

動的に物体検知方法を切り替えるソフトウェアアーキテクチャに関する研究

2019SE026 近藤友樹 2019SE058 寺澤佳紀

指導教員：野呂昌満

1 はじめに

自動運転において物体検知は盛んに研究が行われている。物体検知の研究で重要視されているのは精度と実行時間である。センサフュージョン技術は複数のセンサを用いることで、物体検知の精度を向上させる方法である。センサフュージョンを行うとデータの統合処理や段階的なニューラルネットワーク（以下 NN と省略）の使用によって実行時間が長くなってしまふ。すなわち、二つの非機能特性をともに向上させることは困難である。以上の問題から、物体検知における精度と実行時間は相反する非機能特性であると考えられる。

既存の物体検知の研究では、一つの物体検知方式のみを用いて実験を行っている。すなわち、静的な物体検知を行う研究がなされている。静的な物体検知では、一つの物体検知方式・センシングデバイスの組み合わせでしか動作させることができない。実際に静的な物体検知を用いた場合、用いるセンシングデバイス・物体検知方式が苦手とする走行状況になった時に精度や実行時間の問題から事故を起こすことが考えられる。この問題に対し、動的に物体検知方法を切り替える構造にすることでセンシングデバイス・センサフュージョンの方法を可変化することで、あらゆる走行状況に対応できるようにすることができると考える。走行状況の変化に合わせて最適な物体検知方式に切り替えることにより、物体検知の精度と実行時間の相反する非機能特性の衝突が回避できると考える。

このように、相反する非機能要求の衝突回避を目的とする。動的な切り替え構造を用いた物体検知方式に使用するソフトウェア、ハードウェアコンポーネントの関連を明らかにして、物体検知システムの基盤となるアーキテクチャを提案することによって目的を達成できると考える。

上記の研究目的を具体化すると以下が研究課題になると考える。

課題 1 動的に物体検知方法を切り替えるアーキテクチャを定義する

課題 2 プロトタイプを実装し、提案するアーキテクチャの有用性を実験科学的アプローチから定量的に評価する

走行状況の変化に応じて物体検知方式の切り替えを行うために、コンテキスト指向を取り入れる。コンテキスト指向に基づくアーキテクチャを定義することで動的センサフュージョンを実現できると考える。本研究では、走行環

境として自車速度をコンテキストとして導入し、選択は決定的に行う。提案するアーキテクチャが目的に対して有効であるかはアーキテクチャにもとづくプロトタイプを作成して、実行時間と精度について定量的な評価を行う。

2 関連研究

2.1 センサフュージョン技術

物体検知の精度を向上させる方法としてセンサフュージョン技術があり、近年の研究では NN を用いたセンサフュージョンが主流である。NN を用いたセンサフュージョンの方法は主に Early-fusion[4], Late-fusion[4], Joint-fusion[1] がある。Early-fusion とは得られた環境データをそのまま統合するセンサフュージョンである。センサフュージョンの中でも実行速度は速いが、統合の際に失われる情報が多い。Late-fusion とは、得られた環境データに対し、それに適する NN を用いて周辺環境モデルを作成し、作成したモデルを統合するセンサフュージョンである。センサフュージョンの中でも精度は高いが、時間同期が難しいという問題がある。Joint-fusion は、複数のセンシングデータからニューラルネットワークを用いて特徴量を抽出し、抽出した特徴量を統合することで 1 つのデータを作成する方式である。Early-fusion に比べて情報の消失が少なく、Late-fusion に比べて時間同期が必要となる部分が少ない。

2.2 ニューラルネットワークを用いた物体検知

Joint-fusion を用いた物体検知として BEVFusion[1] がある。LiDAR の点群データとカメラの RGB 画像データの特徴量を統合し、その後新たに得た特徴量からモデルを作成することで、鳥瞰図で精度の高い物体検知を実現している。BEVFusion ではカメラの RGB 画像データから特徴量抽出を行うために Swin-Transformer[3], LiDAR の点群データから特徴量抽出を行うために VoxelNet[2] を利用している。特徴量を統合する際は、それぞれの特徴量を鳥瞰図に直した後、統合を行う。

2.3 動的な NN の切り替え

水谷ら [7] は偽造画像検出を行うために複数の NN で構成されるアーキテクチャを提案している。造画像を検出する NN を動的に選択するためにメタレベルアーキテクチャをもとに設計している。具体的には、偽造検知を行う NN の出力結果をコンテキストとし、NN_selector がコンテキストをもとに NN を選択する構造となっている。我々は、水谷らの動的な切り替え構造をセンシングデバイス・物体

