

Helmholtzの原理に基づく白線検出アルゴリズムの 自動運転ロボットへの応用

神前 克誠^{†1,a)} 松井 信太郎^{†1} 荒木 康利^{†1} 三浦 丈寛^{†1} 本吉 圭吾^{†1} 山下 恵三^{†1} 池原 陽大^{†1}
河津 拓歩^{†1} 黄 雨薇^{†1} 西村 昌浩^{†1} 阿部 柊斗^{†2} 沖野 玄勇^{†2} 橋口 裕太^{†2} 福田 航生^{†2}
柳原 健吾^{†2} 眞邊 泰斗^{†1,†2,b)} 柴田 裕一郎^{†1,†2,c)}

概要: 我々は、FPGA デザインコンテストに向けて、SoC FPGA を用いた自動運転ロボットを開発している。車体が白線に沿って正確に走行するためには直線だけでなく、曲線も正確に検出する必要がある。そこで、我々は直線だけでなく曲線にもロバストな白線検出アルゴリズムを採用し、走行性能の向上を図った。本アルゴリズムを組み込んだ自動運転システムを Digilent Zybo Z7-20 に実装するとともに、シミュレーションに基づく評価を行い、Canny フィルタを利用した場合と比較して白線検出における、誤検出を低減できることを示した。

キーワード: SoC FPGA, 自動運転, 白線検出

1. はじめに

近年、飛躍的な進歩を遂げている自動運転技術は SAE により運転操作の主体に合わせて 6 段階に分けられている [1]。完全自動のレベル 5 では、すべての領域において、システムが運転に関わる操作を行う。そのため、システムはリアルタイムに取り込まれる外界情報に基づき、あらゆる場面で人間と同等の判断を瞬時に行う必要がある。

この要件を満たすために、自動運転システムを FPGA を用いて実装する手法が提案されている。信号機の色や、車道を横断する物体などを検出するには画像処理を行う必要がある。画像処理を高速化する手段として ASIC や GPU が用いられることが多い。しかし、ASIC を使った実装では、設計に変更があるたびに再生産が必要となるためコストや開発効率が問題になり、GPU は電力面での課題がある。これに対して、コストと電力効率の両面で優れる FPGA を用いることで、これらの課題を解決できる。

また、自動運転では、システムが白線を正確に検出し、それに合わせて走行する必要がある。直線を検出する手法としては、Hough 変換などが挙げられるが、曲線の検出精度が高くない。そこで、我々は曲線を高精度に検出できる

ように Helmholtz の原理に基づくアルゴリズムを採用し、ハードウェア向けに最適化して FPGA に実装した。

本論文では、2 章で曲線輪郭検出アルゴリズムについて説明し、3 章では Helmholtz の原理に基づく白線検出アルゴリズムについて説明する。4 章では自動運転ロボットの構成について説明する。5 章では自動運転システムの性能について評価・考察し、6 章で結論と今後の展望を述べる。

2. 輪郭線検出アルゴリズム

カメラを用いた白線検出には様々なアルゴリズムが提案されている [2], [3]。しかし、これらの手法は半径の小さいカーブの検出に弱いという問題がある。古性らは、半径の小さなカーブも高精度で検出可能な輪郭線検出アルゴリズムを提案している [4]。古性らのアルゴリズムは白線検出に応用可能であるが、計算量が多く、自動運転技術に求められる省電力かつ低レイテンシな実装は容易ではない。本章では、古性らの輪郭線検出アルゴリズムとそれを省電力かつ低レイテンシに実現する松井らの FPGA 実装 [5] について述べる。

2.1 Helmholtz の原理に基づく輪郭線検出 (古性ら)

古性らは、Helmholtz の原理 [6] に着目した輪郭線検出アルゴリズムを提案している [4]。このアルゴリズムは、Hough 変換の特徴量に Helmholtz の原理に基づいた特徴量を加えることでノイズを低減し、半径の小さなカーブに

