởi Ô tô 1 mà còn bị ràng buộc bởi Ô tô 5 và 9 khi được kết nối với đường chấm màu xanh lam. Nguồn ảnh: [15].

## 6.1 Lý thuyết

### 6.1.1 Homography

Phép chiếu phối cảnh ánh xạ những điểm tọa độ ở không gian 3D ở thế giới thực đến một mặt phẳng hình ảnh 2D đọc theo một đường thẳng theo ống kính máy ảnh. Điều này có thể được thực hiện thông qua các ma trận ngoại và nội máy ảnh như đã liệt kê ở phần 2.2. Với việc sử dụng homography, ta có thể chuyển đổi giữa tọa độ tương ứng giữa hai mặt phẳng. Nói cách khác, homography là một phép ánh xạ giữa hai mặt phẳng mà vẫn giữ được tính cộng tuyến (collinearity). Ta có thể sử dụng homography để biến đổi các điểm từ góc nhìn của 1 máy ảnh này sang 1 máy ảnh khác:

$$\mathbf{X}_2 = \mathbf{H}\mathbf{X}_1 \quad (6.1)$$

với  $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^3$  lần lượt là tọa độ thuần nhất của điểm trong lần lượt máy ảnh 1 và 2,  $\mathbf{H} \in \mathbb{R}^{3 \times 3}$ . Thực hiện chia tỉ lệ để có được tọa độ trong mặt phẳng hình ảnh, ta có:

$$\lambda_1 \mathbf{x}_1 = \mathbf{X}_1, \lambda_2 \mathbf{x}_2 = \mathbf{X}_2, \quad (6.2)$$

với  $\mathbf{x}_1, \mathbf{x}_2$  lần lượt là tọa độ thuần nhất ở máy ảnh 1 và 2 sau khi đã chia tỉ lệ để tọa độ có dạng là  $x = [u, v, 1]^T$ , trong đó  $[u, v]$  là tọa độ 2D của 1 điểm trong mặt phẳng hình ảnh. Vì vậy ta có:

$$\lambda_2 \mathbf{x}_2 = \mathbf{H} \lambda_1 \mathbf{x}_1, \quad (6.3)$$

$$s \mathbf{x}_2 = \mathbf{H} \mathbf{x}_1, \quad (6.4)$$

với  $s = \lambda_2 / \lambda_1$  là hệ số tỉ lệ,  $s \in \mathbb{R} \setminus \{0\}$ . Như vậy, khi tìm được  $\mathbf{H}$ , ta có thể biến đổi các điểm từ góc nhìn của 1 máy ảnh này sang 1 máy ảnh khác.

Homography là một ràng buộc hình học toàn cục. Bởi vì mọi cặp điểm tương ứng sẽ tham gia vào việc tìm ma trận homography nên giải pháp sử dụng homography sẽ được đảm bảo tối

uru toàn cục. Vì việc ràng buộc bởi việc mọi cặp điểm tương ứng cuối cùng sẽ ảnh hưởng đến toàn quá trình tối ưu. Đồng thời, trong hình học chiếu, homography là một đặc tính của không gian chiếu, nghĩa là tương quan một tập các điểm từ một mặt phẳng đến một mặt phẳng khác mà vẫn bảo toàn được các tính chất hình học, chẳng hạn như tính cộng tuyến.

### 6.1.2 Ước tính ma trận homography

Ở 6.4,  $s\mathbf{x}_2$  thể hiện việc thực hiện phép nhân ở về phải sẽ khiến ta có được một tọa độ thuần nhất có dạng là  $\mathbf{x}'_2 = [x'_2, y'_2, z'_2]^T$  ( $z'_2$  là  $s$ ) ở về trái. Khi viết lại chi tiết ở dạng ma trận, ta có:

$$\begin{bmatrix} x'_2 \\ y'_2 \\ z'_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (6.5)$$

Sau đó có thể suy ra được:

$$\begin{aligned} x'_2 &= H_{11}x_1 + H_{12}y_1 + H_{13}z_1 \\ y'_2 &= H_{21}x_1 + H_{22}y_1 + H_{23}z_1 \\ z'_2 &= H_{31}x_1 + H_{32}y_1 + H_{33}z_1 \end{aligned} \quad (6.6)$$

Sau đó ta chuyển về dạng tọa độ thuần nhất của mặt phẳng hình ảnh bằng cách chia tỉ lệ ( $x_2 = \frac{x'_2}{z'_2}$ ,  $y_2 = \frac{y'_2}{z'_2}$ ), đồng thời  $z_1 = 1$  vì ta đã chuyển  $\mathbf{x}_1$  sang tọa độ thuần nhất của 1 điểm 2D trong 1 mặt phẳng ở công thức 6.2:

$$\begin{aligned} x_2 &= \frac{H_{11}x_1 + H_{12}y_1 + H_{13}}{H_{31}x_1 + H_{32}y_1 + H_{33}} \\ y_2 &= \frac{H_{21}x_1 + H_{22}y_1 + H_{23}}{H_{31}x_1 + H_{32}y_1 + H_{33}} \end{aligned} \quad (6.7)$$

Sau khi sắp xếp lại, ta có:

$$\begin{aligned} x_2H_{31}x_1 + x_2H_{32}y_1 + x_2H_{33} - H_{11}x_1 - H_{12}y_1 - H_{13} &= 0 \\ y_2H_{31}y_1 + x_2H_{32}y_1 + y_2H_{33} - H_{21}x_1 - H_{22}y_1 - H_{23} &= 0 \end{aligned} \quad (6.8)$$

Từ đây, ta có thể chuyển các thành phần của  $H$  thành dạng véc tơ:

$$\begin{aligned} \mathbf{a}_x \cdot \mathbf{h} &= 0 \\ \mathbf{a}_y \cdot \mathbf{h} &= 0 \end{aligned} \quad (6.9)$$

trong đó

$$\begin{aligned} \mathbf{h} &= (H_{11}, H_{12}, H_{13}, H_{21}, H_{22}, H_{23}, H_{31}, H_{32}, H_{33})^T \\ \mathbf{a}_x &= (-x_1, -y_1, -1, 0, 0, 0, x_2x_1, x_2y_1, x_2) \\ \mathbf{a}_y &= (0, 0, 0, -x_1, -y_1, -1, y_2x_1, y_2y_1, y_2) \end{aligned}$$

Với danh sách các điểm tương ứng ở hai mặt phẳng, ta có thể xây dựng một hệ phương trình tuyến tính:

$$\mathbf{A}\mathbf{h} = \mathbf{0} \quad (6.10)$$

trong đó

$$\mathbf{A} = \begin{pmatrix} \mathbf{a}_{x1} \\ \mathbf{a}_{y1} \\ \mathbf{a}_{x2} \\ \mathbf{a}_{y2} \\ \vdots \\ \mathbf{a}_{xN} \\ \mathbf{a}_{yN} \end{pmatrix}$$

với  $N$  là số cặp điểm tương ứng. Lưu ý vì  $\mathbf{H} \in \mathbb{R}^{3 \times 3}$  và công thức 6.4 chứa hệ số tỉ lệ nên bậc tự do  $\mathbf{H}$  là 8, đồng nghĩa với việc ta cần ít nhất 4 cặp điểm tương ứng từ 2 mặt phẳng để tìm  $\mathbf{H}$ . Việc giải phương trình 6.10 có thể sử dụng SVD vì đây là bài toán Homogeneous Linear Least Squares như đã trình bày ở 2.7.