検知方式の切り替えへ応用する。

3 課題解決へのアプローチ及び評価方法

3.1 課題解決へのアプローチ

本研究では複数のセンサを協調させるために動的センサフュージョンを行う。動的センサフュージョンとは、センシングデバイス・物体検知方式をあらかじめ準備しておき、走行状況によってそれを切り替えることである。動的な物体検知方式の切り替えは、水谷らの研究をもとにアーキテクチャを定義することで可能とさせる。切り替える物体検知方式の対象はセンサフュージョンを行う物体検知方式と単一センサでの物体検知方式とした。センサフュージョンを行う場合は Lidar とカメラを用い、単一センサでは Lidar のみを用いる。その理由は第 3 節 3.2 で述べる。提案したアーキテクチャはプロトタイプを用いて実験し、定量的に議論する。センサフュージョンを用いた物体検知、単一センサでの物体検知、プロトタイプをもとにした物体検知の三つにおいて精度と実行時間を比較する。加えて、動的な物体検知方式の切り替えによる実行時間の増加がシステムに与える影響を検証する。

現在、自動運転は盛んに研究がされている。新しい技術や新たに考えるべき走行状況ができたときに、センシングデバイスや物体検知方式の追加・変更が容易に行える必要がある。このことから、本研究では、物体検知アーキテクチャに柔軟性・拡張性を持たせたいと考える。Bucaioni ら [6] の研究では、機能別にコンポーネントを分け、それらを疎結合で連携させることによってアーキテクチャに柔軟性・適応性・保守性を持たせている。本研究では Bucaioni らの考えをもとにアーキテクチャを提案することで柔軟性・拡張性を持たせる。柔軟性・拡張性を定量的に測ることは困難なので、予測されるアーキテクチャの変更例を用いて定性的な議論を行う。

3.2 前提条件

本研究における前提条件は以下とする。

- センシングデバイスとしてカメラと Lidar を用いる
- 晴天時のデータを使用する
- センサフュージョンアルゴリズムの切り替えは自車速度 30km を基準とする。

Lidar の代わりにミリ波レーダでも物体を検知することが可能だが、ミリ波レーダを単一で物体検知に用いた場合、近距離の物体の検知精度が落ちてしまう。二つの物体検知方式において精度を比較するので、単一で用いても距離によって精度の差が生じにくい Lidar を用いる。また、物体検知の研究がされるときはカメラと Lidar が用いられることが多いので、本研究では Lidar とカメラを用いる。Lidar とカメラは悪天候に弱く精度の比較に影響があると考えられるので、本研究では晴天時のみのデータを用いる。

本研究の目的から、動的に物体検知方式を切り替える構造が実験によって有効であることを検証するために、自

車速度のみを用いて切り替えを行う。制限速度と道の危険度には関連性がある。生活道路（地域住民の日常生活に利用される道路）のような狭い道では一般的に制限速度は 30km/h である。このような道では、事故の原因である危険因子が多数存在するので精度の高い物体検知が必要となる。以上の議論から切り替え基準として自車速度 30km/h とする。

4 アーキテクチャ

本研究では自車速度をコンテキストとし、それをもとに Fusion_selector で物体検知方式を選択する構造にすることで、動的に物体検知方式を切り替える物体検知アーキテクチャを作成した。本研究で提案する動的に物体検知方式を切り替えるアーキテクチャを図 1 に示す。このアーキテクチャでは、センサフュージョンを用いて検知する方式と単一センサで検知する方式を使用している。Bucaioni らの考えを応用し、提案するアーキテクチャではコンポーネントをセンサ、コンテキスト、物体検知器、Fusion_selector に分割した。それぞれの物体検知方式は、新たな物体検知方式の追加において変更容易性を持たせるために物体検知器と is-a 関係で結んでいる。それぞれの物体検知方式において、CNN などの機能を物体検知方式と has-a 関係で結んでいる。新たに物体検知方式を追加する際に、CNN やバウンディングボックス作成のためのコンポーネントが同一であれば再利用が可能であると考えられる。センサフュージョン技術は複数のセンシングデータを利用する。複数のセンシングデータを一括で取得するためには、デザインパターンとしてコンポジットパターンが相応しいと考えた。コンポジット_センサがあることによって一つのメソッドで必要なセンシングデータを取得することができると考える。

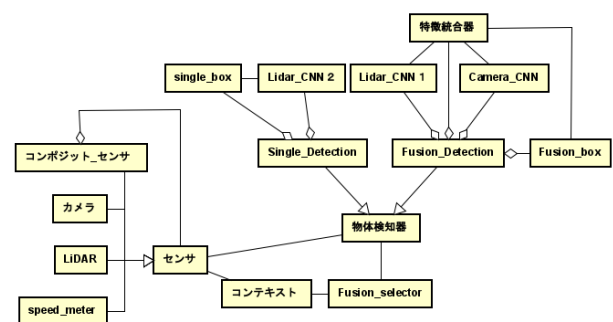


図 1 アーキテクチャ

図 1 に示すアーキテクチャのコンポーネントは以下の様な役割をもつ。

- コンテキスト： 走行環境情報を持つ
- 物体検知器： 周辺状況のモデルを作成する
- センサ： 外部環境データを取得する
- Fusion_selector： 物体検知方式を選択する