^{†1} 現在、長崎大学大学院工学研究科総合工学専攻

^{†2} 現在、長崎大学工学部工学科

a) kamimae@pca.cis.nagasaki-u.ac.jp

b) tmanabe@nagasaki-u.ac.jp

c) shibata@cis.nagasaki-u.ac.jp

においても輝度勾配方向ごとに高精度な輪郭線の検出を可能にする。なお、入力には Sobel フィルタなどでエッジが強調された画像であり、出力は検出した輪郭線を示す二値画像である。これらの処理は、近似を用いることで畳み込みによる処理で実現されている。

アルゴリズムでは、入力画像に対して輝度勾配方向 θ ごとに図 1 に示す「エッジ選択」を行い、これらの論理和を出力画像とする。エッジ選択では、「エッジ抽出」「画像圧縮」「画像拡大」「論理積」を用いた階層構造の処理を行う。 $\theta = \frac{\pi}{4}$ の時のエッジ抽出の処理フローを図 2 に示す。エッジ抽出では、畳み込みと二値化をそれぞれ 2 回ずつ行う。

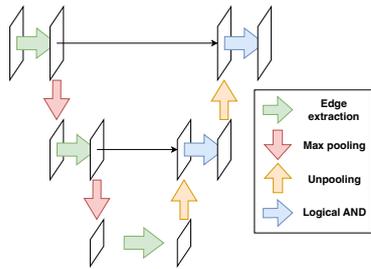


図 1: エッジ選択 (Edge selection) の処理フロー (古性ら, 2017)

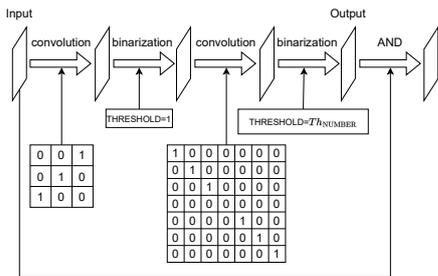


図 2: エッジ抽出 (Edge extraction) の処理フロー (古性ら, 2017)

2.2 Helmholtz の原理に基づく輪郭線検出の FPGA 実装 (松井ら)

松井らは、古性らのアルゴリズムの多量の畳み込み演算に着目し、畳み込みをストリーム処理により低レイテンシで実現できる FPGA に本アルゴリズムを実装した [5]。ハードウェア設計を図 3 に示す。入力されるストリーム画像に対するエッジ選択は edge selection モジュールで行われる。輝度勾配方向ごとに依存関係がないため、それぞれのエッジ選択処理は並列に行われる。edge_selection モジュールは、エッジ抽出を行う edge_extraction モジュールと画像圧縮を行う max_pooling モジュールで構成される。max_pooling モジュールにおいて間隔を空けたカーネルで畳み込みを行うことで画像拡大の処理を省略している。

結果として、組み込み向けの省電力 CPU を用いたソフトウェア実装と比較して精度を落とすことなく、最大で約 1000 倍高速な処理を実現している。

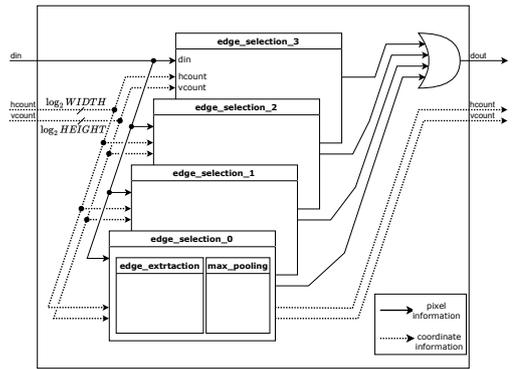


図 3: システムの構成図 (松井ら, 2022)

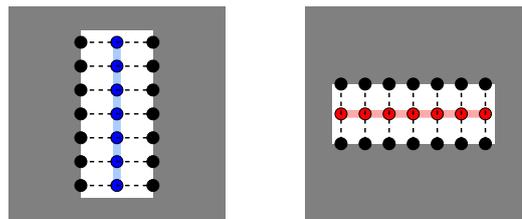
3. 白線検出アルゴリズム

本章では、SoC FPGA 上に実装した白線検出アルゴリズムについて説明する。2 章で説明した松井らの Helmholtz の原理に基づく輪郭線検出アルゴリズムの FPGA 実装を基に、PL (Programmable Logic) 部の輪郭線検出と PS (Processing System) 部の白線検証を組み合わせた、白線検出アルゴリズムについて述べる。

