E-6

火災現場における自律ドローン型レスキューロボットのためのシステム構築

Developments of Control Systems for Autonomous Drone Type Rescue Robots at Sites of the Fire

○大原惟暉[1]，池口諒人[2]，小早川耀[2]，羽多野正俊[3]
*Tadaaki Oohara[1], Ryouto Ikeguchi[2], You Kobayakawa[2], Masatoshi Hatano[2]

Abstract: The purpose of this research is developments of a firefighting drone in order to achieve rescue tasks from the air at sites of the fire. Due to influences of the intense air current in the sites, it is very difficult to maneuver a drone manually. Therefore, we develop an autonomous drone which flies to targets with avoiding obstacles including fires. In this presentation, we report developments of a control system for autonomous flights with object recognitions using the drone's ROS and OpenCV. In addition, it is shown that autonomous experimental flies were performed with the developed system.

１．緒言

本研究の目的は，火災現場においてドローンによる空中から要救助者の探索及びレスキュー活動をする方法を提案することである．火災現場では迅速な要救助者の発見及び救助が求められる．そのため，レスキュー隊員の代わりにドローンを活用することで空中から迅速に要救助者の探索を行い，２次災害を防ぐことができると考えられる．また，鎮火後の残り火の確認にドローンを活用することが考えられる．しかし，火災現場の激しい気流の影響により，ドローンの操縦は非常に困難である．そのため，本研究ではドローンに搭載されているカメラを用いて火の認識及び火災現場の進入経路計画について考える．

本発表では，基礎研究としてドローンの ROS(Robot Operating System)及び OpenCV(Open Source Computer Vision Library)を用いた物体認識による自律飛行の方法について報告する．

２．実験機及びドローンの構造

本研究で使用しているドローンは，Fig. 1 に示す Parrot 社の Bebop 2 である．

Figure 1　Parrot Bebop 2

本研究で使用した理由は，本機の制御システムは ROS 化対応しており PC から制御することが可能なためである．また，機体前方に魚眼カメラが搭載されているため，OpenCV による画像処理も可能である．

３．ドローンのマニュアル操縦

ROS はロボット開発のための様々なソフトウェアの集合のことである．　ROS が提供する主なサービスとして，デバイス制御，プロセス間通信，パッケージ管理などが提供されている[1].

Fig. 1 の Bebop 2 を制御するために bebop_autonomy というパッケージを用いた．このパッケージは Simi Fraser University の Mani Monajjemi により開発された Bebop Drone 用の ROS ドライバであり，ドローンのカメラ，GPS(Global Positioning System)などのセンサ情報の取得やカメラの向きの変更が可能である．また，離着陸，左右移動，左右旋回，上下移動などの制御が可能である．

ドローンをジョイスティックコントローラによりマニュアル操縦が可能である．その時のシステムを Fig. 2 に示す．

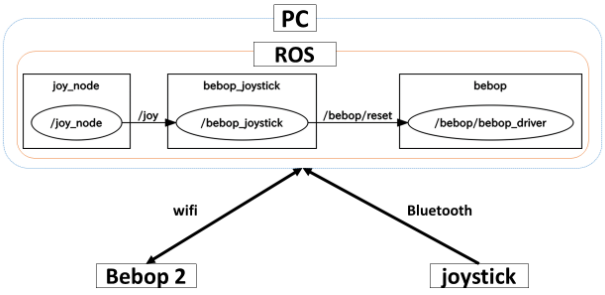


Figure 2　Manual Controls of Bebop 2 by ROS

Fig. 2 に示す様に，joystick の操作信号は Bluetooth により PC に送られ，PC と Bebop 2 間は WiFi によ

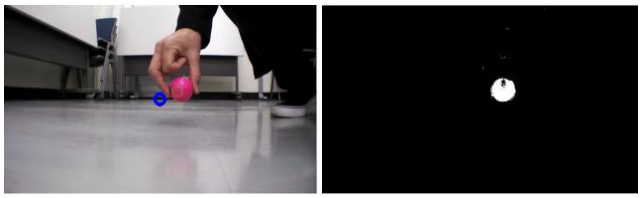
１：日大理工・大学院・精密機械　２：日大理工・学部・精密機械　３：日大理工・教員・精密機械