5 プロトタイプの実装

センサフュージョンを行う物体検知方式として BEVFusion を用いた。BEVFusion は Joint-fusion を用いた物体検知方式として優秀なので用いた。単一センサでの物体検知方式として VoxelNet を用いた。VoxelNet は BEVFusion において Lidar の点群データから特徴量を抽出するために使用されている。それぞれの物体検知方式における点群データの処理に差を生じさせないために同じニューラルネットワークを用いた。

プロトタイプのコンテキストは限られた時間で実験を行うためにコンテキスト指向プログラミングを行わず、条件分岐を用いることで、動的な物体検知方式の選択を実現した。アーキテクチャ設計をもとに作成したプロトタイプを図2に示す。このプロトタイプのコミュニケーション図を図3に示す。

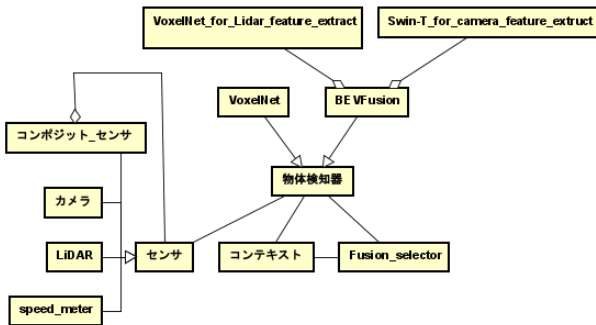


図2 プロトタイプ

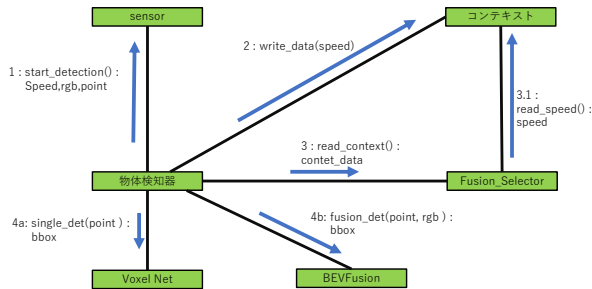


図3 コミュニケーション図

作成したプロトタイプでは、物体検知器によって4つの処理が制御されている。

- (1) 物体検知器がセンサからセンシングデータのスピードデータ、画像データ、点群データを受け取る。
- (2) 走行状況としてスピードデータをコンテキストに書き込む。
- (3) 書き込まれたコンテキストの情報を Fusion_selector が読むことで、物体検知方式を決定する。
- (4) センサフュージョンによる物体検知を行う場合は、Fusion_detection (BEVFusion) に画像データと点群デー

タの両方を送信する。単一センサによる物体検知を行う場合は、Single_detection (VoxelNet) に点群データのみを送信する。それぞれの物体検知の結果として Bounding Box を得る。

本研究では、プロトタイプの物体検知方式が利用するカメラの RGB 画像データと LiDAR の点群データがあること、他のデータセットと比較してデータ数が多いことから実験において nuScenes オープンデータセット [5] を用いた。

6 評価と考察

6.1 相反する衝突の回避

実験によって、センサフュージョンを用いた物体検知、単一センサでの物体検知の精度と実行時間を得た。加えて、動的な物体検知方式の切り替えるにかかる時間を算出した。動的な切り替え構造が目的に対して有効であら確認できれば良いので、図3の4aと4bは導入せず、プロトタイプをもとにした物体検知の精度と実行時間は物体検知方式の切り替え割合から算出した。

実験によって得られた精度を表1に示す。mAP (意味平均適合率) は物体検知で分類するすべての種類の正答率を平均したものである。実験によって得られた実行時間を表2に示す。加えて、実行時間から算出される空走距離を表3に示す。

表1 物体検知の精度 (mAP)

	BEVFusion	VoxelNet	選択割合から予測される精度
mAP (%)	68.57	64.7	67.86

表2 実行時間

計測したモデル	平均の実行時間 (ms/img)
BEVFusion	281.69
VoxelNet	175.43
selector + 選択割合から予測される実行時間	262.74
selector	0.51

表3 空走距離

速度 (km/h)	BEVFusionによる空走距離 (m)	VoxelNetによる空走距離 (m)	選択器による空走距離 (m)	プロトタイプにおける空走距離 (m)
10	0.782	0.487	0.001	0.784
20	1.565	0.975	0.003	1.568
30	2.347	1.462	0.004	1.466
40	3.130	1.949	0.006	1.955
50	3.912	2.437	0.007	2.444
60	4.695	2.924	0.009	2.932
70	5.477	3.411	0.010	3.421
80	6.260	3.898	0.011	3.910
90	7.042	4.386	0.013	4.399
100	7.825	4.873	0.014	4.887
110	8.607	5.360	0.016	5.376
120	9.390	5.848	0.017	5.865