3.1 車道外側線・車道中央線・停止線の検出

まず、輪郭線画像をラスタスキャンし、垂直・斜め方向のエッジとエッジの間の距離が白線の太さに近ければ、車道外側線・車道中央線とみなす。次に、PL 部でこの 2 つのエッジの中心を通る点群を取得し、PS 部で取得した点群から近傍の点を繋いで車線の中心を通る線として構成する (図 4a)。

停止線の場合も同様に、水平方向のエッジとエッジの間の距離が白線の太さに近いものを停止線とみなし、停止線の中心を通る線として構成する (図 4b)。



(a) 車道外側線・車道中央線 (b) 停止線

図 4: 車道外側線・車道中央線・停止線の、中心線上の点群とそれら点群から構成した線

3.2 路面の判断

3.1 節で取得した車道外側線・車道中央線・停止線を用いて、路面形状の状態の判断を行う。

路面状態は、図 5 に示すように 5 通り挙げられる。通常状態は直進の道路または右折カーブに位置していることを示す。停止状態は停止線前に、交差点状態は交差点内に位置していることを示す。カーブ状態は左折カーブ中である

ことを示す。未検出状態は車道外側線・車道中央線・停止線が全て検出できなかったことを示す。

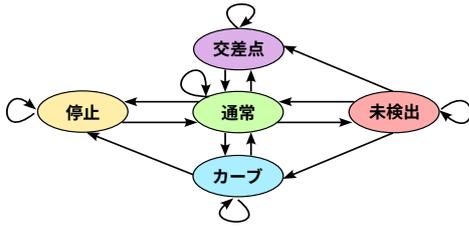


図 5: 路面状態の状態遷移

路面状態の状態遷移は、3.1 節で取得した車線・停止線の位置関係で判断する(図 6)。車道外側線・車道中央線・停止線が見つからない場合は一度通常状態に遷移し、未検出状態に遷移する。

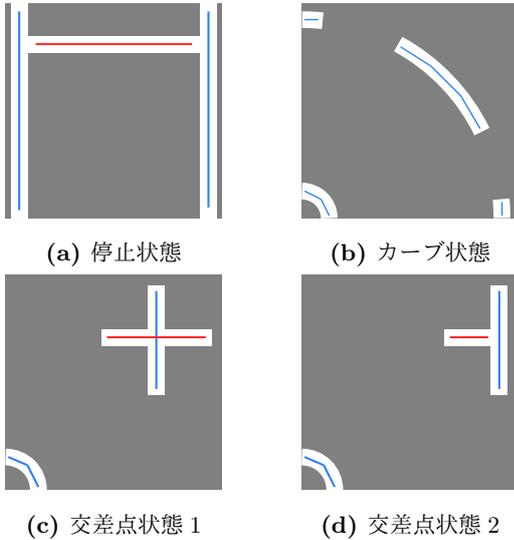


図 6: 各路面状態の車道外側線・車道中央線・停止線の位置関係(青線は垂直・斜め方向の線、赤線は水平方向の線を示す)

4. 自動運転ロボット

本章では、本研究で開発した自動運転システムと、それを搭載する自動運転ロボットについて述べる。

4.1 自動運転ロボットの構成

表 1: 自動運転ロボットの構成要素の詳細

| 構成要素 | デバイス名 | 数量 |
|------------|---------------------------|----|
| SoC FPGA | Digilent Zybo Z7-20 | 2 |
| カメラ | Digilent Pcam5C | 2 |
| モータ | LEGO MINDSTORMS EV3 L | 2 |
| モータドライバ | TOSHIBA TB6612FNG | 1 |
| Wi-Fi ドングル | I-O DATA DEVICE WN-S150UM | 1 |

本研究で開発した自動運転ロボットの構成要素の詳細を表 1 に示す。ロボットは、2 台の SoC FPGA ボード上に構築されるシステムより制御される。SoC FPGA ボードの電源には 5V 給電のモバイルバッテリーを使用しており、カ

