

VNCを介した画像伝送システムを備えた 自動運転車のSoC FPGA実装

本吉 圭吾^{†1,a)} 今村 優太^{†1} 西改 太一^{†1} 藤田 光暉^{†1} 古川 大喜^{†1} 松田 将朋^{†1} 森 龍馬^{†1}
荒木 康利^{†1} 三浦 丈寛^{†1} 山下 恵三^{†1} 池原 陽大^{†2} 大平 開斗^{†2} 神前 克誠^{†2} 河津 拓歩^{†2}
西村 昌浩^{†2} 松井 信太郎^{†2} 友永 航生^{†1} 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 を用いた自動運転システムを開発している。画像ベースの自動運転車両の開発における生産性を向上させるためにはリアルタイムの画像伝送機能を備えた遠隔デバッグ環境が重要である。この論文では、Wi-Fi VNC 接続を介して、車載カメラ画像と特徴検出結果を観測できる画像伝送システムを提案する。本システムは Xilinx Zynq-7000 に実装した。本システムの評価では、VNC と SSH X11 フォワーディングの性能比が、画像サイズ 640×720 画素の場合に最も高くなることを示した。また、本論文では、自動運転システムの開発に使用される他の要素技術についても簡単に説明する。

キーワード：SoC FPGA, VNC, 自動運転

1. はじめに

自動運転は、システムが実行できるタスクによって、5つのレベルに分類される。その中で、多くの研究者が目標としているレベル5は、すべての運転タスクを実行し手動操作を必要としない自動運転システムである。そのため、レベル5自動運転システムは、搭乗者の生命を守るために高い安全性能を備えている必要がある。レベル5の自動運転に必要な安全性能を実現するために、システムは人間のドライバーと同等の広範な認知能力と迅速な意思決定能力を備えなくてはならない。

これらの要求を満たすために、自動運転システムをFPGAアクセラレータを用いたエッジコンピューティングによって実装する手法が提案されている。サーバサイドで画像解析を行う場合、ネットワークの不安定性と遅延によるリスクが生じる。そこで、エッジコンピューティングを行うことでリアルタイム性を向上させ、安全性能の向上が見込める。また近年、物体検出のような重いプロセスを高速化するために、アクセラレータがよく使われる。しかし、データ処理用チップの開発速度は年々向上しているのにも拘ら

ず、ASICの生産には時間がかかり、回路設計の変更がある度に作り直す必要がある。対して、FPGAをアクセラレータとして用いると、実機を用いた検証が容易となり、開発速度を向上させることができる。

このような技術的な要求に応えるために、FPGAテクノロジーを利用したレベル5自動運転システムの開発を加速させることを目的として、FPGAデザインコンテスト [1][2] が近年開催されている。参加者は、FPGAによって制御された小型の自動運転車を用意し、ルートから外れることなくミニチュア道路を走行しなくてはならない。さらに、車両は、信号機、横断歩道、交差点、障害物、歩行者を模倣した人形の認識などの運転タスクを処理しなくてはならない。加えて、大会では、センサとしてCCD/CMOSカメラのみを使用することが推奨されている。

しかし、開発を行うにあたって、遠隔の車両を検証する効率的な方法を実装していなかったため、画像ベースの自動運転車がうまく機能しているか確認することが難しい。そのため、無線で自動運転車をデバッグするためにVNCを利用した画像伝送システムを実装した。

