

画像処理を用いた車両型ロボットの障害物回避

2020SC022 稲谷洋希

指導教員：大石泰章

1 はじめに

近年、自律走行技術の発展により、工場内での材料や製品の運搬や商業施設での清掃など様々な目的で自律走行ロボットが活用されているが、例えば文献 [1] にあるように複数のセンサや複雑な計算、アルゴリズムを必要としたり、障害物検知に LiDAR を使用したりする手法が多い。

これに対し本研究では、シンプルかつ効率的な手法を目指し、小型の車両型ロボットに LiDAR と比較し安価な深度カメラを一つ搭載し、直進中に前方に映る障害物を画像処理を使って自動的に検知し、避けて走行することを実現する。具体的には、深度カメラにより前方の映像を撮影し、障害物の中心座標とそこまでの距離を取得し、これにもとづいて減速と回避動作を行うことで障害物回避を行う。

2 使用機材とシステム

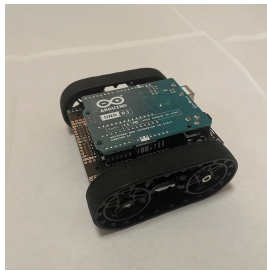


図 1 車両型ロボット Zumo



図 2 深度カメラ Intel RealSense Depth Camera D415

本研究では、車両型ロボット Zumo (図 1)、深度カメラ Intel RealSense Depth Camera D415 (図 2)、PC を用いて図 3 のようなシステムを構築した。この深度カメラはカラー画像と深度画像を取得できる。深度画像は画像内の指定した座標に映る物体までの距離情報を持ち、距離情報は 0.26m から 10m までの範囲で取得可能である。Zumo にはマイコンボード Arduino UNO R3 が搭載されており、開発環境である Arduino IDE を使用してプログラムを書き込むことで Zumo の動作を制御する。モータの速さは、-400 から 400 の範囲で指定でき、左右のモータの速さをともに 100 に設定すると、Zumo は 110mm/s で直進する。本研究では、Zumo の上部に深度カメラを取り付け、Zumo の位置から画像を取得できるようにしている。取得した画像は PC に送り、Python 上で OpenCV を使用して画像処理を行い、障害物の重心座標とそこまでの距離を取得する。OpenCV の使用は文献 [2] を参考にして行った。取得した値はシリアル通信で Zumo へ送信し、その結果にもとづいてモータを制御することで障害物を回避させる。

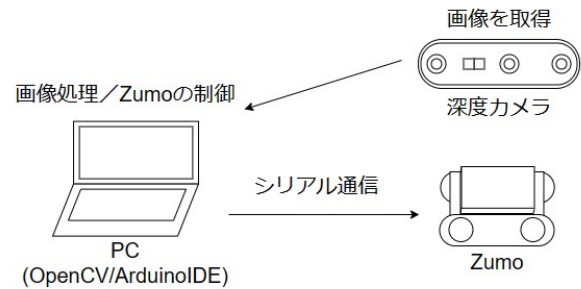


図 3 構築したシステム

3 目指す動作

停止した Zumo の 600mm 前方に障害物を設置する。障害物として横幅 118mm、高さ 109mm、奥行 508mm の段ボール箱を用いる。Zumo はモータの速さが 100 の速度まで加速した後に障害物の手前 400mm の位置を通過するまで減速しながら前進し、障害物の重心が撮影画面の中央よりも右側に映る場合、Zumo は障害物の左側から、中央よりも左側に映る場合は障害物の右側から回避する。図 4 は右側から回避するときの様子である。障害物を回避した後はもと走行していた直線上を走行することを目指す。

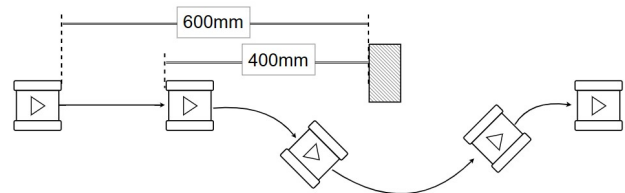


図 4 障害物が画面の左に映る場合の Zumo の軌道

4 アルゴリズム

4.1 OpenCV による画像処理

はじめに、深度カメラからカラー画像と深度画像を取得する。どちらも横 640 画素、縦 480 画素であり、画像の左上を原点として右方向に x 軸、下方向に y 軸をとる。画像中の画素の位置を x 座標と y 座標で表す。深度画像において、写った物体までの距離が 600mm 以上の範囲を求め、カラー画像における同じ範囲を画素値 0 の黒色に変更することで遠方の背景を削除する。次に、カラー画像を HSV 色空間に変換し、画素値 0 の黒色以外の物体の中で一番大きなものが障害物だと仮定し、その輪郭の重心座標を取得する。深度カメラから 400mm の位置に障害物を設置した

ときに、以上の操作を行った様子を図5に示す。図5(a)は背景を削除する前のカラー画像であり、図5(b)は背景を削除し、障害物の重心を赤点で表したものである。深度画像において、取得した重心座標に映る物体までの距離を取得する。これは障害物までの距離を表すと考えられる。その上で Arduino とシリアル通信を行って、障害物までの距離とその x 座標を送信する。以上の操作を行って障害物までの距離とその x 座標を送信することは、1.1 秒ごとに行うことができる。

