

第8回 相磯秀夫杯 FPGA デザインコンテスト論文

チーム名	SAKVEI	氏名	備後 博生
------	--------	----	-------

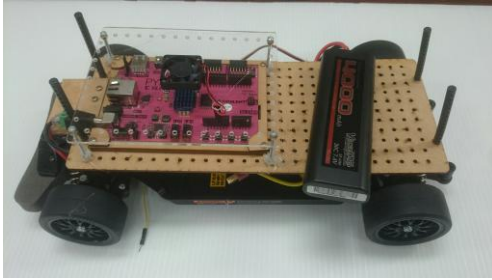


図 1: 使用する車体

1 機体とシステムの構成

車体は図1に示すようにタミヤ社製のRCカーを使用した。車体に搭載するFPGAボードはPYNQ-Z1(以降PYNQと表記)を使用した。このFPGAを採用した理由として、まずはPython言語で回路設計が可能のためである。また、OpenCVが標準で入っていることも採用した理由となっている。PS上ではあるが容易に画像処理が行える。今回は白線検出に用いている。PYNQではBitstreamのことをOverlayと呼んでいるため、以降はOverlayで書き表す。

使用したセンサはカメラ(Go Pro Hero4)のみである。カメラからHDMI経由でFPGAに入力する。

2 シミュレーション環境の構築

走行する道路を実際に設営するのは大変であるため、シミュレーション環境を構築した。Gazeboというシミュレーターを使用した。信号や障害物などのオブジェクトの色付けはしておらず、モノクロ画像として扱ってシミュレーションを行った。

3 画像の取得

車体に搭載しているカメラからHDMI経由でFPGAに繋げて画像を取得できるようにした。PYNQが標準で用意しているビデオ処理用のOverlayを使用しており、USBやPmod経由の接続よりも高画質でフレームレートは高速と思われる。なお、使用するにはコツが必要であった。Overlayの段階ではカメラはPYNQに繋げず、タイムアウトしてから繋げるとカーネルパニックを起こさずに使用できる。

4 白線検出

白線検出するため、エッジ検出(図2)とハフ変換によって車線の左側だけ検出を行なっているが、検出しにくい場合ハフ変換以外の手法でも模索している。なお、現在は全てPSで画像処理している。

5 人間検出

人間検出を行うため、yolo[1]をノートパソコン上で試しに行なった。結果が図3である。人間検出できており、領域も適切であるため車から人間までの距離も推定できる。しかし、PYNQのPL上に実装できていない。

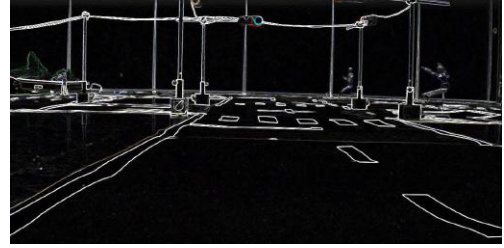


図 2: エッジ検出

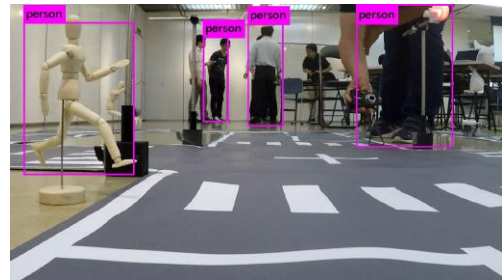


図 3: yoloによる人間検出

参考文献

- [1] YOLOv3: An Incremental Improvement, Redmon, Joseph and Farhadi, Ali, arXiv, 2018