メラ及び Wi-Fi ドングルにはボードより給電される。モータ/モータドライバには 9V 給電の乾電池バッテリーを使用した。

4.2 自動運転ロボットのシステム構成

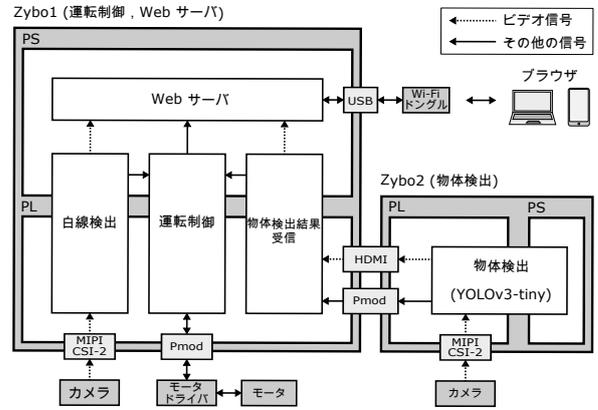


図 7: 自動運転ロボットの概要図

自動運転ロボットに搭載するシステムの概要図を図 7 に示す。

まず、図 7 右部の Zybo2 について説明する。Zybo2 は、ロボットの正面を映すカメラ映像に対して物体検出を行う。物体検出には機械学習を用いており、モデルには YOLOv3-tiny を用いた。多量の畳み込みの処理を PL 部にオフロードすることで省電力かつ高速な物体検出を可能にしている。Zybo2 で得た物体検出の結果は pmod を介して Zybo1 へ転送され、運転制御に用いられる。

次に、図 7 左部の Zybo1 について説明する。Zybo1 は、3 章で述べた白線検出アルゴリズムベースの運転制御を行う。運転制御では、白線の検出結果、Zybo2 より転送される物体検出結果、目標の経路情報をもとにモータドライバを制御することでロボットの進行方向を制御する。なお、モータを正確に制御するために PL 部では PID 制御が行われ、モータの回転数はシステムが目標とするアクセル値へと制御される。

また、Zybo1 上ではロボットのリモート制御やデバッグ用に Web サーバが動作する。この Web サーバは、HTTP 経由でモータのアクセル値、白線検出結果の出力映像、物体検出結果の出力映像等の走行情報の転送やロボットの運転経路を決めるための経路計画等を可能にする。

なお、Zybo1 の PS 部では、外部ストレージ(SD カード)より Ubuntu OS がブートする。Linux システムを Zynq 上で動作させるために必要なカスタムカーネルのビルドには Xilinx 社が提供する Linux 組み込み SDK Petalinux ツールを用いた。

5. 評価と考察

5.1 フィルタによる白線検出

本節では、古性らのアルゴリズムを用いて検出した白線

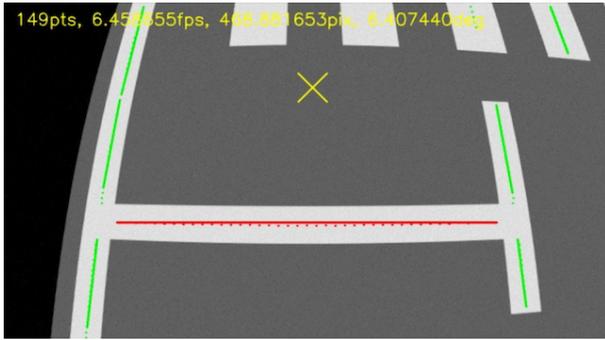


図 8: 古性らのアルゴリズムを用いて検出した白線の中心線

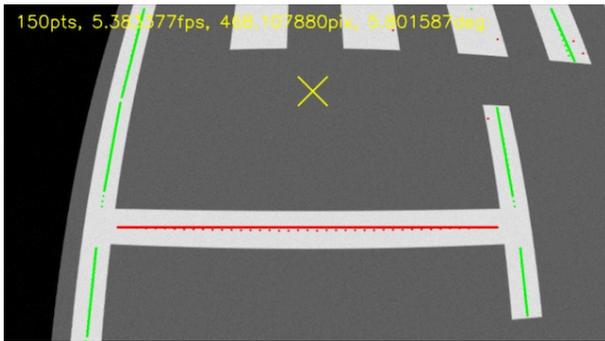


図 9: Canny フィルタを用いて検出した白線の中心線

の中心線と、Canny フィルタを用いて検出したものとの比較を行う。

まず、古性らのアルゴリズムを用いて検出した白線の中心線の画像を図 8 に示す。この画像では、白線の中心のみ点がプロットされており、期待通りの結果が得られている。次に、Canny フィルタを用いて検出した白線の中心線の画像を図 9 に示す。この画像では、右上の横断歩道の部分に停止線を誤検出してしまっていることがわかる。

