

# 複数 ROS2-FPGA ノードの統合による自律走行ロボットカーの設計と実装

森 隼人<sup>1</sup>, 天野 隼斗<sup>2</sup>, 水谷 彰伸<sup>2</sup>, 岡崎 英佑<sup>3</sup>, 今野 湧稀<sup>3</sup>, 佐田 公平<sup>3</sup>,  
小野 智寛<sup>2,4</sup>, 吉元 裕真<sup>2,5</sup>, 田向 権<sup>2,5</sup>, 大川 猛<sup>3</sup>, 菅谷 みどり<sup>1</sup>

<sup>1</sup> 芝浦工業大学大学院理工学研究科 <sup>2</sup> 九州工業大学大学院生命体工学研究科

<sup>3</sup> 東海大学情報通信学部組込みソフトウェア工学科 <sup>4</sup> 日本学術振興会特別研究員

<sup>5</sup> 九州工業大学ニューロモルフィック AI ハードウェア研究センター

## I. はじめに

運転手によるハンドルやブレーキ操作を必要としない完全自動運転の実現に向けて研究が盛んに行われている。自動運転に必要な画像認識技術には、低消費電力かつ高性能であることが求められる。Field Programmable Gate Array (FPGA) は低消費電力かつ高い並列処理性能を特徴とする再構成可能デバイスとして、自動運転への応用が期待されている。第 11 回相磯秀夫杯 FPGA デザインコンテストは、レベル 5 相当の完全自動運転に必要な FPGA 技術の研究を促進する事を目的とした競技会である。

ロボットシステム開発において一般的に用いられるミドルウェアである Robot Operating System (ROS) [1] は自動運転分野においても注目されており、オープンソースの自動運転ソフトウェアである Autoware[2] では ROS 及び ROS2 を採用している。本研究では FPGA 実装した各コンポーネントを ROS2 ノードとしてシステムを構築する事で、開発の生産性向上を目指す。

本論文では、本競技会に向けて ROS2 を用いて設計した自律走行システムに、複数の FPGA コンポーネントを統合した小型の自律走行ロボットカーの概要及び各機能について述べる。

## II. ロボットカーの概要

ロボットカーの構成を図 1 に示す。車体には標準搭載されている LiDAR センサ及び Raspberry Pi を取り除いた TurtleBot3 Burger[3]を用いた。更に、FPGA ボードとして Ultra96v2 を 2 枚搭載した。Ultra96v2#1 では、車両制御及び白線検出、信号認識を行う。各機能の統合には ROS2 を用いた。ROS2 を用いる事で、障害物の分離によるデバッグの効率化、開発の生産性向上を目指す。Ultra96v2#2 では YOLOv4(You Only Look Once version 4)を用いた障害物認識を行う。障害物情報は、ROS2 のトピックを介して Ultra96v2#1 に送信される。

## III. ULTRA96#1 ボードの実装

Ultra96v2#1 ボードでは、白線検出と信号認識、及び車体制御を行う。本論文では、ロボットカーに実装した白線検出と信号認識について紹介する。

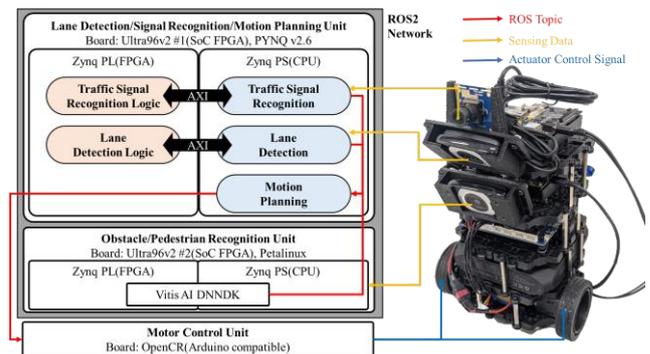


図 1 ロボットカーの構成

### A. 白線検出

本競技では、ロボットカーが人による操作なく白線に沿って走行する必要がある。従って、白線に追従して走行する為に、Hough 変換を用いた直線検出による白線検出機能を実装した。Hough 変換とは、自動運転システムで一般的に用いられる直線検出アルゴリズムである[4]。計算量が多い事から組み込み向けの CPU では処理時間を要する。その為、本システムでは白線検出処理のうち直線検出機能を FPGA 上に実装した。

白線検出の処理フローを図 2 に示す。はじめに入力画像を HSV 領域の画像へと変換する。その後、画像から白線に該当する範囲である白色領域を抽出する。出力画像に対して Canny 法を用いてエッジ抽出を施す。最後に Hough 変換によって直線検出を行い、検出した直線の角度  $\theta$  と原点からの長さ  $\rho$  を求める。この結果を用いて車両制御を行う。

