自動運転向けマルチセンサフュージョン処理プラットフォームの開発

福元和真[†1] 福島悠史[†1] 成沢文雄[†1]

アブストラクト： 自動運転技術の高度化に伴い，車両周囲の交通状況の認識には多数のセンサデータを総合的に解析し認識精度を高めるセンサフュージョン技術が自動運転制御に不可欠となっている．

多種多様な複数のセンサデータを扱うフュージョン技術は，個別の異なるデータを入力しそれらを統合した認識結果を出力するものであるため，センサの仕様変更への対応と同時に統一的なデータ処理の実現が課題であり，そのソフトウェアの開発・修正には多大な工数を要していた．

提案手法では，各センサ固有のデータと共通化した認識結果を分離した階層構造を導入し階層間で異なるデータを統一的に処理可能なアーキテクチャとデータ構造を定義しデータの扱いを簡単化することで仕様変更時の影響伝搬を抑制し開発工数低減を実現する．

The Development of Multi Sensor Fusion Processing for Autonomous Driving

Kazuma Fukumoto[†1], Yuji Fukushima[†1], Fumio Narisawa[†1]

Abstract: Along with the advancement of autonomous driving technology, the sensor fusion technology, which use much data and comprehensively analyzed to recognize the traffic situation around the vehicle, is indispensable for realizing the autonomous driving.

Sensor fusion technology, however, need to input different type of sensor data and output a recognition result created by integrating these data. Therefore, realizing a unified data processing and responding to sensor specification changes was a problem, and it took a lot of effort to develop and modify the software.

In the proposed method, we introduce a hierarchical data structure that specific to each sensor and common recognition results. In addition, we define communication protocols that can uniformly process different data among a hierarchical structure. This reduces the influence at the time of system change and reduces development effort.

1. はじめに

現在，システムが人間を介さず全ての運転操作を担う完全自動運転に向けた研究開発が進んでいる．このような自動運転の実現のため，車両には 10 以上のセンサが取り付けられ自車の周囲の情報をセンシングし，走行環境の理解や制御量の演算に用いられる．

複数のセンサデータを一度に扱う技術として，センサフュージョンが広く研究されている．センサフュージョンの目的は，複数のセンサデータを統合し，単一のセンサでは不十分な認識精度やセンシング出来ない情報を補間する事にある．センサフュージョンは様々な産業で導入が進んでおり，3 レイヤ構成のセンサフュージョンアーキテクチャに関する研究が複数報告されている[2][5].

自動車分野においても導入されつつあるが，一つの車両に異なる複数のベンダのセンサを用いるなどの理由でレイヤ毎に扱うデータのフォーマットが異なり，フュージョン処理が複雑となるため，異なるレイヤ間で容易にデータを授受する仕組みが必要である．

また，車両に取り付けられるセンサ毎に扱うデータのフォーマット，単位，座標系等が異なり，従来はエンジニアが手作業で補正している．加えて，センサの仕様変更の都度，ソフトウェアへ変更が発生し，工数増加の要因となっているため，工数削減が望まれている．

本論文は，センサフュージョンアーキテクチャとデータフローに着目し，異なるレイヤ間のデータ授受を容易

†1 日立オートモティブシステムズ（株）

化する仕組みを設け，マルチレイヤのフュージョン処理に適した自動運転向けプラットフォームを提案する．

2. 関連研究

これまで，様々な分野でセンサフュージョンのアーキテクチャが提案されているが，特に，Low-level (raw-data fusion)，Mid-level (feature fusion)，High-level (decision fusion)で構成される 3 レイヤのアーキテクチャが複数提案されている [3][4].

