

貴方だけに懐くペットロボットの開発

システム科学技術学部 知能メカトロニクス学科

2年 山内 洸太郎

2年 四ツ家 大智

指導教員 システム科学技術学部 知能メカトロニクス学科

助教 Woo Hanwool

教授 佐藤 和人

准教授 間所 洋和

1, はじめに

私たちが「人に懐くペットロボットの開発」を研究テーマに決定した理由はペットのように飼い主に対して従順であり, 利口な動作をロボットでも表現できるのではないかと考えたからである. ペットに着目した理由としては生きているペットを途中で飼育放棄してしまう人が悲しい事であるが少なくないのが現状である. 捨てられたペット達を保護するのに税金が使われている. その点ペットロボットは日常的に餌を与える必要がないので, 飼い主の負担を減らす事ができる. 私たちはロボットを活用する事でこの課題の解決に繋がるのではないかと考えた. 近年カメラの小型化やマイクロコンピュータの性能が飛躍的に進歩したことによって処理を高速化することが可能になり, リアルタイムでの制御が可能になった. そこで今回はペットロボットにとって必要不可欠である視覚情報, すなわちカメラから受け取る画像情報やセンサー情報と一年生時の自主研究で培った機械学習の知識を基礎として研究を進めていきたい.

2, 実験方法

2-1, 実験に使用した機器

今回は構築済みのロボットTurtlebot3 waffleを使用した. 図1にロボットの全体写真を示す.

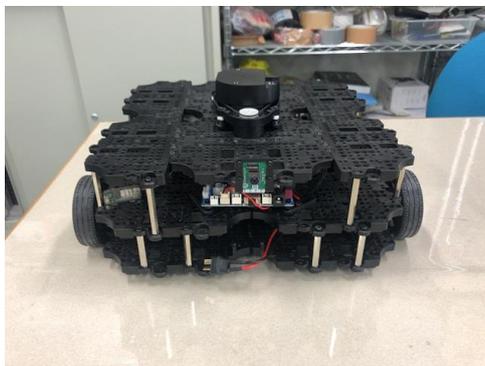


図 1 ロボットの全体写真

マイクロコンピュータは二つ搭載しており、一つは動力系統を制御するためのものとセンサー系とカメラからの情報を扱う処理を得意としているラズパイを搭載している。ラズパイはLidarからのセンサー情報とカメラの情報を送信する役割を担っており、実際の処理はノートパソコンにラズパイを通信させることで可能にした。センサー系は360° Lidar, カメラはラズパイ付属の純正カメラである。開発環境はオープンソースが豊富なROSが用意されており、基本的なコマンド, ROS対応のプログラムの記述の仕方は[1]を参考にした。

2-2, ペットロボットの動作フローチャート

ペットロボットの動作フローチャートを図2に示す。

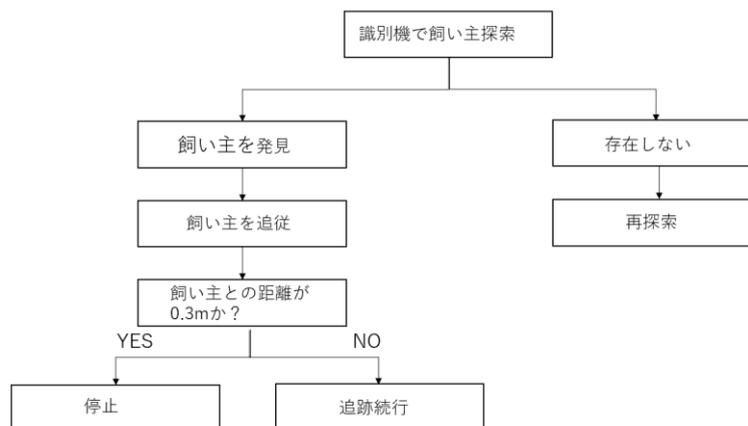


図 2 ペットロボットのフローチャート

識別機はYOLOv3を用いて作成した。追従アルゴリズムはkeras-yolo3のソースコードをROSで使用できるように手を加えて、追従アルゴリズムを付け加えた。説明は2-3節, 2-4節で示す。

2-3, 識別機の作成

2-2節のフローチャートで示したように飼い主を識別するためにkeras-yolo3を使用して識別機を作成した。元々YOLOv3は深層強化学習を用いた高速物体検出手法であり、今回はPythonで処理を記述するのでkeras実装されたkeras-yolo3を使用した。YOLOv3を使用した理由としてはリアルタイムで遅れなくロボットを制御するため、高速で処理する必要があったからである。YOLOv3についての説明は省略する。識別機を作成するには自前の画像データを用意する必要があるが、三体のフィギアとぬいぐるみを各100枚ずつ用意した。画像を学習させるには画像の中のどこに学習対象が位置しているかの座標情報をネットワークに教える必要があるため、アノテーションソフトVottでその作業を行った。図4にはアノテーションの作業画面を添付する。