り通信される．後述の自律制御の際には，Bebop2 に搭載されたカメラ画像情報を用いて自律飛行し，joystick は用いられない．

４．ビジュアルフィードバックによる自律飛行

安全のため 3 章で述べたシステムを用いて離着陸のみを行い，それ以外は，ドローンに搭載されているカメラを用いて OpenCV により画像認識を行い，自律飛行させた．OpenCV とは，膨大な関数を用意した画像処理ライブラリ集である．一般的な 2 次元の画像処理，顔認識，など多様なアプリケーションを開発できる関数群が用意されている[2]．

ROS により取得した Fig. 3 の左図に示すようなカメラ映像を RGB 画像から HSV 画像に変換させた．HSV 変換した理由は RGB 画像では原色の組み合わせで色を表現しているが，各要素を変動させた場合に色の変化が分かりづらく，細かく調整したい場合には不向きなためである．その後，色の範囲を指定し，Fig. 3 の右図の様にマスク画像を生成した．



RGB　　マスク

Figure 3　RGB image and Mask image

指定した範囲内の色は白で表示され，それ以外は黒で表示される．指定した色の物体を認識し，物体を追従するように自律飛行させた．

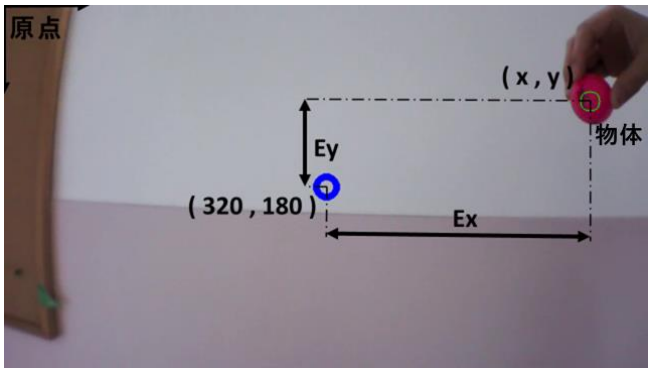


Figure 4. Drone's Camera Picture

Fig. 4 の様にカメラ映像の中心座標(320，180)と物体の座標(x，y)から誤差 Ex，Ey を求めた．

$$Ex = x - 320 \quad (1)$$

$$Ey = y - 180 \quad (2)$$

式(1)，(2)とゲイン k より速度制御を行い，自律飛行させた．速度 Vx，Vy を以下に示す．

$$Vx = k * Ex \quad (3)$$

$$Vy = k * Ey \quad (4)$$

これらを用いて，飛行実験を行った．物体をカメラの中心の右側に置き，60 秒間，式(3)の P 制御をさせた．今回は k = 0.05 に設定した．この実験を 3 回行い，Fig. 5 のグラフにまとめた．

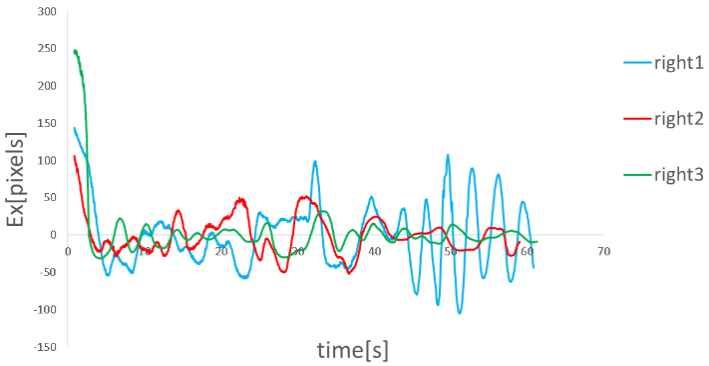


Figure 5　Result of Autonomous Flight Experiment of Drone

Fig. 5 の実験結果より，時間の経過と共に誤差 Ex が 0 に近づいていることより，ドローンが物体に追従していることが分かる．また，大きな振動を示した原因として，ゲイン k の値の設定が大きすぎたことと，ドローンによる気流の影響が考えられる．

５．結言

ドローンの ROS 及び OpenCV を用いた物体認識による自律飛行の方法について述べた．

今後，ドローン搭載のカメラを用いて火の認識及び火災現場の 3 次元マップを生成し，ポテンシャル法により進入経路計画の方法を検討する予定である．

６．参考文献

[1] 小倉崇:「「ROS」(Robot Operating System)とは」, ROS ではじめるロボットプログラミング，pp.8-11，2015 年.
[2] 北山洋幸 :「OpenCV 概要」，さらに進化した画像処理ライブラリの定番 OpenCV3 基本プログラミング，pp.1-2，2016 年.