Low-level (raw-data fusion)処理では，センサから取得した画像データなどの raw データに対し，特徴量等を取り出し，単一のセンサ空間で統合することで，オブジェクト等，より意味のあるデータを生成する処理を行う．

Mid-level (feature fusion) 処理では，物体までの距離情報等を用い，各センサ空間のオブジェクトデータを統合し，認識精度を高める．

High-level (decision fusion) 処理では，カメラ等のセンサで取得された動的に変化するオブジェクトデータと地図情報のような静的なデータを統合することで，検知精度のさらなる向上と共に，周囲の状況把握を行う．

既存のモデルは，異なるレイヤ間のデータの授受まで詳細に考慮していないため，実装に落とした際，異なるレイヤ間のデータ構造不一致の課題がある．

Darms らは自動運転を対象とし，Esteban らのモデルを参考にした 3 レイヤ構成のセンサフュージョンシステムを報告している [2]. 彼らは，自動運転の主な処理を担う知能系 ECU に複数のセンサを接続し，Low-level 処理から High-level 処理までを知能系 ECU で行った．彼らの手法は，システム内の全てのセンサの仕様が固定という仮定の下，raw データを知能系 ECU に直接入力している．しかし，実際の開発では仕様変更が頻繁に入るため，変更の都度データ構造等を修正し仕様に適合する必要がある．また，複数レイヤ構成のアーキテクチャの場合，レイヤ間でデータのやり取りを行うため，下位レイヤの仕様変更が上位レイヤに伝搬する．そのため，センサの仕様変更時は，上位のレイヤにも修正・変更が発生する課題がある．

3. 提案手法

3.1. システム構成

本論文では，センサフュージョンアーキテクチャとそのデータフローに着目した異なるレイヤ間のデータ授受を容易にする仕組み、マルチレイヤのフュージョン処理に適した自動運転向けプラットフォームを提案する．

本論文は図 1 に示す通り，ステレオカメラ，レーダ，Lidar，単眼カメラ，高精度地図ユニット，GPS ロケータなどのセンサとそれらのデータを使い認知・制御などの処理を行う知能系 ECU で構成された自動運転システムを対象とする．

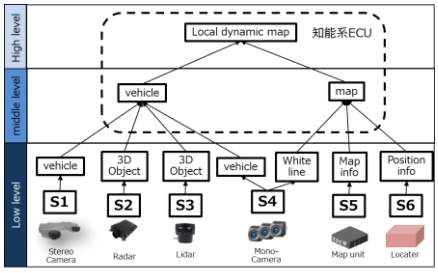

図 1 全体構成

センサフュージョンは大部分の処理を知能系 ECU で行うが ECU の処理性能の課題により，Low-level 処理に該当する物体検出等はセンサ内で行われることが多い．本論文でも Low-level 処理はセンサで行う．

そして，知能系 ECU 内の Middle-level 処理で各センサから取得した動的なオブジェクトデータや地図データ等の静的なオブジェクトデータをそれぞれ統合する．統合には，自車と物体間の相対距離，自車と物体との相対速度を用い，これらの差が一定の閾値以内の場合は同じ物体であると決定する．次に，High-level 処理にて，前レイヤまでの統合処理から推定した自車位置と自車位置の周囲の状況を統合する．

3.2. レイヤ間のデータ授受における課題

複数レイヤで構成されるセンサフュージョンでは，異なるセンサ情報を統合するため，下記違いが課題となり，容易に統合処理が行えない．

A) センサ間のデータフォーマットの違い

センサデータのフォーマットはセンサのベンダごとに異なる．そのため，同じ対象物のデータでもペイロード部分は異なる内容となり，データの比較が煩雑となる．

B) センサで扱う単位の違い

通常，仕向けやベンダの違いによりセンサデータごとに単位が異なるため，データを適切に補正した後，比較，統合処理を行わなければならない．

C) センシングされた時刻の違い

センシングされたタイムスタンプに遅延があると異なる物体と認識され統合が行われない，異なる時刻にセンシングされたデータ同士が統合される可能性がある．また，センシングされた時刻から知能系 ECU がデータを受信するまでの伝送遅延が加算される課題もある．

D) センサで扱う座標系の違い

センサ毎に取り付け位置や座標の原点に対する各軸の正負の向きが異なるため，自車と周囲の物体の位置情報等の比較を行うにはアプリケーションで値を補正する必要がある．また，センサの追加や仕様変更時にこの補正が毎回発生し，工数増加を引き起こす．

提案手法では，A)から D)の課題に対し，3 つの手順で解決を図る．はじめに，各レイヤ間の差異を抑制するためのレイヤを追加する．次に，追加したレイヤ内で異なる単位や座標などの変換を行う．最後に上記それぞれの修正を容易に実現ためのツールを設けた．