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T = \sum_{i=1}^9 \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad (6.11)$$

Các giá trị  $\sigma_i$  sau khi tính toán sẽ được sắp xếp theo thứ tự nhỏ dần và  $\sigma_9$  nhỏ nhất. Có ba trường hợp cho giá trị của  $\sigma_9$ :

- Nếu đủ chính xác số cặp điểm tương ứng thì  $\sigma_9 = 0$ , khi ấy tồn tại một ma trận homography có thể ánh xạ chính xác các cặp điểm.
- Khi số cặp điểm nhiều hơn 4 thì  $\sigma_9 >= 0$ , khi ấy  $\sigma_9$  biểu diễn một ánh xạ xấp xỉ.
- Với trường hợp số cặp điểm nhỏ hơn 4 thì ta sẽ không xét.

Từ kết quả của SVD, ta lấy cột cuối cùng từ  $\mathbf{V}$ , vec tơ tương ứng với giá trị của  $\sigma_9$ . Đây là kết quả  $\mathbf{h}$  mà ta cần tìm, chứa các hệ số của ma trận homography phù hợp với ánh xạ giữa các cặp điểm giữa 2 mặt phẳng nhất. Cuối cùng, ta chuyển dạng của  $\mathbf{h}$  thành ma trận  $\mathbf{H} \in \mathbb{R}^{3 \times 3}$  và xây dựng được phương trình  $\mathbf{x}_2 \approx \mathbf{Hx}_1$ .

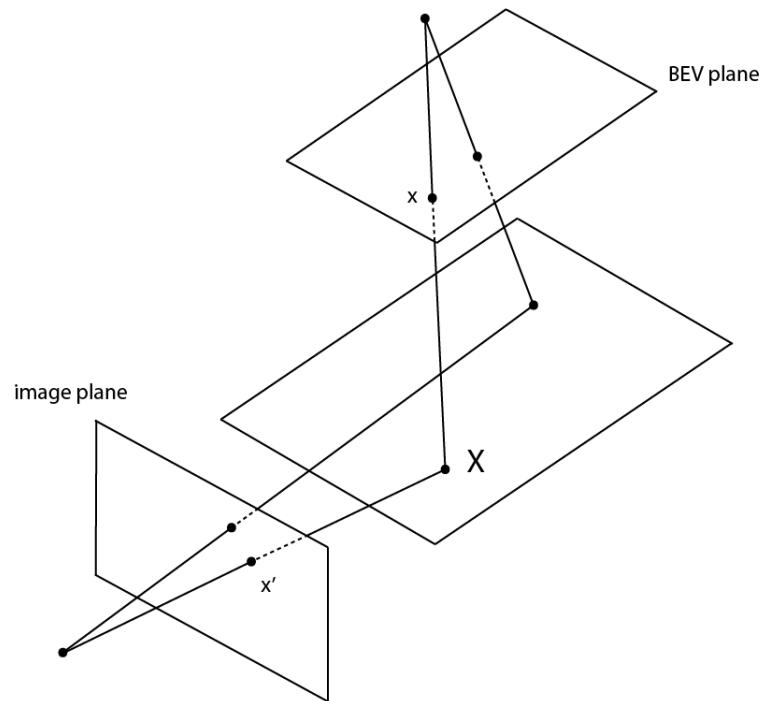
## 6.2 Hàm măt măt Homography

Dựa vào các cơ sở lý thuyết trên, hàm măt măt homography được đề xuất với mục đích tạo ra các liên kết hình học giữa toàn bộ các vật thể bằng cách tận dụng ma trận homography. Thông thường, các phương pháp phát hiện vật thể 3D trong hình ảnh đơn có thể dự đoán các hộp bao quanh 3D dưới việc học có giám sát với nhãn và cùng với đó là phân lớp và sử dụng hàm măt măt hồi quy trong quy trình. Hàm măt măt Homography sẽ phạt mỗi quan hệ sai giữa tất cả các hộp được dự đoán và điều chỉnh các vị trí sau cùng. Các bước chính được liệt kê như sau.

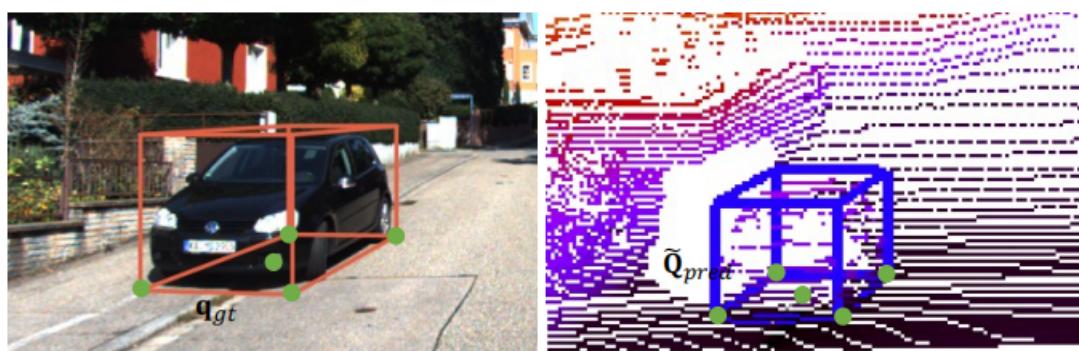
### 6.2.1 Mô hình hóa các điểm ứng viên

Giả sử chúng ta có các hộp dự đoán  $box_{pred}$  thu được từ bộ phát hiện 3D tùy ý và hộp 3D thực tế tương ứng  $box_{gt}$ . Như đã nhắc đến ở đầu chương, ta sẽ sử dụng ma trận homography để mô tả mối quan hệ hình chiếu giữa mặt phẳng hình ảnh và mặt phẳng Bird Eye's View (BEV). Quan hệ này được minh họa ở hình 6.3. Đối với mỗi đối tượng, như được minh họa trong hình 6.4, ta chọn năm điểm ở đây  $\mathbf{Q}_{pred} = [x_{pred}, y_{pred}, z_{pred}]^T$  của  $box_{pred}$  làm đại diện, bao gồm một điểm là tâm đáy và bốn điểm ở góc đáy.

