



自動運転ソフトウェア

—オープンソースソフトウェアの利活用—



加藤真平（東京大学大学院情報理工学系研究科／名古屋大学未来社会創造機構）

自動運転における共通プラットフォームの必要性

自動運転技術には2つの大きな流れがある。1つは高度運転支援システム（Advanced Driver Assistance Systems：ADAS）の延長であり、高速道路や自動車専用道路であれば2020年前後に実用化されると見られる技術である。アメリカ合衆国運輸省（National Highway Traffic Safety Administration：NHTSA）の定義で言うレベル1～2に相当する。一方、車両が自律的に運転行動を決定し、制御を行う完全自動運転システムは、NHTSAの定義で言うレベル3～4に位置づけられる。これまでのADASとは一線を画し、ロボット技術や人工知能（Artificial Intelligence：AI）を中心とした研究技術も必要となってきた。DARPA Urban Challengeでもその有用性が実証された。

ADASはすでに一部の自動車メーカーや部品メーカーによる商品化にまで発展している。スバルのアイサイト製品やMobileye社のMobileye製品は独自のハードウェアとソフトウェアを統合した代表的なADAS技術として知られている。東芝のViscontiは画像処理チップとしてシェアが高く、画像認識ソフトウェア等とともに部品メーカー各社に供給されている。近年ではNVIDIA社のDrive PXが画像処理専用プロセッサ（Graphics Processing Unit：GPU）を採用した自動運転用システムオンチップ（System on Chip：SoC）として注目されており、機械学習やディープラーニングのような新しい技術を導入した物体検出システムも登場してきている。さらに従来の自動車業界の枠を超え、Google社に代表されるIT

ベンダも自動運転技術の開発競争に一石を投じており、ADASというよりは完全自動運転という観点でハードウェアとソフトウェアの開発を進めている。

自動車メーカーやITベンダの開発競争の激化の一方で、自動運転技術の共通プラットフォームはいまだ見えてこない。共通プラットフォームの欠如は、研究開発のオープン性を阻害し、大学はおろか企業の先行研究においても開発効率を悪化させている。特に完全自動運転には多くの新技術が必要とされる見込みであり、それらの統合化や評価実験に際して、共通プラットフォームの創出は急務である。自動運転システムはコンピュータシステムであり、コンピュータシステムの歴史を見ると、UNIXやLinux、Androidといったオープンソースのソフトウェアプラットフォームが普及し、研究開発を加速させてきた。本稿では、自動運転システムの研究開発を加速させるべく、名古屋大学で開発を行ってきたオープンソースの自動運転ソフトウェアプラットフォーム「Autoware」¹⁾を紹介する。

想定する車両とセンサ

Autowareでは、車載コンピュータから車両操作（アクセル、ブレーキ、ステアリングなど）を可能にするインターフェースが存在することが前提となる。自動車メーカーや部品メーカーは、自社でそのようなインターフェースを持たせるように車両を改造できるが、電機メーカーやITベンダのようなサードパーティ企業にとっては、しばしばそのような改造は困難となる。今日、市場で購入可能な自動運転の車両ベースとしてはZMP社のロボカーがある。ロボカーはア



図-1
ロボカー車両と各種センサから構成される自動運転車の事例

クセルやブレーキ、ステアリングの電子制御ユニット（Electronic Control Unit：ECU）にコマンド送信するゲートウェイが取り付けられているため、ロボカーを導入して、必要なセンサとコンピュータを取り付ければ、自動運転システムの研究開発プラットフォームにすることができる。

図-1に名古屋大学で使用している車両、センサ、車載コンピュータの構成例を示す。図の車両はロボカーHVという製品であるが、ZMP社のロボカーであればどのラインナップでも利用可能である。センサは多様な組合せが考えられる。最小構成としては3次元の光学距離計測（Light Detection and Ranging：LIDAR）センサと前方カメラであるが、全地球航法衛星システム（Global Navigation Satellite System：GNSS）や2次元LIDARセンサ、全方位カメラなども利用可能である。車載コンピュータは現状デスクトップ型かラップトップ型が利用可能である。将来的には組込みSoCボード型にも対応予定である。

Autoware

Autowareの特徴は、自動運転に必要となるソフトウェア一式がモジュール化されている点である。基本的な環境はLinuxとなる。その上にRobot Operating System（ROS）をインストールし、各機能のモジュール化を実現している。画像処理関係の