Canny フィルタを用いて検出した場合、斜めの線と横の線に挟まれた部分を誤って停止線と検出してしまっていると考えられる。これに対して、古性らのアルゴリズムを用いた検出では白線を検出する際の計算にエッジの輝度勾配方向を利用できる。停止線の検出に真横のエッジのみを用いることで、誤検出の防止が可能となっている。

5.2 直進性

本節では、松井らによる実装を用いた車体制御による、直進時の左右方向のブレを評価する。本論文では、C++ で実装したシミュレータを用いて 4,400 mm の直進道路を走行させ、道路の中心からの車体の距離を評価した。その結果を図 10 に示す。

図 10 において、横軸は車体が前に進んだ距離、縦軸は車線の中心からの距離（進行方向左側が正）である。この結果より、走行距離 4,400 mm に対し最大のブレが約 0.55 mm と非常に小さいことがわかる。

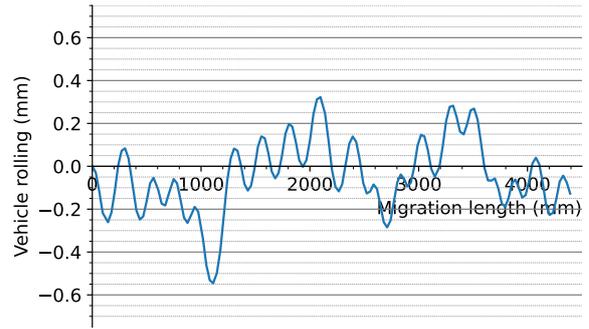


図 10: 直進時の左右のブレ（縦軸は進行方向左側が正）

6. 結論

我々は、Helmholtz の原理に基づく曲線輪郭検出アルゴリズムを用いた白線検出アルゴリズムを自動運転システムに組み込み、Diligent 社の Zybo Z7-20 上に実装した。

本論文の評価では、走行距離 4,400 mm に対してのブレを最大で約 0.55 mm に抑えられることを示した。また、Canny フィルタを用いて検出した白線検出と比較して誤検出を防ぐことができることを示した。

本システムが、従来のアルゴリズムを利用したものと比較して誤検出を防ぐことができ、自動運転に有用であることを確認することができた。今後は障害物の検出精度の向上に向けて継続して開発を行っていく。

謝辞 本研究の発表の場を設けてくださった相磯秀夫杯 FPGA デザインコンテスト実行委員長 渡邊実教授、副委員長 泉知論教授ならびに、実行委員会のみなさまに、この場を借りて深く御礼申し上げます。

参考文献

- [1] SAE International: SAE Levels of Driving Automation Refined for Clarity and International Audience, <https://www.sae.org/blog/sae-j3016-update>. (参照 2022-09-14).
- [2] 山口直人, 田森信行, 塩見彰睦: 適応エッジ保存平滑化を用いた白線検出手法, 電子情報通信学会論文誌 D, Vol. 88, No. 8, pp. 1421-1431 (2005).
- [3] J.C. McCall and M.M. Trivedi: Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 1, pp. 20-37 (2006).
- [4] 古性裕之, 毛利宏: Helmholtz の原理に基づく曲線輪郭線検出と道路白線検出への適用, 日本機械学会論文集, Vol. 83, No. 856, 17-00203 (2017).
- [5] 松井信太郎, 眞邊泰斗, 柴田裕一郎: Helmholtz の原理に基づく曲線輪郭線検出の FPGA 実装, 信学技報, Vol. 122, No. 60, pp. 113-118 (2022).
- [6] Desolneux, A., Moisan, L. and Morel, J.-M.: *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, Springer (2008).