Tác giả cũng giả sử rằng tất cả các vật luôn nằm trên mặt phẳng mặt đất, do đó các điểm dưới cùng trên mặt phẳng BEV có thể là được đơn giản hóa thành  $\tilde{\mathbf{Q}}_{pred} = [x_{pred}, y_{pred}]^T$ . Tương tự,



Hình 6.3: Hình minh họa mối quan hệ hình chiếu giữa hai mặt phẳng Bird Eye's View và hình ảnh.



Hình 6.4: Điểm ứng viên 2D và 3D của một đối tượng. Nguồn: [15].

chúng ta có  $\mathbf{Q}_{gt} = [x_{gt}, y_{gt}, z_{gt}]^T$  thu được từ  $box_{gt}$ . Sau khi thực hiện phép chiếu máy ảnh, hộp 3D thực tế sẽ được chuyên đổi vào không gian hình ảnh, được xác định bởi công thức :

$$\mathbf{q} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\mathbf{Q} \quad (6.12)$$

Trong đó  $\mathbf{K}$ ,  $[\mathbf{R}|\mathbf{t}]$  lần lượt là ma trận tham số nội và ngoại máy ảnh,  $\mathbf{q} = [u, v]^T$  đại diện cho điểm ảnh là kết quả của phép chiếu trên mặt phẳng hình ảnh, áp dụng được cho cả  $box_{gt}$  và  $box_{pred}$ . Do đó, nếu tồn tại N đối tượng, chúng ta có được 5N cặp điểm ứng viên  $\mathbf{q}_{pred}$ ,  $\tilde{\mathbf{Q}}_{pred}$  cho  $box_{pred}$  và  $\mathbf{q}_{gt}$ ,  $\tilde{\mathbf{Q}}_{gt}$  cho  $box_{gt}$ , tương ứng, được chuẩn bị để tính toán ma trận homography.

### 6.2.2 Tính toán Homography

Để ràng buộc ngầm vị trí tương đối của từng đối tượng mà không làm mất tính tổng quát, ta chọn  $\mathbf{q}_{gt}$  và  $\tilde{\mathbf{Q}}_{pred}$ . Cụ thể, tác giả sử dụng tọa độ  $\mathbf{q}_{gt}$  trong mặt phẳng hình ảnh 2D để dẫn dắt, sửa các vị trí cuối cùng  $\tilde{\mathbf{Q}}_{pred}$  trong không gian 3D. Công thức được xác định với tọa độ thuận nhất như sau (tạm bỏ qua hệ số scale):

$$\begin{aligned} \tilde{\mathbf{Q}}_{pred} &= \mathbf{H}\mathbf{q}_{gt}, \\ \begin{pmatrix} x_{pred} \\ y_{pred} \\ 1 \end{pmatrix} &= \mathbf{H} \begin{pmatrix} u_{gt} \\ v_{gt} \\ 1 \end{pmatrix}. \end{aligned} \quad (6.13)$$

Ở đây,  $\mathbf{H}$  lưu trữ các mối quan hệ lẫn nhau của tất cả các đối tượng bằng cách phép ánh xạ giữa hai khung nhìn. Ma trận homography  $\mathbf{H}$  khi hiện thực có thể được tính bằng SVD của Pytorch. Trong thực tế, ma trận homography trong phương trình 6.13 được ước tính trong khi  $\tilde{\mathbf{Q}}_{pred}$  có thể sai lệch rất nhiều so với thực tế tại giai đoạn đầu huấn luyện. Tác giả kí hiệu ma trận  $\mathbf{H}$  khi còn trong giai đoạn sai lệch nhiều này là  $\hat{\mathbf{H}}$  và  $\tilde{\mathbf{Q}}_{homo} = \hat{\mathbf{H}}\mathbf{q}_{gt}$ . Trong quá trình huấn luyện, giá trị được ước tính  $\tilde{\mathbf{Q}}_{homo}$  sẽ dần tiếp cận  $\tilde{\mathbf{Q}}_{pred}$  và  $\tilde{\mathbf{Q}}_{gt}$ .

### 6.2.3 Hàm măt măt

Thông thường, việc phát hiện 3D được xem xét độc lập, riêng biệt với mỗi vật thể và bị ràng buộc bởi một hàm măt măt hồi quy như là  $L_{reg} = L1(\tilde{\mathbf{Q}}_{gt} - \tilde{\mathbf{Q}}_{pred})$ . Nhóm sử dụng một hàm măt măt mới, được gọi là hàm măt măt Homography, để tối ưu hóa các vị trí với các ràng buộc không gian mạnh mẽ. Hàm măt măt Homography được định nghĩa như sau:

$$\begin{aligned} L_{homo} &= \text{SmoothL1}(\tilde{\mathbf{Q}}_{gt} - \tilde{\mathbf{Q}}_{homo}) \\ &= \text{SmoothL1}(\tilde{\mathbf{Q}}_{gt} - \hat{\mathbf{H}}\mathbf{q}_{gt}) \end{aligned} \quad (6.14)$$

Khác với hàm măt măt hồi quy, việc tính toán ma trận Homography  $\hat{\mathbf{H}}$  sẽ xem xét tất cả các cặp điểm tương ứng. Do đó, đây là hàm măt măt toàn cục đối với ràng buộc hình học, được sử dụng để hướng dẫn dự đoán các vị trí 3D từ vị trí 2D của nhãn. Một khác, bằng cách tối ưu hóa 6.14,  $\hat{\mathbf{H}}$  cũng có xu hướng gần với ma trận homography chuẩn giữa hai mặt phẳng ( $\mathbf{H}$ ) hơn. Một ưu điểm của hàm măt măt Homography là nó có thể khả vi. Nó có thể là một mô-đun dễ dàng thêm vào cho bất kỳ phương pháp phát hiện đối tượng 3D trong hình ảnh đơn nào và đóng vai trò như một ràng buộc không gian mạnh mẽ để định vị các đối tượng 3D.

Tuy nhiên, vì một số giá trị trong  $\mathbf{Q}_{homo}$  sau khi tính toán bị lệch khá lớn với tọa độ hộp thực tế dẫn đến quá trình huấn luyện không ổn định, nên nhóm lọc đi các giá trị đó để quá trình huấn luyện ổn định hơn. Các giá trị này sẽ được tính toán riêng biệt dưới dạng hàm măt măt hồi quy. Hàm măt măt Homography sau cùng được thể hiện như sau:

- Nếu toàn bộ điểm nằm trong ngưỡng giá trị:

$$L_{homo\_filter} = \frac{2}{3}L_{homo} + \frac{1}{3}L_{reg\_z} \quad (6.15)$$

trong đó  $L_{reg\_z}$  là hàm mất mát hồi quy đối với tọa độ  $z$ . Việc thêm vào hàm mất mát này sẽ giúp cải thiện về độ cao khi ta đã thực hiện đơn giản hóa độ cao ở phần 6.2.1.

