(305) 通信　その他

オブジェクト検出機能付きラジコンクローラーの製作実習

Production training of radio control crawler with object detection function

柿本 侑亮 † 枡見 健吾 † 磯崎 稜太 †† 都築 伸二 †† 杉本 大志 ††
Yusuke Kakimoto † Kengo Masumi † Ryota Isozaki †† Shinji Tsuzuki †† Masashi Sugimoto ††
† 愛媛大学　工学部　電気電子工学科
†† 愛媛大学　大学院理工学研究科　電子情報工学専攻

1　まえがき

筆者らの研究室では, 卒業研究の導入教育で Arduino と RaspberryPi(以下, RasPi とする．) の基礎的な制御方法について学ぶ. 導入教育では Arduino を用いることによって, 各種センサーの使い方やクローラーのモーター制御などについて学びことができ, RasPi を用いることによって, コンピュータ間通信の各種プロトコルについて学んだり, 学習済みモデルを用いたオブジェクト検出を行う経験が得られる. 筆者らのグループはこれらの応用実習として, 所定のオブジェクトを検出すると停止するラジコンクローラーを製作した. この応用実習では, グループ開発として, コンピュータ間通信, オブジェクト検出, 各種センサーやモーターの制御といった要素技術を有機的に組み合わせて 1 つのシステムを開発することとなる. この点において, 卒業研究で必要となる事前調査と計画立案, システム要件会議, システム開発, 各種テストと振り返りといった一連の流れを主体的に体験する意義を持つ.

2　クローラーの製作方法

製作したクローラーは, コントローラー操作によって前後左右に進み, 前方のカメラで人物を検出すれば, コントローラー操作に関係なく停止する. 製作したクローラーシステムのブロックを図 1 に示す.

実線矢印で示しているのがコントローラー操作部分であり, 点線矢印で示しているのがオブジェクト検出に関する部分である. 次項ではこれら 2 つの部分について具体的に述べる.

2.1　コントローラー操作部分

図 1 の通り, コントローラーにある 4 つのタクトスイッチでクローラーの進行方向を前後左右に制御する. またボリュームでクローラーの速度を制御する.

進行方向制御について述べる. どのタクトスイッチが押されているかにより, 状態 0 から状態 4 までの番号を割り振っている. 状態 0 はどのタクトスイッチも押されていない, または複数のタクトスイッチが同時に押されている状態であり, このときクローラーは停止するように制御する. つまり, 状態 1 から状態 4 は各 4 つのスイッチがいずれか 1 つだけ押されている状態であり, それぞれ前後左右の進行方向に対応している.

速度制御用ボリュームで分圧した電圧値を 0 から 1023 までの値に変換して, その値により DC モーターの PWM 値を制御する.

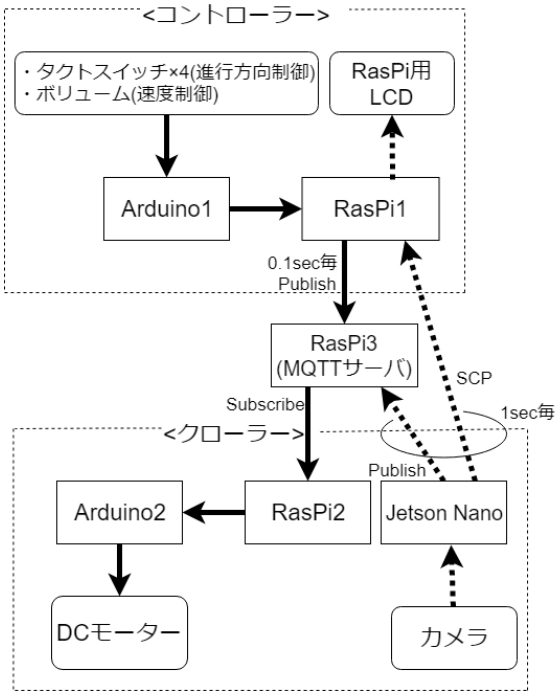


図 1: クローラーシステムのブロック

コントローラー内の Arduino1 ではタクトスイッチとボリュームの状態値を 0.1 秒ごとに読み取り, その値を文字列にしたコマンドを RasPi1 へシリアルで送信する. 例として, どのスイッチも押さずにボリュームの値が 1023 であれば, “s0p1023un” のコマンドが RasPi1 に送信される. “s0” がタクトスイッチの状態, “p1023u” がボリュームの状態とその区切り文字, “n” がコマンドの区切り文字である.