3.3. アーキテクチャ検討

はじめに，レイヤ間に生じるデータフォーマットの差異を吸収するアーキテクチャを図 2 に示す．

本研究では，アプリケーション間のデータの差異を吸収する車両適合レイヤを追加し，異なるレイヤ間のデータプロトコルを統一する．次節以降で，この車両適合レイヤで行う処理について説明する．

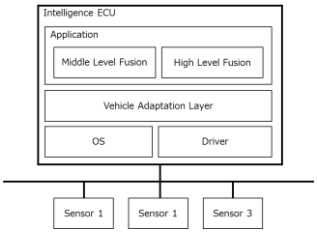

図 2 アーキテクチャ

3.4. データ構造の共通化

3.4.1. データフォーマットの共通化

異なるセンサのデータを統合するには，共通に保持する値の比較が必要である．提案手法は，センサデータを各データの統合や管理に用いるメタデータとそれ以外の固有データに分ける．メタデータを用いることにより，センサ固有のデータに因らない統合が可能となり，異なるレイヤ間のデータ統合が容易となる．

提案手法で用いるメタデータを，表 1 に示す．メタデータはさらに，Common data と Object data で構成する．Common data 内のセンサ種別には，センサ種類を格納する．オブジェクト ID には，データ管理のためにセンサ毎に管理される ID を格納する．また，タイムスタンプには，時刻補正を行ったタイムスタンプ情報を格納する．物体情報は，センシングされたオブジェクトの種別（歩行者，車両等）を格納する．

一方，Object data には，動的なデータと静的なデータの種別をそれぞれ定義する．センサで扱うオブジェクトデータには，カメラ等から取得される物体等の動的に変化するデータと，地図のような静的なデータがある．

表 1 Low-Middle Level 間で付加する共通項

Common data	ⅰ) センサ種別
	ⅱ) オブジェクト ID
	ⅲ) タイムスタンプ
	ⅳ) 物体種別
Object data	ⅴ) オブジェクトデータ

主に動的データは，自車周囲の状況を把握する事が可能であり，静的データは地図情報等にカメラでセンシングした白線や路肩の情報をマッピングし，地図上の自車位置を正確に推定する事が可能である．そして，これらの動的，静的なデータの統合には，両者で共通に持つ位置情報を比較する．動的，静的データでそれぞれ保持するデータを表 2 のように定義する．

表 2 オブジェクトデータの内訳

オブジェクトデータ	動的データ	ⅰ) X 座標
		ⅱ) Y 座標
		ⅲ) X 方向速度
		ⅳ) Y 方向速度
	静的データ	ⅵ) 相対方向
		ⅶ) 経緯度情報

これらの処理を 3 レイヤのセンサフュージョンに当てはめた場合，動的データ，静的データの各ノード内の統合処理が Middle-level 処理に相当し，動的データと静的データの統合が High-level 処理に当たる．

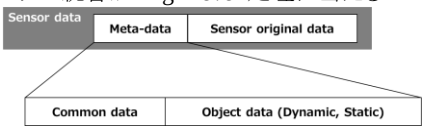

図 3 メタデータの構造

3.4.2. 各データの共通化

各センサで異なるデータの単位や座標系の変換処理も車両適合レイヤで一元的に行う．

提案手法では，はじめに Low-level で処理されたデータを車両適合レイヤに渡す．車両適合レイヤでは予め設定されたデータの単位や座標系，伝送遅延等の情報を元に，共通の単位や座標系に変換・補正する．そして，更新されたデータを次のレイヤへ渡す．例えば，図 4 に示す通り，Low-level で処理されたセンサデータ