- Nếu có điểm nằm ngoài khoảng ngưỡng giá trị:

$$L_{homo\_filter} = \frac{2}{3}(L_{in} * r + L_{out} * (1 - r)) + \frac{1}{3}L_{reg\_z} \quad (6.16)$$

trong đó  $r$  là tỉ lệ của các điểm nằm trong ngưỡng giá trị,  $L_{out}$  là hàm mất mát hồi quy đối với các điểm nằm ngoài ngưỡng và  $L_{in}$  là  $L_{homo}$  khi chỉ xét các điểm nằm trong ngưỡng giá trị. Lưu ý: các hàm mất mát phía trên liên quan đến Homography đều sử dụng SmoothL1.

Các ngưỡng giá trị dùng để lọc được chọn từ việc quan sát các khoảng giá trị thực tế của các vật thể trong tập dữ liệu.

## 6.3 Hiện thực hàm mất Homography

Hàm mất Homography đã được tác giả hiện thực vào hai công trình phát hiện vật thể 3D điển hình là ImVoxelNet[49] và MonoFlex[66] thuộc hai phương pháp hiện đối tượng một giai đoạn. Vì vậy nhóm xin đề xuất **H-GUPNet**, tích hợp thiết kế cơ sở GUPNet - một nghiên cứu phát hiện đối tượng hai giai đoạn với hàm mất Homography.

Xét về mặt thiết kế, MonoFlex và GUP đều dự đoán tâm chiếu 3D bằng bản đồ nhiệt và hộp 3D được chiếu (bao gồm độ sâu, kích thước và hướng), nên nhóm vẫn giữ lại phần lớn các thiết lập mà tác giả đã áp dụng khi tích hợp hàm mất Homography vào MonoFlex cho GUP.

### 6.3.1 Hàm mất tổng thể

Hàm mất chính có thể được mô tả là sự kết hợp của hàm mất tổng của GUPNet[33] và hàm mất Homography đã nêu ở phần 6.14.

$$L = L_{GUP} + w_{homo}(t)L_{homo\_filter} \quad (6.17)$$

Trong đó  $L_{GUP}$ ,  $L_{homo\_filter}$  lần lượt là hàm mất tổng của GUPNet[33] và hàm mất Homography sau khi đã qua việc lọc nhiễu, trọng số  $w_{homo}(t)$  được thêm vào cơ chế học phân tầng (với hàm mất Homography được coi là nhiệm vụ ở giai đoạn thứ 4), có giá trị trong khoảng [0,0.2].

### 6.3.2 Chiến lược huấn luyện

Đầu tiên, tạo độ trễ bằng cách thêm hàm mất Homography sau 100 epoch, khi mà kết quả dự đoán độ sâu nhất quán và đáng tin cậy. Điều này sẽ giúp ma trận homography đủ tốt để không dẫn đến sai lệch, bởi vì độ sâu ở giai đoạn đầu huấn luyện sẽ không ổn định và cả các vị trí trong mặt phẳng Bird Eye's View cũng có độ tin cậy thấp.

Thứ hai, tạo thêm các hộp khác nhau bằng cách thay thế lần lượt các thành phần dùng để khôi phục tâm 3D (hình chiếu của tâm 3D  $\mathbf{q}$  và độ sâu  $d$ ) bằng giá trị thật từ nhãn trong khi thành phần còn lại vẫn là từ dự đoán. Do đó, hàm mất Homography được lặp lại ba lần. Cụ thể dựa vào các cặp giá trị sau để khôi phục tâm 3D:

- Hình chiếu của tâm 3D dự đoán và độ sâu dự đoán ( $\mathbf{q}_{pred}$  và  $d_{pred}$ )
- Hình chiếu của tâm 3D dự đoán và độ sâu thực tế ( $\mathbf{q}_{pred}$  và  $d_{gt}$ )
- Hình chiếu của tâm 3D thực tế và độ sâu dự đoán ( $\mathbf{q}_{gt}$  và  $d_{pred}$ )

Từ đó có được ba giá trị của hàm mất mát homography khác nhau. Hàm mất mát homography cuối cùng là kết hợp cả ba giá trị này bằng cách tính trung bình. Việc này góp phần làm cho ma trận Homography trở nên tin cậy hơn.

### 6.3.3 Thủ nghiệm

- Batch size: 16
  - Tỉ lệ lật ngược ảnh và cắt ảnh ngẫu nhiên là 0.5
  - Sử dụng optimizer là Adam với learning rate là 0.000625 và lần lượt giảm 90% khi qua các epoch 90 và 120.
  - Sử dụng chiến thuật warm up ở 5 epoch đầu tiên với learning rate là 0.00001 và tăng dần, để tạo sự ổn định trong quá trình huấn luyện.
  - Kết quả cuối cùng của mô hình được thử nghiệm trên tập dữ liệu KITTI 3D
  - Huấn luyện mô hình trên 2 GPU V100 và chạy với 140 epoch.
- Nhóm thử nghiệm **H-GUPNet** với hai phiên bản như mô hình cơ sở:
1. Huấn luyện **H-GUPNet** trên cả ba lớp (Car, Pedestrian, Cyclist).
  2. Huấn luyện **H-GUPNet** chỉ với mỗi lớp Car.
- Chi tiết về mã nguồn, có thể tham khảo tại link sau: [https://github.com/loiprocute/H\\_GUP-Net](https://github.com/loiprocute/H_GUP-Net).

# Chương 7

## Kết quả và đánh giá

### 7.1 Kết quả định lượng

Việc đánh giá hiệu suất phát hiện đối tượng bao gồm bốn loại điểm chuẩn : 2D detection, orientation, BEV detection, 3D detection trên ba lớp dữ liệu (Car, Pedestrian, Cyclist); các loại điểm chuẩn này được chia thành ba mức độ (Easy, Moderate, Hard). Trong đó nhóm chỉ tập trung vào hai loại điểm chuẩn là BEV detection, 3D detection . Đối với các đối tượng thuộc lớp Car độ trùng lặp hộp 3D được yêu cầu là 70%, trong khi Pedestrian, Cyclist là 50%.

Các bảng dưới đây đánh giá và so sánh **H-GUPNet** trên tập Validation KITTI(trong các bảng so sánh, các chỉ số tô đậm thể hiện kết quả tốt nhất) của hai phiên bản:

1. Bảng 7.1 so sánh kết quả với **H-GUPNet** với thiết kế cơ sở GUPNet trên tập Validation KITTI, gồm một loại điểm chuẩn (3D detection) của ba lớp Car, Pedestrian, Cyclist.
2. Bảng 7.2 so sánh kết quả với **H-GUPNet** với các phương pháp khác trên tập Validation KITTI, gồm hai loại điểm chuẩn (BEV detection, 3D detection) của lớp Car.
  - Đối với phiên bản huấn luyện mô hình trên cả ba lớp (Car, Pedestrian, Cyclist). So với kết quả thử nghiệm lại, H-GUPNet đạt được mức tăng 0.18%, 0.19%, 0.22% trên lớp Car và tăng 1.1%, 0.39%, 0.33% trên lớp Cyclist lần lượt trên các mức độ Easy, Moderate, Hard so với mô hình GUPNet mà nhóm đã thử nghiệm lại. Tuy nhiên ở lớp Pedestrian, kết quả lại giảm 0.85%, 0.46%, 0.22%.
  - Đối với phiên bản chỉ huấn luyện mô hình với mỗi lớp Car. So với kết quả mà tác giả cung cấp, H-GUPNet đạt được mức tăng 0.79%, 0.12%, 0.89% trên lớp Car với điểm chuẩn 3D detection; tăng 0.89%, 0.27%, 0.32% trên lớp Car với điểm chuẩn BEV detection.