モジュールにはOpenCVを利用する。必要に応じてオリジナルのアルゴリズム実装を提供している。

Autowareの処理フローを図-2に示す。各種センサから取得できるデータとして、カメラからの画像、LIDARからの点群（ポイントクラウド）、GNSSからの緯度経度（測位情報）がある。目的地までの道路経路はスマートフォンなどのナビアプリから取得できる。それらの入力データに基づいてAutowareのコア機能である認知、判断、操作のモジュールが動作する。

物標認識：Deformable Part Model（DPM）²⁾やRCNN（Region with Convolutional Neural Network）³⁾などのアルゴリズムをベースとし、Kanade-Lucas-Tomasi（KLT）トラッカーアルゴリズムやKalman Filter（KF）を用いた物体追跡、さらにはLIDARセンサからの距離情報を付加するように実装されている。そのほかにLIDARセンサを使った物体検出機能も提供しており、ポイントクラウドライブラリ（Point Cloud Library：PCL）を利用した実装である。

位置認識：Autowareでは、3次元ポイントクラウド地図と3次元LIDARセンサデータをマッチングして自己位置推定を行っているが、その実装にはPCLに含まれているNormal Distributions Transform（NDT）アルゴリズム⁴⁾の実装を利用している。

経路計画：経路生成や軌道生成にはA*探索アルゴリズムとState Latticeプランナー⁵⁾を利用して

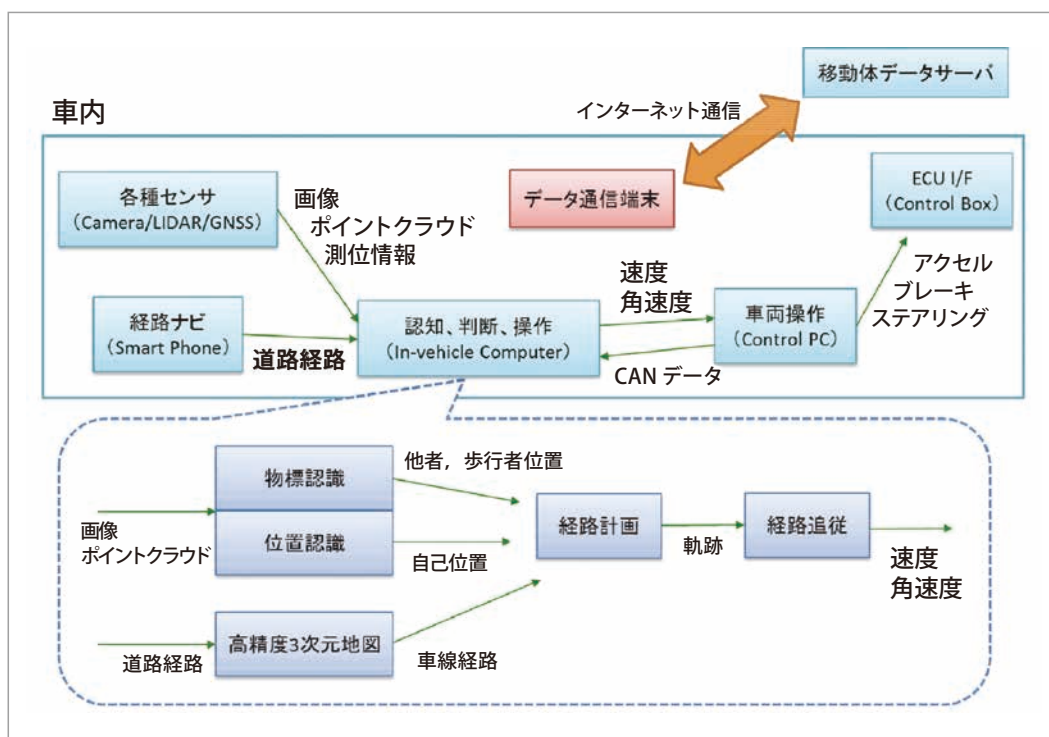


図-2
Autoware の処理フローと関連するコンポーネント

いる。公道で車線のある場所を自動運転する場合には車線中心線を基本経路としているが、駐車場のようないフリースペースが多い場所では A* 探索によって目的地までの経路を算出できる。いずれの場合も基本経路を生成したあとに別途、障害物回避を目的とした複数経路候補を State Lattice アルゴリズムによって生成できる。

経路追従：経路計画モジュールから経路が生成されたあとは、その経路を Pure Pursuit アルゴリズム⁶⁾によって追従する機能を提供している。経路追従モジュールは適切な速度と角速度を出力するため、車両側のコントローラが速度と角速度を受け取るようなインターフェースになっていれば、いかなる車両でも Autoware を用いることができる。