はセンサ毎に単位や分解能が異なるが，車両適合レイヤで補正をした後は共通の単位となり，次の Middle-level で行う値の比較・統合を容易にする．

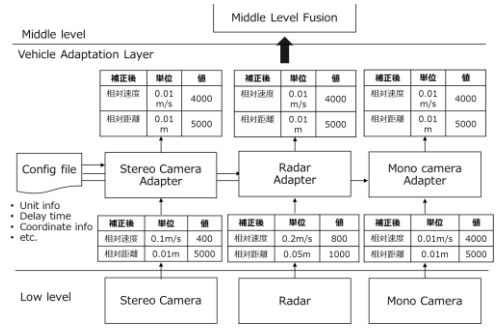

図 4 センサの単位変換

4. 検証

提案手法の有効性を確認するため，今回開発したアーキテクチャ及びツールを用いた開発と，従来のように車両適合レイヤを追加せず，手作業で仕様変更に対応した場合の開発工数を比較した．比較のシナリオとして，既存のセンサフュージョンのシステムに，新規にセンサを追加した開発を例に執る．

はじめに，図 5 に a) 提案手法を用いた方法と，b)従来の開発方法の作業フローを示す．図 5 に示す通り，提案手法を用いて変更を加える場合，すべてツール上で変更することが可能であり，従来手法のようなアプリケーションの大幅な修正を抑制できる．我々の試算では，1 メッセージあたり半分の時間で処理を行える．

今回，検証に用いたプロジェクトでは対象ステップが 18.9kstep あった．1kstep の作業量を 0.5 人月と見積もった場合，提案手法を適用した場合と従来の手作業で全て修正する場合の開発工数の比較を図 6 に示す．

開発工数の比較結果より，提案手法は従来の開発方法に比べ 42%開発工数を削減出来ている．

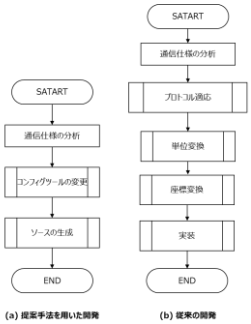

図 5 提案手法と従来手法の開発フローの比較

さらに，今回の比較は一度の仕様変更を対象としたが，開発中に仕様変更は何度も発生することが想定される．したがって，その都度開発工数を削減できる提案手法はさらに優位と言える．また，提案手法ではソースコードを自動生成するため，手作業で修正を加える従来方法に比べ，不具合を作り込む可能性が低く，ソフトウェアの品質を高めることができる．

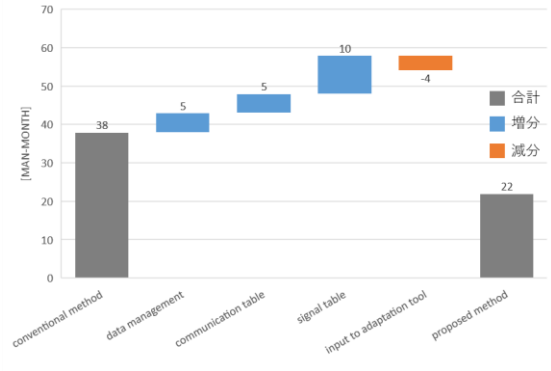

図 6 作業項目毎の工数比較

5. まとめ

複数レイヤで構成されるセンサフュージョン処理において，異なるレイヤ間のデータの授受を効率的に行う手法を提案した．提案手法では，各フュージョンレイヤ間のデータを共通プロトコルで扱う車両適合レイヤを追加し，レイヤ間のデータの差異を吸収する方法を提案した．また，車両適合レイヤの設定を変更するツールを作成し，従来に比べ 42%の開発工数削減を実現した．

参考文献

[1] Hall, David, and James Llinas. Multisensor data fusion. CRC press, 2001.
[2] Cho, Hyunggi, et al. "A multi-sensor fusion system for moving object detection and tracking in urban driving environments." Robotics and Automation (ICRA), 2014 IEEE International Conference on. IEEE, 2014.
[3] Esteban, Jaime, et al. "A review of data fusion models and architectures: towards engineering guidelines." Neural Computing & Applications 14.4 (2005): 273-281.
[4] Harris, C. J., A. Bailey, and T. J. Dodd. "Multi-sensor data fusion in defence and aerospace." The Aeronautical Journal 102.1015 (1998): 229-244.
[5] Haberjahn, Mathias, and Karsten Kozempel. "Multi level fusion of competitive sensors for automotive environment perception." *Information Fusion (FUSION), 2013 16th International Conference on*. IEEE, 2013.