

ドローンの自動追従

1. 研究テーマ概要

Tello のドローンに人の顔を認識させ自動で追従するプログラムを作成する

2. 使用機器・使用ソフトウェア

Tello ドローン

パソコン

Python

OpenCV (画像認識ソフト)

Visual Studio Code

3. 研究過程

1 学期

課題研究最初の時間は、ドローンをどのような動きをさせて飛ばすのかを考え自動追従を選びました。ドローンを飛ばすプログラム言語は Python のほかに Scratch や Swift などの言語があり色々調べた結果、使いやすく簡単な Python を選びました。そのほかにも OpenCV や Visual Studio Code などのドローンを飛ばすために必要なものをインストールしプログラムを作成しました。

2 学期

2 学期の課題研究の時間は、Chat GPT を使用して 1 学期のプログラムの改良をメインに行い、自動追従をするプログラムを完成させることができました。さらにプログラム完成後には take off の時の高さの調整や、追従性能を向上させるためのプログラム改良を行いました。

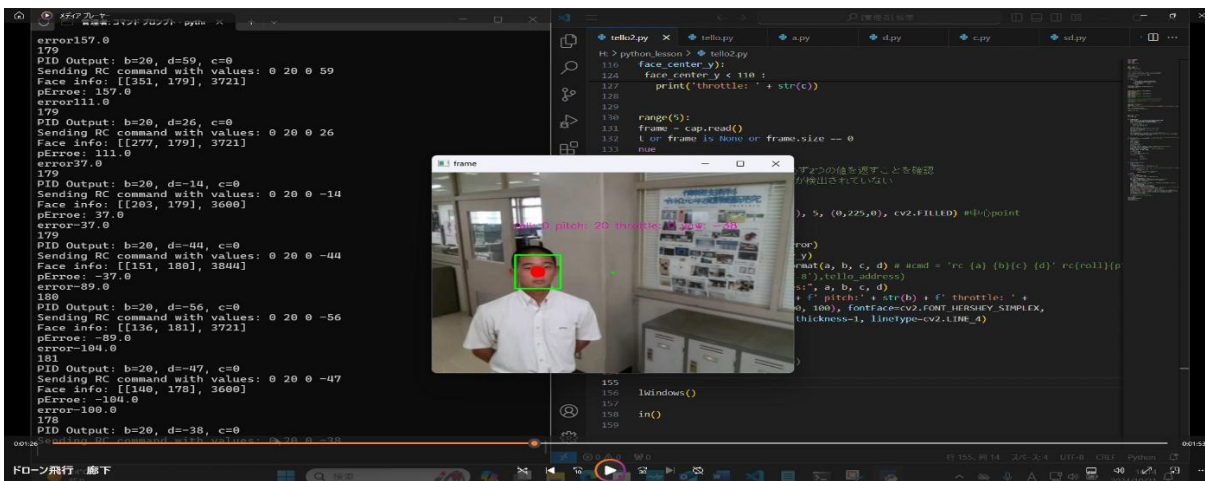
3 学期

課題研究発表会の要旨、PowerPoint、レポートの作成をしました。

4. 研究成果

Python を使ってプログラミングによりドローンを飛ばし、自動追従させることができました。

実行画面 (写真 1)



プログラム画面の一部(写真2)

```
a = b = c = d = 0 # #cmd = 'rc {a} {b} {c} {d}' rc{roll}{pitch}{throttle}{yaw}
def TrackFace( info, w, pid, pError): # x軸に対するPI制御
    global b, d
    # pError = error
    # pid_output = d
    area = info[1]
    x, y = info[0]
    b = 0 #bを初期化
    #yaw(左右運動)算出用
    error = x - w/2*1.1
    d = 0.8 * d + 0.2 * (pid[0] * error + pid[1] * (error - pError)) #PID制御のうちPI制御
    d = int(np.clip(d, -100, 100)) #RCコマンドのリミッター
    print('pError: ' + str(pError))
    print('error' + str(error)) #pitch(前後運動)算出用
    if area > 6200 and area < 6800:
        b = 0
    elif area > 6800:
        b = -20
    elif area < 6200 and area != 0:
        b = 20
    if x == 0:
        d = 0
        error = 0
        print(b, d)
    return error

def throttle(face_center_y):
    global c, img
    c = 0
    print(face_center_y)
    if face_center_y > 180 and face_center_y < 40:
        c = 0
    elif 280 > face_center_y > 180 + 10 : #frame h:360の1/2=180 顔の下降 (座標が大きくなる)
        c = -30
    elif 50 < face_center_y < 110 :
        c = 50
        # c = int(np.clip(c, -100, 100))
        print('throttle: ' + str(c))
    return

while True:
    for i in range(5):
        ret, frame = cap.read()
        if frame is None or frame.size == 0:
            continue

        cv2.circle(frame, (int(480), int(360)), 5, (0, 225, 0), cv2.FILLED) # 中心point
        img = cv2.resize(frame, (w, h))

        img, info = findFace(img) # ここでアンバック
        print("Face info:", info)
        # info が正しい形式であるかをチェック
```

5. 今後の課題

エラーの修正を最初独力で実施していましたが、時間がかかりすぎて途中より ChatGPT を使用してプログラムを完成させました。自分の力だけで完成させることができなかったので、最初に python についてもっと勉強したほうが良いと思います。

6. まとめ・感想

ドローンをプログラムで動かすのは難しそうだと思っていましたが、自分でドローンや python について調べ、プログラムを組み立てていく中で、自分のプログラミング力が少しずつ上がっていくことを実感でき、プログラミングの面白さに気づくことができました。