そのほか、Autoware には自動運転システムを操作、監視する機能もある。特に開発者向けにはグラフィックユーザインタフェース（Graphic User Interface：GUI）ツールを提供し、各モジュールの起動などを簡易化している。システムの状態は RViz という ROS 付属のビューワを利用している。図-3 にその一例を示す。一方、運転席に座っているユーザに対しては、Android タブレットのアプリを提供

し、ナビアプリによるルート検索や自動運転システムの ON/OFF を可能にしている。

画像処理やポイントクラウドの処理は計算量が多く、しばしば高度なアルゴリズムはリアルタイム用途に向かない傾向にある。Autoware では、そういったアルゴリズムは GPU やマルチコアを活用して並列化し、高速なリアルタイム処理を実現している。GPU プログラミングには Compute Unified Device Architecture（CUDA）を使用し、マルチコアプログラミングには Pthread や OpenMP を利用している。これらの並列化プログラミング技術により、各処理に対して 5 ～ 10 倍の高速化を実現できている。その結果、高度なアルゴリズムを使いつつ、ラップトップ PC だけで交通量の多い公道においても自動運転の実験が可能になっている。

公道実験の状況

実際、名古屋大学は、共同研究先の企業および愛知県その他自治体の協力を得て、名古屋市守山区の県道 15 号線を使った公道での自動運転技術の実証実験を実施している。この実験に利用している車両、



図-3
Autoware による
自動運転システム
状態の可視化

センサ、およびソフトウェアは、本稿で紹介してきたロボカー、LIDAR / カメラ、そして Autoware である。この実験は県道 15 号線のバス優先レーンを利用している。これは交通量の多い環境において周囲の車両からの割込みや右左折の妨害、そして何より実験中の事故を極力減らすための対策である。今後は、バス優先レーンでの自動運転実験の成果を応用して、過疎地などで追い越し車線なども含めた実験を展開していく予定である。公道とテストコースでの実験を重ねて、達成できている成果を以下に挙げる。

- 最高時速 60km/h での速度制御
- 信号認識による停止と発進
- 周囲車両とのディスタンスキープ
- 歩行者がいる場合の一旦停止
- 交差点の右左折（対向車なし）
- ナビアプリで指定した経路の走行

自動運転ソフトウェアの今後の発展

本稿では、オープンソースの自動運転ソフトウェアプラットフォーム「Autoware」の概要を紹介した。Linux や ROS といった汎用ソフトウェアをベースとしており、物体検出、自己位置推定、経路計画、軌跡生成、車両制御など、自動運転に必要な機能がひととおり揃っている。GPU やマルチコアを利活

用し、処理の遅いアルゴリズムは高速化されているため、リアルタイム用途にも対応できている。これからの自動運転システムの研究開発に有用なソフトウェアプラットフォームとして開発を継続していく予定である。最新版の Autoware は <https://github.com/cpfl/autoware/> からダウンロード可能である。

参考文献

- 1) Kato, S., et al. : An Open Approach to Autonomous Vehicles, *IEEE Micro*, Vol.35, No.6, pp.60-69 (2015).
- 2) Felzenszwalb, P., et al. : Object Detection with Discriminatively Trained Part-Based Models, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.32, No.9, pp.1627-1645 (2010).
- 3) Girshick, R., et al. : Region-based Convolutional Networks for Accurate Object Detection and Semantic Segmentation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.31, No.1, pp.142-158 (2016).
- 4) Magnusson, M., et al. : Scan Registration for Autonomous Mining Vehicles using 3D-NDT, *Journal of Field Robotics*, Vol.24, No.10, pp.803-827 (2007).
- 5) Pivtoraiko, M., et al. : Differentially Constrained Mobile Robot Motion Planning in State Lattices, *Journal of Field Robotics*, Vol.26, No.3, pp.308-333 (2009).
- 6) Coulter, R. : Implementation of the Pure Pursuit Path Tracking Algorithm, Robotics Institute, Tech. Rep. CMURI-TR-92-01 (1992).

(2016 年 2 月 15 日受付)

加藤真平（正会員） shinpei@is.nagoya-u.ac.jp

2004 年慶應義塾大学理工学部卒業。2008 年同大学院理工学研究科博士課程修了。博士（工学）。2009～2011 年までカーネギーメロン大学、2011～2012 年までカリフォルニア大学にて客員研究員、2012～2016 年まで名古屋大学大学院情報科学研究科准教授。現在、東京大学大学院情報理工学系研究科准教授。