Vì tính ngẫu nhiên ở mô hình cơ sở đã thảo luận ở phần 5.3, các số liệu nêu trên có thể chưa phải là tốt nhất.

Method	CarAP <sub>3D R<sub>40</sub></sub>			PedestrianAP <sub>3D R<sub>40</sub></sub>			CyclistAP <sub>3D R<sub>40</sub></sub>		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
GUPNet[33]	<b>23.17</b>	<b>16.23</b>	<b>13.56</b>	<b>11.47</b>	8.17	6.74	9.48	5.00	4.13
GUPNet(reproduce)	21.25	15.46	13.04	11.42	<b>8.73</b>	<b>7.01</b>	10.78	5.42	4.46
H-GUPNet(Ours)	21.43	15.65	13.26	10.57	8.27	6.79	<b>11.88</b>	<b>5.81</b>	<b>4.79</b>

Bảng 7.1: Hiệu suất phát hiện đối tượng 3D của 3 lớp trên tập Validation KITTI. (train với cả ba lớp). Số liệu của GUPNet được lấy từ checkpoint đã được công bố trên Github của tác giả.

Các số liệu tốt nhất được in đậm.

Method	3D @ IoU=0.7			BEV @ IoU=0.7			3D @ IoU=0.5			BEV @ IoU=0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
MonoGRNet[45]	11.90	7.56	5.76	19.72	12.81	10.15	47.59	32.28	25.50	48.53	35.94	28.59
M3D-RPN[2]	14.53	11.07	8.65	20.85	15.62	11.88	48.53	35.94	28.59	53.35	39.60	31.76
MonoPair[10]	16.28	12.30	10.42	24.12	18.17	15.76	55.38	42.39	37.99	61.06	47.63	41.92
GUPNet[33]	22.76	16.46	13.72	31.07	22.94	19.75	57.62	42.33	37.59	61.78	47.06	40.88
GUPNet(reproduce)	23.29	16.37	14.54	30.30	22.54	19.60	60.36	<b>45.94</b>	40.16	<b>67.09</b>	<b>50.18</b>	<b>43.99</b>
H-GUPNet(Ours)	<b>23.55</b>	<b>16.58</b>	<b>14.61</b>	<b>31.96</b>	<b>23.21</b>	<b>20.07</b>	<b>60.96</b>	44.99	<b>40.35</b>	66.85	50.10	43.94

Bảng 7.2: Hiệu suất phát hiện đối tượng 3D của lớp Car trên tập Validation KITTI khi chỉ huấn luyện với lớp Car. Các số liệu được lấy từ GUPNet[33]. Các số liệu tốt nhất được in đậm.

## 7.2 Kết quả định tính

Ở phần này, chúng em sẽ trình bày trực quan hóa một cách đa dạng kết quả của **H-GUPNet** được huấn luyện với ba lớp.

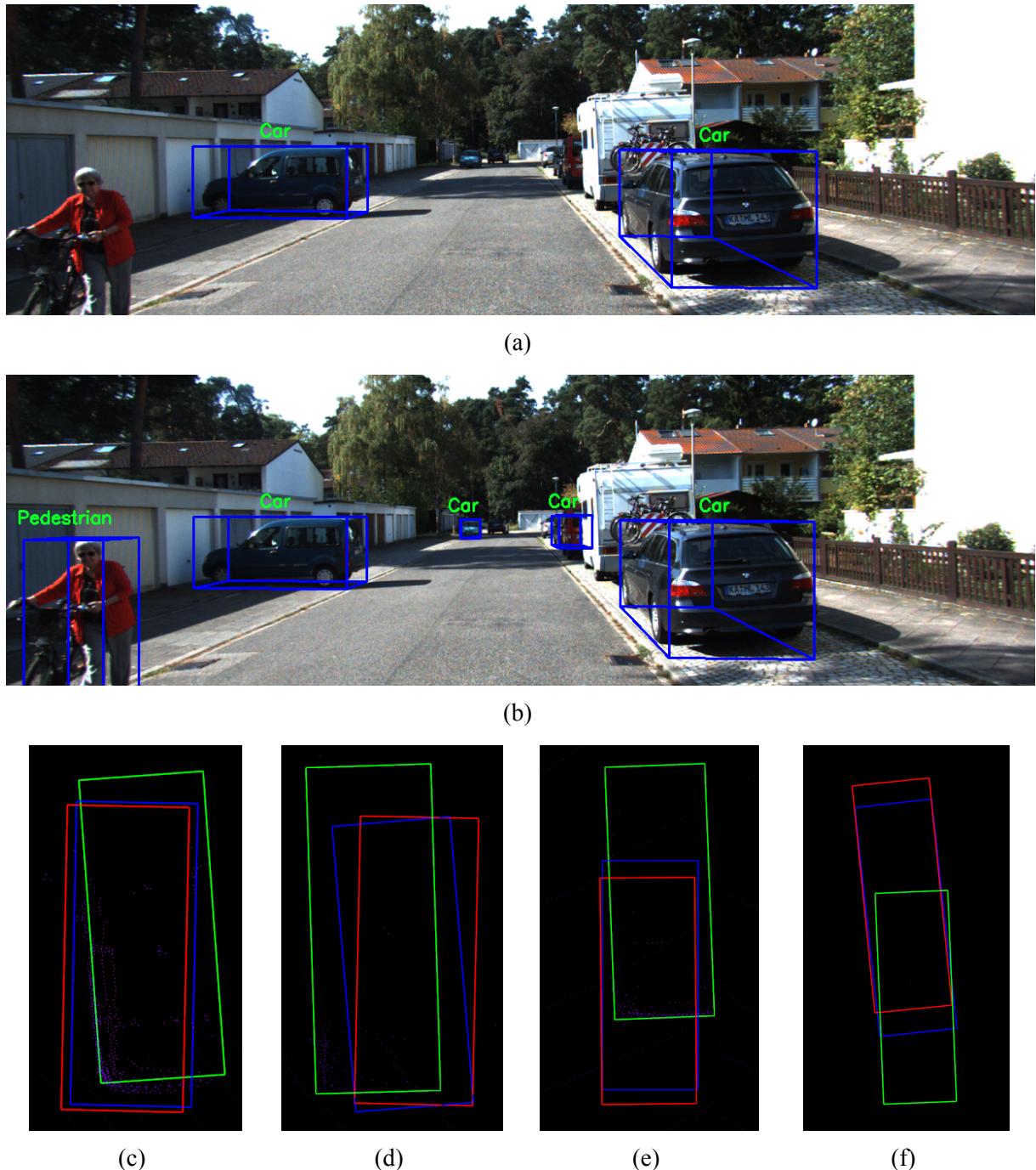
Hình 7.1 trực quan hóa tâm 2D qua đầu ra bản đồ nhiệt. Hình 7.2 minh họa việc dự đoán được một số trường hợp không được đánh nhãn. Ở hình gốc, có hai xe ở góc trái không được đánh nhãn khi huấn luyện, trong đó với xe màu tối là do KITTI đánh nhãn là ‘DontCare’ còn xe màu sáng bị bỏ qua trong quá trình tiền xử lý dữ liệu do bị che khuất phần lớn. Tuy nhiên cả mô hình cơ sở và **H-GUPNet** đều dự đoán được cả hai xe này. Tiếp đến, hình 7.3 trực quan hóa khả năng phát hiện tốt hơn của H-GUPNet khi so sánh với mô hình cơ sở. Ở đây, mô hình cơ sở ở hình 7.3a không dự đoán được cho người đi bộ bên góc trái và hai xe ở xa trong khi ở 7.3b thì H-GUPNet phát hiện được những vật thể này. Đồng thời, ở những hình 7.3c, 7.3d, 7.3e, 7.3f thể hiện được sự cải thiện về độ sai lệch của H-GUPNet khi so với nhãn. Có những trường hợp, khi quan sát kết quả phát hiện 3D qua ảnh 2D, ta sẽ không thấy được sự sai lệch nhưng khi quan sát trong không gian 3D thì sự sai lệch này thể hiện rõ hơn. Điều này là do sai lệch về hướng hoặc là độ sâu của vật thể, được minh họa ở hình 7.4. Đồng thời, khi quan sát trên không gian 3D ở 7.4, ta cũng thấy rõ được H-GUPNet có thể phát hiện được trường hợp không được đánh nhãn (hộp bao quanh màu đỏ đứng một mình trong hình).