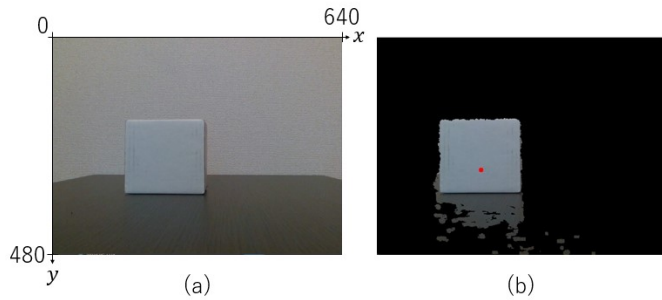


図5 背景を削除し重心を求めた様子 ((a) 背景削除する前, (b) 背景削除をし重心を赤点で表したもの)

4.2 Arduino による Zumo の制御

Arduino 側では、受信した距離と x 座標の値を、それぞれ変数 d と x とし、これらに基づいて Zumo のモータを制御することを繰り返す。

受信した距離が 400~600mm の間は、(1) 式により左右のモータの速さ v を定める：

$$v = \frac{d}{6}. \quad (1)$$

これにより Zumo 上のカメラと障害物の距離が 600mm のときに左右それぞれのモータの速さが (100, 100) となり、400mm 未満になるまでは減速しながら前進する。

障害物との距離が 400mm 未満かつ重心の x 座標が 320 以下で画面中央よりも左側にある場合を考える。このとき、まず左右それぞれのモータの速さを (150, 60) に設定して 2 秒間動かすことで、右に曲がりながら前進する。次に (60, 150) に設定して 3 秒間動かすことで、左に曲がりながら前進する。最後に (150, 60) に設定して 1 秒間動かすことで、右に曲がりながら前進する。これによって Zumo は図 4 のように走行し、回避動作をする。一方、障害物との距離が 400mm 未満であり、かつ障害物の重心の x 座標が 320 より大きくて画面中央よりも右側にある場合は、上記の左右それぞれのモータの速さを反対にして制御することで回避動作をする。

5 実験結果

図 6 に走行させた結果を示す。初期位置 (図 6(a)) の状態から前進し、深度カメラにより自動で障害物の重心まで

の距離と x 座標を取得して、減速した。その後、障害物の手前約 240mm の位置で初めて 400mm 未満の距離を受信したため回避動作を開始 (図 6(b)) して、障害物を避けて走行 (図 6(c)) し、元の直線に戻り (図 6(d)) 走行することができた。もともと回避動作の開始は障害物までの距離が 400mm 未満になった時点を目安としていたが、これが 240mm になったのは、障害物までの距離等を Arduino に送信することが 1.1 秒ごとに行えないためだと考えられる。

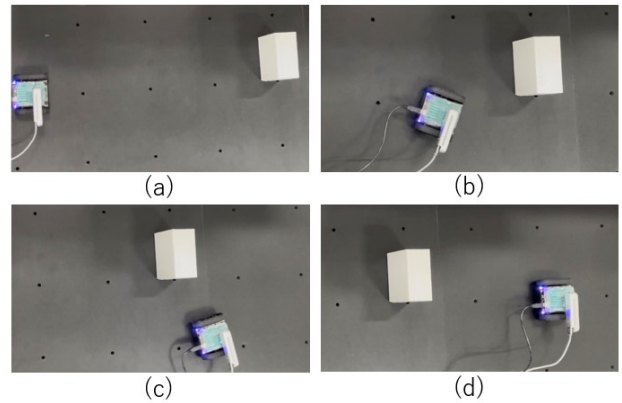


図6 動作結果 ((a) 初期位置, (b) 方向転換時, (c) 曲がる動作時, (d) 回避完了時)

6 おわりに

本研究では、車両型ロボットの前方の障害物までの距離と位置を深度カメラで検知し、一定の速度まで加速した後、障害物の手前まで減速しながら前進し、障害物の手前で進行方向を変え、避けて走行することを行った。

今回のシステムについて、見つかった課題が 2 つある。1 つ目はシリアル通信が 1.1 秒ごとでしか行えなかったため、次の値を受信するまで直前の値によるモータ制御を続けてしまうことや、Arduino 側の実行速度が遅いため、想定よりも実際のモータ制御のタイミングに遅れが生じてしまうことである。2 つ目は有線通信を行っているので通信ケーブルに引っ張られて進行方向がずれてしまうなどの影響があることである。通信速度に関しては、本来はより速い通信ができると考えられるため、原因を追究し通信速度を上げることが課題である。また、無線通信ができればシステムの改善になると考える。

参考文献

- [1] 竹下真美・日高浩一：「ステレオカメラと LRF を利用した自律走行運搬ロボット回避法の提案」, 第 65 回自動制御連合講演会講演論文集, 宇都宮, 2022 年 11 月, pp. 208–211.
- [2] 小枝正直・上田悦子・中村恭之：『OpenCV による画像処理入門』, 講談社, 東京, 2022.