本稿では、このシステムの概要及び評価を中心に述べるとともに、VNCの有益性について考察する。

2. 概要

本節では、本システムのハードウェア構成とシステムに

^{†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) kmoto@pca.cis.nagasaki-u.ac.jp

b) tmanabe@nagasaki-u.ac.jp

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

使用されている要素技術について説明する。

2.1 ハードウェア構成

開発中の自動運転車の姿を図 1 に、このシステムの構成部品の詳細を表 1 に示す。また、本システムのハードウェア設計概要を図 2 に示す。本システムの設計にあたって 2 台の SoC FPGA ボードと 3 台のカメラを使用した。また、車輪用に 2 個のモータ (左右) とそれに付随したモータドライバを使用し、電源には 5V 給電が可能なモバイルバッテリーを使用した。SoC FPGA ボードには、Digilent 社の ZYBO Z7-20 を用いた。FPGA ボードのうち 1 台は、自己位置推定、運転制御を行い、もう 1 台は、機械学習による物体検出を行う。本稿では、それぞれのボードを ZYBO 1, ZYBO 2 とする。本研究で実装した VNC サーバによるデバッグシステムは、ZYBO 1 の PS(Processing System) 部に実装した。

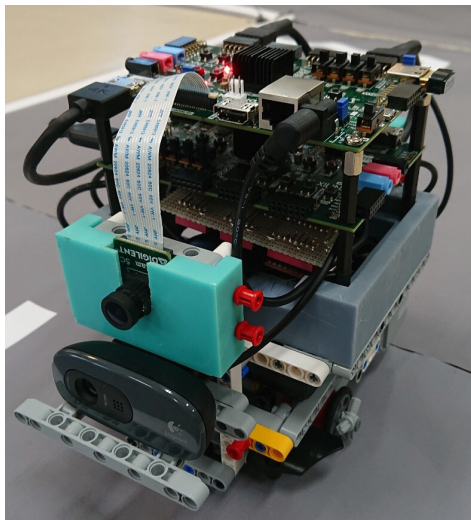


図 1 開発中の自動運転車両

表 1 使用機材の詳細

機材	製品名
FPGA ボード	Digilent ZYBO Z7-20
カメラ (前方/後方)	Digilent Pcam5C
USB カメラ	Logicool C270c
モータ (右/左)	LEGO MINDSTORMS EV3 L Motor
モータドライバ	TOSHIBA TB6612FNG
バッテリー	ANKER PowerCore 13000
Wi-Fi ドングル	TP-Link AC600

3 台のカメラのうち 2 台は、車体前後の道路情報を取得するために使用した。白線情報を取得するため、道路に対して 15° 下に傾けて車体前後に設置されている。これらのカメラには、Digilent 社の Pcam5C を用いており、それぞれ ZYBO 1, ZYBO 2 に MIPI コネクタを通して接続される。もう一台のカメラは、車体前方の物体を検出するため

に、車体前方に路面と水平に設置している。このカメラには、Logicool 社の C270c を用いており、USB ポートを通して接続される。モータドライバ、モータは、ZYBO 1 に接続されており、Pmod コネクタを通して接続される。

2.2 自動運転における他の要素技術

まず、自己位置推定のために、ZYBO 1 の Programmable Logic(PL) 部では、前方後方 2 つの画像情報に対して線分検出を行う。ZYBO 2 に送信される前方カメラの映像は、HDMI RX/TX ポートを通して ZYBO 1 へと送信される。線分検出のアルゴリズムには、[3] で提案したものをを用いた。得られた線分情報と、元の画像情報は、AXI を通して ZYBO 1 上の PS 部へと送信される。PS 部では、これらの画像情報とモータからのフィードバック情報を用いて自車位置を求める。また、PL 部で求めた線分情報から左右の白線を認識し、その傾きや車との位置関係を基に制御を行う。

一方、ZYBO 2 では、接続された USB カメラより得られた車体正面の映像に対して機械学習による物体検出を行う。検出結果は、Pmod コネクタを通して ZYBO 1 へ送信され、運転の制御に役立てられる。

3. VNC を用いた画像転送システム

3.1 VNC を用いるメリット

自動運転車両の開発における生産性を向上させるためには充実した遠隔デバッグ環境の構築が重要である。特に、遠隔操作する車両が画像ベースの処理を中心に動作する場合、デバッグ環境には画像転送システムが必要不可欠である。このとき、FPGA が持つ処理のリアルタイム性をできるだけ損なわないような検証環境が求められる。画像転送の方法の 1 つは SSH X11 フォワーディングである。しかし、Zynq-Linux 上では 1280×720 のフレーム画像をスループット 30 fps で取得できるが、SSH X11 フォワーディングではネットワークを介した画像転送がボトルネックになってしまい、システムのリアルタイム性が損なわれ、効率的な開発を行うことが困難になる。