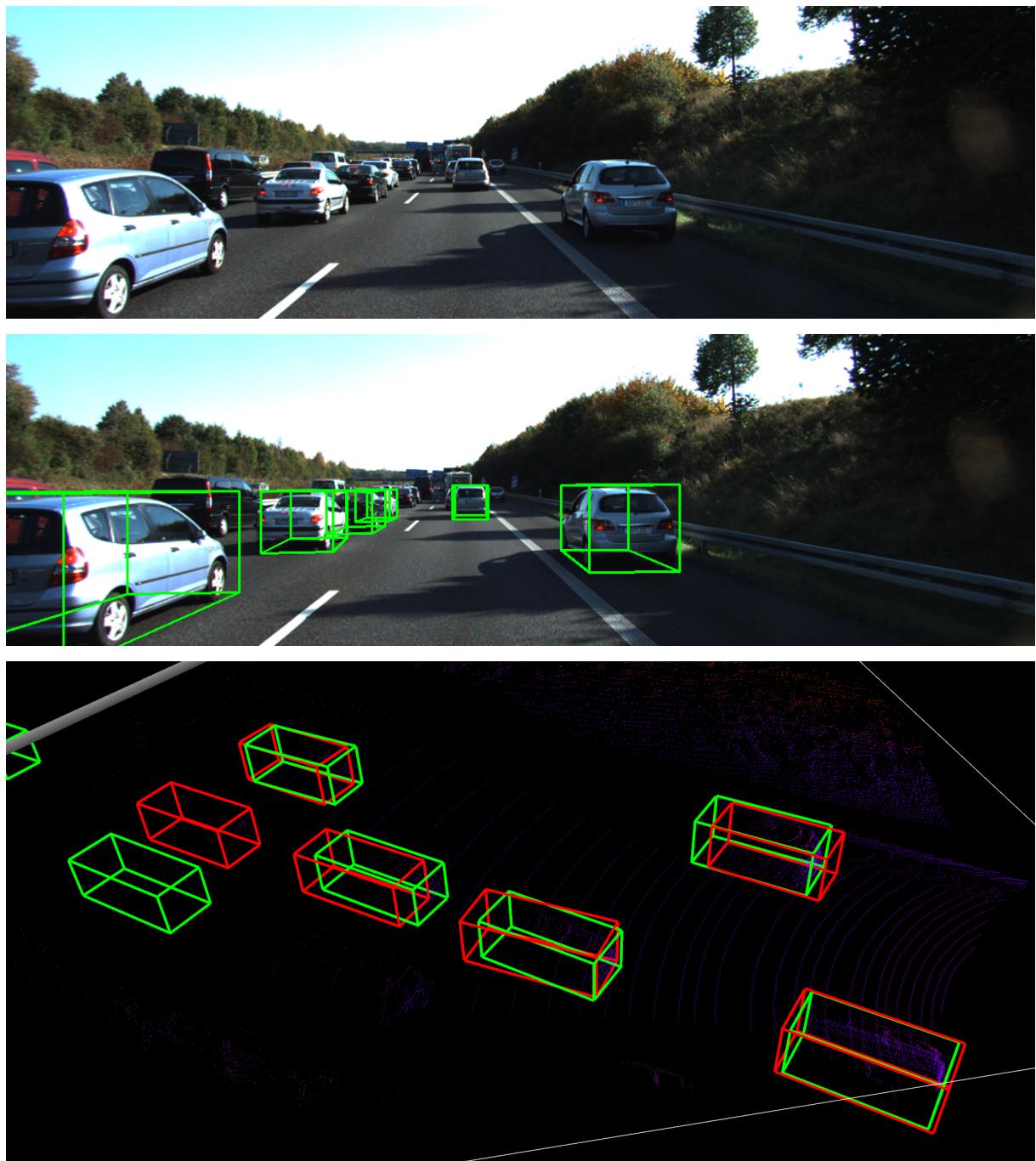
Hình 7.1: Trực quan hóa bản đồ nhiệt tâm 2D của vật thể. Bên trái là hình ảnh đơn từ tập kiểm thử của KITTI. Bên phải là bản đồ nhiệt của tâm 2D của vật thể trong hình tương ứng.



Hình 7.2: Minh họa về trường hợp không được đánh nhãn trong tập kiểm thử KITTI. Ảnh đầu tiên: ảnh gốc. Ảnh thứ hai: Nhãn thực tế của KITTI. Ảnh thứ ba: kết quả từ mô hình cơ sở. Ảnh cuối cùng: kết quả dự đoán của **H-GUPNet**.