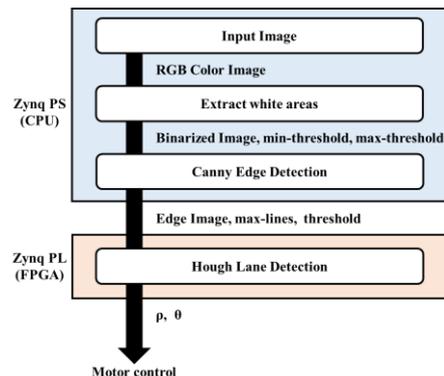


図 2 白線検出の処理フロー

## B. 信号認識

信号認識 IP コアの実装は、色情報と Hough 変換による円検出を組み合わせた先行研究[5]を参考に HW 実装を行った。まず、照明条件の影響低減の為に正規化 RGB 色空間に変換する。次に、RGB の各色情報を用いて信号機の候補領域を抽出する。抽出した候補領域に対し、Hough 変換により円を検出し、検出結果を信号領域として色情報（赤・黄・緑）を出力する。

白線検出と信号認識のハードウェアリソース使用量を表 1 に示す。2つの IP コアのリソース使用量の合計は、LUT は 100.5%、FF は 19.4%、BRAM18K は 57.9%、DSP48E は 38.3%であった。LUT の使用量が 100% を超過している為、リソース使用量の削減が必要である事が分かった。

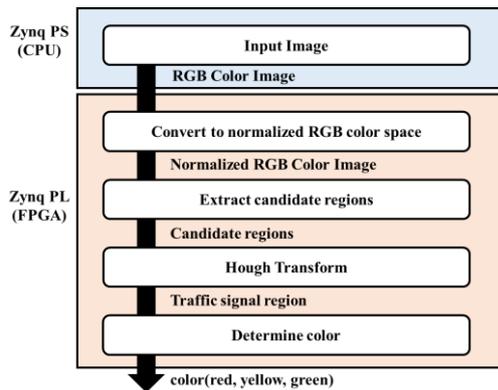


図 3 信号認識の処理フロー

表 1 ハードウェアリソース使用量

	Lane Detection	Traffic Signal Recognition	Total
LUT	56,142	14,745	70,887/70,560(100.5%)
FF	16,578	10,831	27,409/141,120(19.4%)
BRAM18K	205	45	250/432(57.9%)
DSP48E	5	133	138/360(38.3%)

## IV. 障害物認識

障害物を認識するため、FPGA 上で動作する物体認識システムを構築した。システムの訓練に必要なデータセットはシミュレータを用いて生成した。物体認識アルゴリズムには、You Only Look Once (YOLO)v4[6]を使用した。YOLOv4 の FPGA デバイスへの実装には AMD Xilinx が提供する Vitis AI[7]を用いた。

YOLOv4 をはじめとする深層学習による物体認識には、質の高い大量の画像データと、画像中にある物体の位置とラベルを示すアノテーションデータが必要になる。高品質で大量のアノテーションデータを人手で用意することは困難であるため、小野らにより提案されたサービスロボット向けの 3D シミュレータを用いた方法[8]を改良して、データセットの生成に使用した。

この手法によるデータセット生成の手順を以下に示す。

### 1. 3D モデルのスキャン

公開されている 3D オブジェクトにはテクスチャ情報がないため、3D プリンタを用いて印刷した

オブジェクトの 3D モデルを、3D スキャナ (EinScan-SP[9]) を使用して取得する。

### 2. 環境の生成

競技環境を模した環境をシミュレータ上に生成する。コースをランダムに並べた背景上に障害物や人形をランダムに配置する。同一の条件で配置された物体を、複数の視点から撮影する。

### 3. 入力データと教師データの生成

3D シミュレータで生成する際、生成場所や生成した物体のラベル情報はプログラム上で把握可能である。したがって、物体の配置からデータセットの生成まですべてをすべて自動的に実行することが可能である。



図 4 シミュレーターを用いたデータセット生成概要

## V. おわりに

本論文では、第 11 回相磯秀夫杯 FPGA デザインコンテストに向けて開発した ROS2 ベースの自律走行ロボットカーの構成及び各機能を紹介した。今後は白線検出と信号認識 IP コアの統合を行う。

## 謝辞

本研究の一部は JST, CREST, JPMJCR19K1 の支援を受けたものです。

## 参考文献

- [1] Quigley Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2, 2009.
- [2] Shinpei Kato, et al. "An open approach to autonomous vehicles." IEEE Micro 35.6, 2015
- [3] ROBOTIS, "turtlebot3," <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>, 14 Sep. 2022 Accessed.
- [4] El Hajjouji Ismaïl, et al. "A novel fpga implementation of hough transform for straight lane detection." Engineering Science and Technology, an International Journal 23.2, 2020.
- [5] Masako Omachi, and Shinichiro Omachi. "Traffic light detection with color and edge information." 2009 2nd IEEE International Conference on Computer Science and Information Technology. IEEE, 2009.
- [6] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).
- [7] Xilinx, "Vitis AI," <https://www.xilinx.com/products/designtools/vitis/vitis-ai.html>, 30 Aug. 2021 Accessed.
- [8] Ono, Tomohiro, et al. "Solution of World Robot Challenge 2020 Partner Robot Challenge (Real Space)." *Advanced Robotics* (2022): 1-20.
- [9] EinScan-SP, <https://www.einscan.com/desktop-3d-scanners/einscan-sp/>, 2022 Jan 25 Available