この問題を緩和するために、画像転送において SSH X11 フォワーディングとは別の手段を導入する必要がある。そこで、本画像転送システムでは VNC を用いる。VNC はネットワーク上の離れたコンピュータを遠隔操作するためのソフトウェアである。VNC の通信には RFB プロトコル [4] を採用しており画像の転送に適している。本システムでは、VNC サーバをマスタ Zynq ボード (図 2 における“ZYBO 1”) の Zynq-Linux 上に構築し、その後、リモート PC で VNC クライアントを起動して画像を受信する。

3.2 VNC のためのソフトウェアとハードウェア

Xilinx が提供している Linux カーネル [5] によって生成されるイメージファイルなどを microSD に書き込み、その

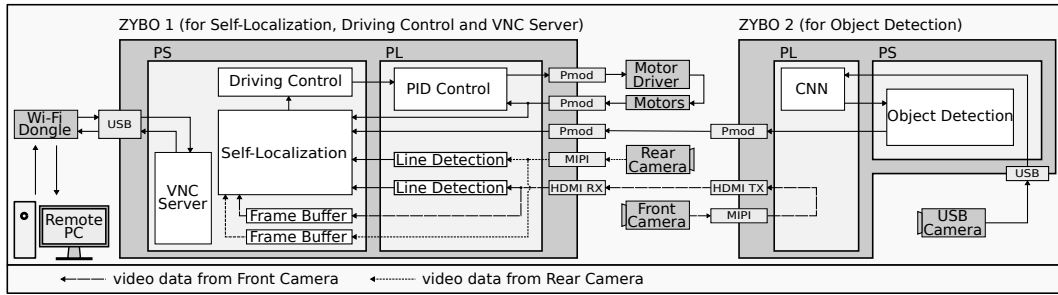


図 2 SoC FPGA 上に実装される自動運転システムの全体概要



図 3 USB アダプタ

microSD を ZYBO に挿入することで、Zynq 上で Ubuntu 20.04 LTS で動作させている。また、画像転送のために、VNC サーバの Xvnc TigerVNC 1.10.0 [6] を使用した。

ZYBO にイーサネットケーブルを接続すれば VNC を介して画像転送を行うことができるが、独立して走行する自動運転車のデバッグを行う際にはこれは走行の妨げになる。そのため、無線通信（特に Wi-Fi）による接続が好ましい。しかしながら、ZYBO Z7-20 には無線接続モジュールはないため、図 3 に示す、Zynq-Linux 上で利用可能な USB 無線アダプタ（TP-Link Ac600）を導入した。

USB アダプタは Realtek RTL8812AU チップセットを内蔵しており、Zynq 上で RTL8812AU を駆動するためにオープンソースのドライバ [7] を使用している。また、ドライバを使用するために Zynq 用の Linux カーネルの機能をいくつか変更している。Zynq 上で変更したカーネルイメージをブート、ドライバをインストールすることで、無線通信が可能になる。

4. 評価

本節では我々の車両に搭載されている画像転送システムの評価を行う。図 4 は画像転送システムによってリモート PC が取得できるデバッグビューの一例である。前方カメラと後方カメラで撮影された 1280×720 画素の 2 フレームをそれぞれ 640×360 画素にリサイズし、垂直に連結して送信している。2 節で説明した線分検出器の出力（検出線分数、各線分の始点や終点の座標とその勾配）を PS 側で取得し、前後それぞれのカメラ画像にオーバーレイして描画する。そしてこの線分情報をオーバーレイした画像をリモート PC へ転送する。そうすることで、自動運転システ