Hình 7.3: Trực quan hóa cải thiện của H-GUPNet so với mô hình cơ sở. Hình a) và b) lần lượt là dự đoán của mô hình cơ sở và H-GUPNet trên hình ảnh. Hình c), d), e), f) thể hiện cải thiện sai lệch trên BEV. Kí hiệu: màu đỏ là mô hình cơ sở, màu xanh lá là nhãn, màu xanh dương là H-GUPNet



Hình 7.4: Trực quan hóa chi tiết kết quả phát hiện vật thể 3D. Ảnh 1: ảnh từ tập KITTI. Ảnh 2: kết quả trực quan hóa trên ảnh 2D. Ảnh 3: kết quả trực quan hóa trên không gian 3D. Lưu ý: ở hình 3, hộp màu xanh đại diện cho hộp từ nhãn thực tế, còn màu đỏ đại diện cho kết quả.

# Chương 8

## Tổng kết

### 8.1 Thành quả đạt được

Mục tiêu của nghiên cứu đề ra ban đầu là đề xuất một giải pháp phát hiện vật thể 3D trong hình ảnh đơn có thời gian xử lý ổn định và độ chính xác tương đối tốt hơn so với mô hình cơ sở nói riêng và những nghiên cứu khác trước đây nói chung. Thông qua việc hoàn thành các nhiệm vụ đã đề ra:

- Nghiên cứu cơ sở về bài toán phát hiện đối tượng 3D trong hình ảnh đơn.
- Khảo sát về các hướng tiếp cận và đánh giá.
- Trên cơ sở nghiên cứu và đánh giá trên, đề xuất một phương pháp phát hiện đối tượng 3D trong hình ảnh đơn với cải tiến phù hợp với mục tiêu nhóm đã đề ra.
- Phân tích và đánh giá kết quả của việc thí nghiệm phương pháp đã đề xuất.

Nhóm nhận thấy rằng mặc dù mình chưa đạt được hoàn toàn mục tiêu ban đầu nhưng cũng đã có một vài kết quả đáng lưu ý. Đối với phiên bản huấn luyện mô hình trên cả ba lớp (Car, Pedestrian, Cyclist). So với kết quả thử nghiệm lại, H-GUPNet đạt được mức tăng 0.18%, 0.19%, 0.22% trên lớp Car và tăng 1.1%, 0.39%, 0.33% trên lớp Cyclist lần lượt trên các mức độ Easy, Moderate, Hard so với mô hình GUPNet mà nhóm đã thử nghiệm lại. Tuy nhiên ở lớp Pedestrian, kết quả lại giảm 0.85%, 0.46%, 0.22%. Đối với phiên bản chỉ huấn luyện mô hình với mỗi lớp Car. So với kết quả mà tác giả cung cấp, H-GUPNet đạt được mức tăng 0.79%, 0.12%, 0.89% trên lớp Car với điểm chuẩn 3D detection; tăng 0.89%, 0.27%, 0.32% trên lớp Car với điểm chuẩn BEV detection. Tuy không thể thử nghiệm lại mô hình cơ sở với hiệu suất tương đương với số liệu của tác giả cung cấp, nhưng giải pháp đề xuất đã đem lại hiệu suất tốt hơn, đồng thời đáp ứng được mục tiêu về thời gian khi việc thêm vào hàm măt măt Homography không ảnh hưởng đến thời gian suy luận. Ngoài ra, vì lý do ngẫu nhiên số liệu này có thể vẫn còn cải thiện được.

### 8.2 Ý nghĩa

Nghiên cứu này đã có những đóng góp nhất định trong công cuộc phát triển của mảng Phát hiện vật thể 3D trong hình ảnh đơn. Những đánh giá và so sánh ở những trường hợp và mô hình khác nhau mang lại một góc nhìn tổng quát về điểm mạnh và điểm yếu của phương pháp, từ đó mở ra tiềm năng cho những cải thiện sau này. Ở góc nhìn thực tế, kết quả nghiên cứu này mang nhiều ý nghĩa đối với các ứng dụng trong nhiều mảng như hệ thống xe tự lái, rô bốt cầm nắm, đây là những mảng hiện đang được quan tâm. Việc cải thiện được độ chính xác và đảm

bảo được thời gian xử lý ổn định sẽ góp phần nâng cao hiệu suất cũng như độ an toàn cho các hệ thống trên.

### 8.3 Định hướng phát triển

Mặc dù nghiên cứu này đã đạt được kết quả tiến triển, thế nhưng trong mảng “Phát hiện vật thể 3D trong hình ảnh đơn” vẫn còn nhiều hướng để tiếp tục nghiên cứu trong tương lai. Đầu tiên là tìm hiểu về khả năng và quy mô của các mô hình trong việc xử lý nhiều lớp vật thể hơn nói riêng cũng như huấn luyện với các tập dữ liệu lớn và đa dạng hơn như Waymo Open[54] và nuScenes[4] nói chung, điều này sẽ làm tăng khả năng khai quát hóa của mô hình.Thêm vào đó là nghiên cứu về sự tích hợp của đa phương thức vào mô hình như việc tích hợp mạng ước tính độ sâu phụ trợ như đã nhắc đến ở phần 4.2 nhằm tối ưu hóa hiệu suất cũng như cải thiện dự đoán ở nhiều ngữ cảnh phức tạp hơn. Cuối cùng là phát triển khả năng diễn giải các thuật toán và nâng cao các kỹ thuật trực quan hóa vì sự tin cậy và minh bạch của các ứng dụng trong thế giới thực rất quan trọng.

# Tài liệu tham khảo

- [1] Deniz Beker, Hiroharu Kato, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Monocular differentiable rendering for self-supervised 3d object detection. *CoRR*, abs/2009.14524, 2020.
- [2] Garrick Brazil and Xiaoming Liu. M3D-RPN: monocular 3d region proposal network for object detection. *CoRR*, abs/1907.06038, 2019.
- [3] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. *CoRR*, abs/2007.09548, 2020.
- [4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CoRR*, abs/1903.11027, 2019.
- [5] Yingjie Cai, Buyu Li, Zeyu Jiao, Hongsheng Li, Xingyu Zeng, and Xiaogang Wang. Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. *CoRR*, abs/2002.01619, 2020.
- [6] Julie Chang and Gordon Wetzstein. Deep optics for monocular depth estimation and 3d object detection. pages 10192–10201, 2019.
- [7] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. *CoRR*, abs/1911.02620, 2019.
- [8] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. *CoRR*, abs/2103.12605, 2021.
- [9] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. pages 2147–2156, 2016.
- [10] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. *CoRR*, abs/2003.00504, 2020.
- [11] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. *CoRR*, abs/1912.04799, 2019.
- [12] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. pages 2002–2011, 2018.
- [13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.
- [14] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CoRR*, abs/1609.03677, 2016.
- [15] Jiaqi Gu, Bojian Wu, Lubin Fan, Jianqiang Huang, Shen Cao, Zhiyu Xiang, and Xian-Sheng Hua. Homography loss for monocular 3d object detection. 2022.