図3 学習対象のオブジェクト

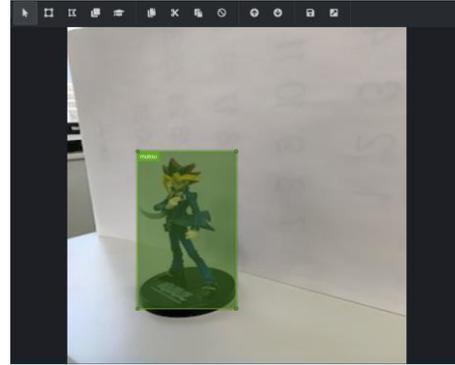


図4 Vott での作業画面

2-4, 追従アルゴリズムの作成

追従アルゴリズムはROSに対応しているPythonでコーディングした。ペットのように飼い主が移動したらそれに付いていき、飼い主が停止したらペットも止まるといった動作を参考にして作成した。今回飼い主として四つの物体の中からピカチュウを飼い主として設定した。keras-yolo3では識別した時の座標データを検出結果と一緒に出力するので、その座標データを基に追従アルゴリズムを考えた。手順としては以下のようになる。

- ①飼い主をkeras-yolo3で作成した識別器で探索する。
- ②飼い主がない場合は回転運動をさせながら探索させる。
- ③飼い主が見つかったら座標情報を配列に格納させて、回転運動を停止させる。
- ④識別した飼い主の座標がカメラの中心となるように座標の範囲毎に異なった回転運動をさせる条件分岐を行った。この時ロボットは同時にX軸方向に直進運動している。
- ⑤最後に飼い主との距離が0.3m以下になったら停止するようにLidarのrange(距離データ)を参照して条件分岐を行った。

3, 実験結果

まず2-3節で作成した識別機の検出結果を図5に示す。



図5 識別機の検出結果

少ない画像データであったが、問題なく検出することができた。

次に**2-4節**で作成したアルゴリズムを実機で実験してみた所、ノートPCが識別機の処理にかかる負荷に耐えきれない事と識別対象との距離が離れすぎてしまうと認識されにくくなってしまい完璧に追従することができなかった。

4, 考察

3節の実験結果を踏まえて考えられる改善策は三つある。一つ目は処理をしていたノートPCを新調することである。画像データを扱う重い処理であるため、GPUが非搭載のノートPCでは役不足であるのではないかと考えられる。最近ではマイクロコンピューターでもGPU搭載している製品があり、GPU付きのPCよりも安価で入手することができるので導入を検討したい。

二つ目は識別器の学習である。元々は人間を追従させる予定で学習を進めており、私たちの顔を学習させていた。しかし問題が発生してしまい方向転換したので、学習が手薄になってしまった。元々YOLOv3は小さい物体の検出は苦手としているが、学習枚数を増やす事には効果があると考えられる。そこで画像を回転、反転、フィルターを通し画像データの増しをすることで学習効率を上げる事ができるのではないかと考えられる。

三つ目は追跡アルゴリズムの改良である。現状のアルゴリズムでも問題はないが、より完璧な制御を実現させるためにはPID制御が必要不可欠であると考えられる。**2-4節**に記述したように分岐処理のみで作成した。しかしこれができるのは何回か実験して回転運動の値を詰めていくことができたからである。異なる条件下でロボットを動作させる場合はそれが難しくなってしまう。そこでPID制御、数学の恩恵を受ける事で頑丈なアルゴリズムを作成することができるのではないかと思う。

5, 結言

今回の自主研究では実際に手を動かしてロボットを動かし、制御することでROSの使い方やロボットプログラミングの基礎を学ぶ事ができた。ロボットは工学の分野では様々な技術を詰め合わせたものであり、今回はその中で視覚情報、センサーを使ったセンシングの分野に取り組んだ。他にもロボットの運動、行動を制御する分野がある。その中で極めたい分野を見つけ、今後学習に努めていきたい。

最後に、今後日本の人口は現在よりもさらに単調減少していくと考えられる。そこでロボットを活用して私たちと共存していくような世界になることが期待されるだろう。

6, 参考文献

[1] 小倉崇「ROSではじめるロボットプログラミング」工学社, 2015/06/25, 191ページ