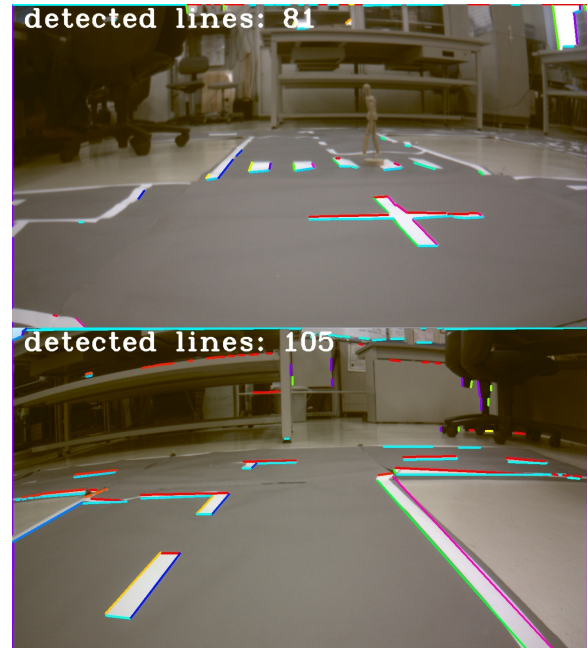


図 4 デバッグ画面（上：前方カメラ，下：後方カメラ）

ムの画像処理や PS/PL 協調処理が正常に動作しているかどうかを確認することができる。また、取得した画像は物体検出や人検出にも使われるので、トリミング処理を行わない。

停車中の車におけるデバッグビュー動画のスループットを、SSH X11 フォワーディングと VNC を比較することで評価した。画像の解像度は、連結後の 1280×1440, 640×720, 320×360, 160×180 の 4 種類で評価した。SSH X11 フォワーディングではリモートアクセス時に `ssh -X` を、VNC では転送画像の解像度に応じてサーバ/クライアントともに `vncserver/vncviewer -geometry WxH` を設定した上で評価した。SSH や VNC のその他の設定はデフォルトとした。スループットについてはリモート PC 側のデバッグビューを 60 fps で 60 秒間キャプチャし、フレームの切り替わりをカウントすることでスループット (fps) を計算した。実験環境は以下に示す。

- リモート PC
 - CPU: Intel® Core M-5Y70 CPU
 - メモリ: 8.00 GB

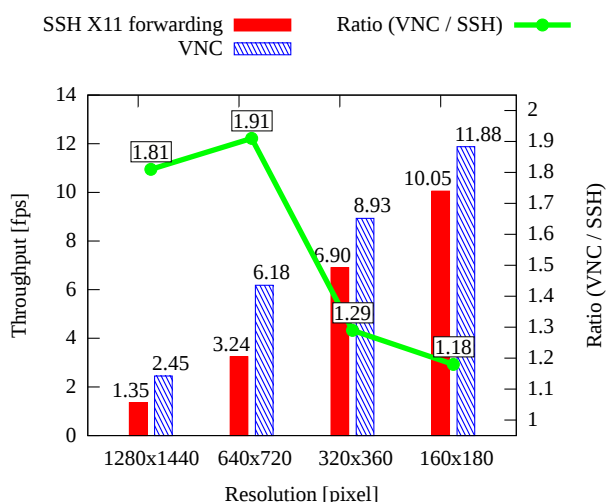


図 5 各手法のスループットと性能比

- OS: Windows 10 Pro
- VNC クライアント: TigerVNC Viewer 64-bit v1.7.0 [6]
- WLAN
 - ルータ: Aterm WX6000HP
 - FPGA-Router 間通信
 - * 帯域: 5 GHz
 - * 通信規格: IEEE802.11ac
 - * リンクスピード: 433 Mbps
 - Router-リモート PC 間通信
 - * 帯域: 5 GHz
 - * 通信規格: IEEE802.11ac
 - * リンクスピード: 866 Mbps