- [16] Chenhang He, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Aug3d-rpn: Improving monocular 3d object detection by synthetic images with virtual depth. *CoRR*, abs/2107.13269, 2021.
- [17] Yang He, Wenyuan Tao, and Chung-Ming Own. Dfr-net: Learning dense features at the resolution level. *IEEE Access*, 7:97013–97020, 2019.
- [18] Jonas Heylen, Mark De Wolf, Bruno Dawagne, Marc Proesmans, Luc Van Gool, Wim Abbeloos, Hazem Abdelkawy, and Daniel Olmeda Reino. Monocinis: Camera independent monocular 3d object detection using instance segmentation. *CoRR*, abs/2110.00464, 2021.
- [19] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krähenbühl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. *CoRR*, abs/1811.10742, 2018.
- [20] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Level 5 perception dataset 2020. <https://level-5.global/level5/data/>, 2019.
- [21] Jason Ku, Alex D. Pon, and Steven L. Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. *CoRR*, abs/1904.01690, 2019.
- [22] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. Groomed-nms: Grouped mathematically differentiable NMS for monocular 3d object detection. *CoRR*, abs/2103.17202, 2021.
- [23] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. GS3D: an efficient 3d object detection framework for autonomous driving. *CoRR*, abs/1903.10955, 2019.
- [24] Peixuan Li. Monocular 3d detection with geometric constraints embedding and semi-supervised training. *CoRR*, abs/2009.00764, 2020.
- [25] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. RTM3D: real-time monocular 3d detection from object keypoints for autonomous driving. *CoRR*, abs/2001.03343, 2020.
- [26] Yingyan Li, Yuntao Chen, Jiawei He, and Zhaoxiang Zhang. Densely constrained depth estimator for monocular 3d object detection, 2022.
- [27] Zhuoling Li, Zhan Qu, Yang Zhou, Jianzhuang Liu, Haoqian Wang, and Lihui Jiang. Diversity matters: Fully exploiting depth clues for reliable monocular 3d object detection, 2022.
- [28] Qing Lian, Botao Ye, Ruijia Xu, Weilong Yao, and Tong Zhang. Exploring geometric consistency for monocular 3d object detection. pages 1675–1684, 2022.
- [29] Xianpeng Liu, Nan Xue, and Tianfu Wu. Learning auxiliary monocular contexts helps monocular 3d object detection. *CoRR*, abs/2112.04628, 2021.
- [30] Yuxuan Liu, Yixuan Yuan, and Ming Liu. Ground-aware monocular 3d object detection for autonomous driving. *CoRR*, abs/2102.00690, 2021.
- [31] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: single-stage monocular 3d object detection via keypoint estimation. *CoRR*, abs/2002.10111, 2020.
- [32] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. *CoRR*, abs/2108.11127, 2021.
- [33] Yan Lu, Xinzu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. *CoRR*, abs/2107.13774, 2021.
- [34] Shujie Luo, Hang Dai, Ling Shao, and Yong Ding. M3DSSD: monocular 3d single stage object detector. *CoRR*, abs/2103.13164, 2021.

- [35] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. *CoRR*, abs/2008.04582, 2020.
- [36] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3d object detection from images for autonomous driving: A survey. *CoRR*, abs/2202.02980, 2022.
- [37] Xinzhu Ma, Zhihui Wang, Haojie Li, Wanli Ouyang, and Pengbo Zhang. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. *CoRR*, abs/1903.11444, 2019.
- [38] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. *CoRR*, abs/2103.16237, 2021.
- [39] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. ROI-10D: monocular lifting of 2d detection to 6d pose and metric shape. *CoRR*, abs/1812.02781, 2018.
- [40] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A review and new outlooks, 2022.
- [41] Andretti Naiden, Vlad Paunescu, Gyeongmo Kim, ByeongMoon Jeon, and Marius Leordeanu. Shift R-CNN: deep monocular 3d object detection with closed-form geometric constraints. *CoRR*, abs/1905.09970, 2019.
- [42] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? *CoRR*, abs/2108.06417, 2021.
- [43] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The H3D dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. *CoRR*, abs/1903.01568, 2019.
- [44] Liang Peng, Xiaopei Wu, Zheng Yang, Haifeng Liu, and Deng Cai. Did-m3d: Decoupling instance depth for monocular 3d object detection, 2022.
- [45] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. *CoRR*, abs/1811.10247, 2018.
- [46] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution network for monocular 3d object detection. *CoRR*, abs/2103.01100, 2021.
- [47] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.
- [48] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *CoRR*, abs/1811.08188, 2018.
- [49] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Imvoxelnet: Image to voxels projection for monocular and multi-view general-purpose 3d object detection. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1265–1274, 2022.
- [50] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. *CoRR*, abs/2104.03775, 2021.
- [51] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. *CoRR*, abs/2104.03775, 2021.
- [52] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kortschieder. Disentangling monocular 3d object detection. *CoRR*, abs/1905.12365, 2019.

- [53] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kortschieder. Single-stage monocular 3d object detection with virtual cameras. *CoRR*, abs/1912.08035, 2019.
- [54] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *CoRR*, abs/1912.04838, 2019.
- [55] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. *CoRR*, abs/1904.01355, 2019.
- [56] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3d object detection. *CoRR*, abs/2103.16470, 2021.
- [57] Li Wang, Li Zhang, Yi Zhu, Zhi Zhang, Tong He, Mu Li, and Xiangyang Xue. Progressive coordinate transforms for monocular 3d object detection. 34:13364–13377, 2021.
- [58] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. FCOS3D: fully convolutional one-stage monocular 3d object detection. *CoRR*, abs/2104.10956, 2021.
- [59] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. *CoRR*, abs/2107.14160, 2021.
- [60] Xinlong Wang, Wei Yin, Tao Kong, Yuning Jiang, Lei Li, and Chunhua Shen. Task-aware monocular depth estimation for 3d object detection. *CoRR*, abs/1909.07701, 2019.
- [61] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9:36–45, January 1966.
- [62] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. *CoRR*, abs/1903.09847, 2019.
- [63] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. *CoRR*, abs/1903.09847, 2019.
- [64] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2345–2353, 2018.
- [65] Xiaoqing Ye, Liang Du, Yifeng Shi, Yingying Li, Xiao Tan, Jianfeng Feng, Errui Ding, and Shilei Wen. Monocular 3d object detection via feature domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 17–34, Cham, 2020. Springer International Publishing.
- [66] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. *CoRR*, abs/2104.02323, 2021.
- [67] Zuoxi Zhao, Yuchang Zhu, Yuanhong Li, Zhi Qiu, Yangfan Luo, Chaoshi Xie, and Zhuangzhuang Zhang. Multi-camera-based universal measurement method for 6-dof of rigid bodies in world coordinate system. *Sensors*, 20(19), 2020.
- [68] Xichuan Zhou, Yicong Peng, Chunqiao Long, Fengbo Ren, and Cong Shi. Monet3d: Towards accurate monocular 3d object localization in real time. *CoRR*, abs/2006.16007, 2020.
- [69] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019.

- [70] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monoef: Extrinsic parameter free monocular 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10114–10128, 2022.