RasPi1 では Arduino1 からシリアルで受信したコマンドを, トピック名 “sw” として RasPi3 で動いている MQTT(Message Queue Telemetry Transport) サーバーへ Wi-Fi でパブリッシュする. クローラー側の RasPi2 はトピック名 “sw” のデータを Wi-Fi でサブスクライブする. 受け取ったデータを Arduino2 へシリアルで送信する.

Arduino2 では受け取ったコマンドからスイッチとボリュームの状態値を切り出し, 値に応じてモーターを制御する.

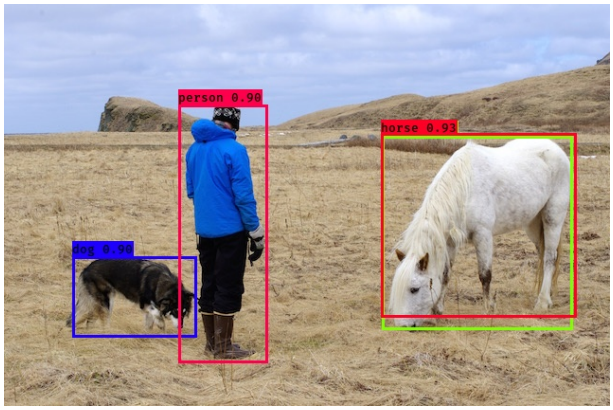


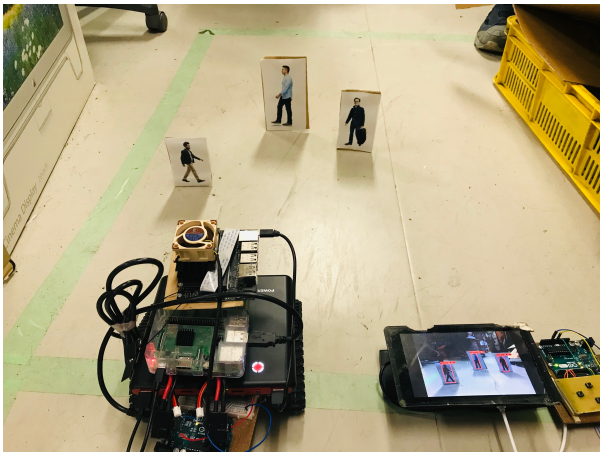
図 2: YOLO の出力画像例

2.2 オブジェクト検出部分

図 1 に示した Jetson Nano は, クローラー前方で撮影された画像に何が写っているか解析する. Jetson Nano は 128 基の CUDA コアを持つ Maxwell アーキテクチャ GPU を搭載する SoM（System on Module）で, CPU には Arm Cortex-A57 のクアッドコア CPU, メモリは LPDDR4 を 4GB 搭載している [1]. この Jetson Nano でリアルタイム物体検出アルゴリズムである YOLO を用いてオブジェクト検出を行う. YOLO はオブジェクトの検出とクラス分類の 2 つのプロセスを同時に行う [2]. 例として 3 つのオブジェクトを検出した画像を図 2 に示す. 図 2 のようにオブジェクト毎に画像を区切り, そのオブジェクト名を表示した画像を出力する. オブジェクト名の横にある数字は検出確度である.

Jetson Nano はクローラー前方で撮影された画像をおよそ 1 秒おきに YOLO へ渡す. 検出が終了したら図 2 で示したような出力画像を, コントローラー側の RasPi1 へ SCP プロトコルにより Wi-Fi でアップロードする. アップロードは 1 秒ごとに上書きする. RasPi1 ではアップロードされた画像ファイルの更新時刻を監視するプログラムを動かしていて, 更新されたときその画像を RasPi1 の LCD に全画面表示する.