以上の実験によって得られた結果を図 5 に示す。SSH X11 フォワーディングと VNC のスループットは、転送する画像の解像度の低下に伴って増加することが図 5 より読み取れる。VNC の SSH X11 フォワーディングに対する性能比に着目すると、解像度が大きいほど値が大きい傾向にある。しかし、640×720 画素は 1280×1440 画素を上回り、本実験においては、640×720 画素が VNC 導入の恩恵が最も大きい結果となった。

これは VNC/RFB の通信方法が影響していると考えられる。VNC/RFB はフレーム更新時にフレーム間で変化のあった領域を検出し、矩形領域 ($x, y, width, height$, 圧縮データ) の集合のみを転送する。そのため、解像度が低いと画像中の変化しない領域が限られ、上記の VNC/RFB の効果が得られづらい。逆に、解像度が高いと変化しない領域が増加し得る。しかしながら、元々デスクトップ画面のような変化領域の少ない画像転送に優れている VNC にとって、走行中の自動運転車のような変化領域の多い画像転送に関しては高解像度であっても必ずしも VNC の恩恵は大きいとは言えない。

しかし、どの解像度においても、VNC のスループットは

SSH X11 フォワーディングのものを上回り、VNC 導入の効果は現れた。また、3 節で述べたように Zynq-Linux 内での画像取得が 30 fps だったことを考慮すると、依然として画像転送がボトルネックになっており、改善の余地がある。そのためには VNC のさらなるオプションの使用や、別もしくは独自の圧縮方法での画像転送を検討するなどが考えられるだろう。

5. 結論

我々は、自動運転システムのデバッグ環境として、VNC を用いた画像転送システムを提案、実装した。本システムは ZYBO Z7-20 という SoC FPGA ボード上に実装されている。また、ZYBO Z7-20 には Wi-Fi 接続モジュールが搭載されていないため、TP-Link AC600 という Wi-Fi ドングルを使用した。加えて、VNC サーバとして TigerVNC を使用した。本論文の評価では、解像度が高くなるにつれて、フレームレートは低下することを示した。VNC と X11 フォワーディングの性能比に着目すると、640×720 画素のときに性能比が最大になり、VNC が効果的に動作した。

本システムが遠隔の自動運転車の画像処理のデバッグを容易にすることが確認できた。今後は、スループットだけでなくレイテンシに注目した評価と、Wi-Fi 転送速度が車の位置や方向にどのくらい依存しているかの評価を行う。また、画像転送システムを活用して、自動運転システムの開発を継続していく。

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 渡邊実教授および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] 相磯秀夫杯実行委員会: 第 10 回 相磯秀夫杯 FPGA デザインコンテスト. <https://wp.rs.cs.okayama-u.ac.jp/design-contest-aiso10/>.
- [2] FPT'21: CONTENTS AND CONDITIONS. <https://parallel.auckland.ac.nz/fpt21/competition.html>.
- [3] Manabe, T., Egawa, H., Kawamata, Y., Kida, T., Tsugami, R., Kakizaki, R., Katayama, T., Tomonaga, K., Fukui, S., Yoshinaga, N., Imamura, Y., Saikai, T., Fujita, K., Matsuda, M., Miyata, K., Mori, T. and Shibata, Y.: Autonomous Vehicle Driving Using the Stream-Based Real-Time Hardware Line Detector, *2019 International Conference on Field-Programmable Technology (ICFPT)*, pp. 461–464 (online), DOI: 10.1109/ICFPT47387.2019.00093 (2019).
- [4] Richardson, T. and Levine, J.: The Remote Framebuffer Protocol, *RFC*, Vol. 6143, pp. 1–39 (2011).
- [5] Xilinx Inc.: <https://github.com/Xilinx/linux-xlnx>. (last accessed 18 Sept. 2021).
- [6] TigerVNC: <https://tigervnc.org/>. (last accessed 18 Sept. 2021).
- [7] rtl8812au: <https://github.com/aircrack-ng/rtl8812au>. (last accessed 18 Sept. 2021).