表3から選択の時間による空走距離は、物体検知による空走距離に比べて無視できるほど小さいとわかる。このことから、動的な物体検知方式の切り替えによる実行時間の増加はほとんど影響がないと考えられる。高速度域では、センサフュージョンを行う物体検知による空走距離の増加が大きくなっていることがわかる。実行時間が空走距離に与える影響の少ない低速度域のみでセンサフュージョンによる物体検知を行うことで、空走距離の増加を抑えながら精度を重視した物体検知ができたと考える。

表1, 表2より, 予測される精度は単一センサでの物体検知より良くなり, 予測される実行時間はセンサフュージョンを用いた物体検知よりも短くなった。このことから, 提案したアーキテクチャは相反する非機能特性の衝突の回避に有効であると検証できた。提案したアーキテクチャを用いて物体検知システムを作成すれば, 他の相反する非機能特性の衝突が起きる状況にも対応できる。その一例として天候の変化がある。悪天候に強いセンシングデバイス・物体検知方式を物体検知システムに導入しておくことで, 天候が変化した時に動的に最適のセンシングデバイス・物体検知方式が選択されるので, 必要となる精度・実行時間が保証できると考える。

本研究では, 予測される精度・実行時間はセンサフュージョンを行う物体検知に寄った。これは nuScenes の走行データにおいて 30km/h 以下の速度が多かったからであると考えられる。選択基準を変えることやコンテキストの情報を追加することで, この偏りは変化すると考える。

6.2 アーキテクチャの柔軟性・拡張性

提案したアーキテクチャにおいて新たにセンシングデバイスを追加する場合, センサと追加するセンシングデバイスを is-a 関係で結ぶ。追加を行ってもセンシングデバイス間に関連はできないので, 他のセンシングデバイスには影響を与えない。物体検知方式を追加する場合も同様に, 物体検知器と追加する物体検知方式を is-a 関係で結ぶだけなので, 物体検知方式の追加が他の物体検知方式に影響を与えない。以上の議論から, センシングデバイス・物体検知方式の変更は容易であると考えられる。コンテキストに情報を追加する場合, センシングデバイスや物体検知方式の追加が考えられる。先の議論から, センシングデバイスの追加や物体検知方式の追加は容易に行えるのでコンテキストに情報を追加する際のアーキテクチャの変更は容易である。Fusion_selector はすべての変更に対して影響を受ける。その際, Fusion_selector に物体検知方式の選択肢の追加とその選択基準を追加するだけで変更が可能である。以上の議論から, アーキテクチャ全体において変更容易性があるので, 提案するアーキテクチャには柔軟性・拡張性を持たせることができたと考える。

7 おわりに

本研究では, 物体検知における相反する非機能特性の衝突を回避することを目的とし, 目的を達成するために動的に物体検知方式を切り替えるアーキテクチャを定義した。定義したアーキテクチャが目的に対して有効であることを確かめるためにプロトタイプを作成し実験・評価を行った結果, 有効であると判断できた。今後の課題は以下である。

1. 新たなセンシングデバイスの追加
2. 新たな物体検知方式の追加
3. 動的な切り替え構造の LSTM などの走行判断器への拡張
4. コンテキスト内容の増加

参考文献

- [1] Z.Liu, H.Tang, A.Amini, X.Yang, S.Jiao, H.Mao, D.Rus, S.Han, “BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird’s-Eye View Representation,” *arXiv*, 2022.
- [2] Y.Zhou and O.Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” *CVPR*, 2018.
- [3] Z.Liu, Y.Lin, Y.Cao, H.Hu, Y.Wei, Z.Zhang, S.Lin, and B.Guo, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” *ICCV*, 2021.
- [4] R.Zhang, S.A.Candra, K.Vetter, A.Zakhori, “Sensor Fusion for Semantic Segmentation of Urban Scenes,” *IEEE ICRA*, 2015.
- [5] H.Caesar, V.Bankiti, A.H.Lang, S.Vora, V.E.Liong, Q.Xu, A.Krishnan, Y.Pan, G.Baldan, and O.Beijbom, “nuScenes: A Multimodal Dataset for Autonomous Driving,” *CVPR*, 2020.
- [6] A.Bucaioni, P.Pelliccione, “Technical Architectures for Automotive Systems,” *IEEE ICASA*, 2020.
- [7] A.Mizutani, M.Noro, and A.Sawada, “Design of Software Architecture for Neural Network Cooperation: Case of Forgery Detection,” *Proc. 28th APSEC*, 2021.