クローラーは前方に人がいると認識している間, リモート操作に関係なく停止し, 人がいないと認識するとリモート操作が可能となる. 具体的には Jetson Nano で動かしている YOLO の検出したオブジェクトに “person” がなければ, “s5p0un” のコマンドを MQTT サーバー経由で RasPi2 へトピック名 “sw” にてパブリッシュする. YOLO の検出したオブジェクトに “person” がある場合, “s6p0un” の文字列を同様にしてパブリッシュする. “s5” は人がいないという状態であり,“s0” から “s4” に応じてモーターを動かす. “s6” を受け取ると, 直ちにモーターを停止し, 次の “s5” を受け取るまで “s1” から “s4” は無視する.



(a) クローラー (左) とコントローラー (右)

(b) コントローラーの表示画面

図 3: 動作確認の様子

3 動作結果

製作したクローラーとコントローラーの写真を図 3(a) に示す. 図 3(a) のように, クローラーの前方に 3 人の写真を立てた時の, コントローラーの LCD の表示画像を図 3(b) に示す. LCD には人物部分が “person” と書かれた赤い枠で囲まれ, 3 人を認識していることがわかる. このときコントローラーのタクトスイッチを押したがクローラーは停止し続け, 人の写真をクローラーのカメラに映らない場所へと移し, タクトスイッチを押すと操作が可能となることを確認した.

4 検出結果に基づく動作の変更例

応用実習として製作したクローラーは検出結果に人がいるとモーターを停止させたが, 人以外の任意のオブジェクトを検出することによりクローラーの動作を変更することも可能である.

筆者らの研究室では高校生向けのオープンキャンパスにて本稿で製作したクローラーを高校生に操作してもらうという体験学習を行った. 体験学習ではクローラーに新しく赤, 青, 白色の 3 つの LED を取り付け, これらの LED はそれぞれ指定したオブジェクトを検

出すると点灯する．高校生には LED が点灯するオブジェクトの写真がちりばめられたコースでクローラーを操作してもらい, カメラでオブジェクトを検出して 3 つの LED を点灯させた状態でゴールするという体験学習である．

5 おわりに

導入教育の応用実習として, オブジェクト検出機能付きラジコンクローラーを製作した．作成した通信プログラムでは, コントローラーのタクトスイッチとボリュームのデータをクローラーへ送信する手段として MQTT プロトコルを用いた．またクローラーから検出結果の画像をコントローラーへ送信する方法として SCP プロトコルを用いた．

MQTT プロトコルを用いることによって, メッセージの送信者である Publisher, 受信者である Subscriber や中継サーバーである Broker について学び, それらの役割を果たす Python プログラムを作成することができた．また, SCP は SSH の機能を使ったファイル転送を行うコマンドであり, クローラー・コントローラー間の SSH 接続において公開鍵認証を使用しているので, その仕組みや設定について学ぶことができた．

コントローラーの LCD に表示される検出結果の画像は本稿では 1 秒おきに更新した．Jetson Nano はクローラー前方の画像をおよそ 0.3 秒でオブジェクト検出して YOLO の出力画像を生成したが, その画像をコントローラーまで送信するのにおよそ 0.6 秒かかったためである．クローラーの速度が速いので 1 秒より速く更新する必要があり, SCP より速い通信プロトコルを探すことが今後の課題である．

参考文献

[1] 松本貴志, “いま話題の Jetson Nano、予想以上に AI 処理が速い,” モノづくり総合版メルマガ 編集後記.
https://monoist.atmarkit.co.jp/mn/articles/1904/18/news043.html, 2019 年 8 月 24 日アクセス.

[2] ReNom, “物体検出 YOLO 物体検出アルゴリズムの使い方,”
https://www.renom.jp/ja/notebooks/tutorial/image_processing/yolo/notebook.html, 2019 年 8 月 24 日アクセス.