第3章 PythonとOpenCVを使って

ドローンの航路を決める画像処理プログラミング

橋口 宏衛

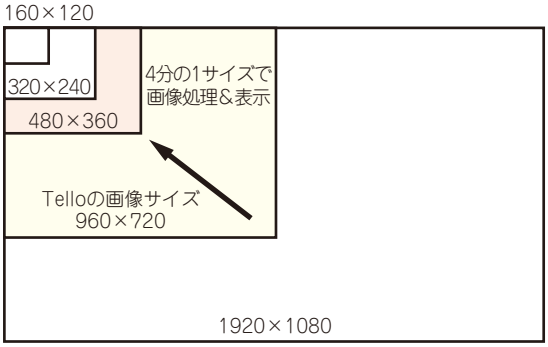

図1 Telloが搭載しているカメラの画像サイズ

PythonとOpenCVを使って，Telloのカメラ画像に画像処理を施します．特定の色の物体をTelloが追跡する，ビジュアル・フィードバックも試してみます．

ラズパイで画像処理プログラミングするときの注意点

Telloのカメラ性能はHD画質720pですが，解像度は16：9の1280×720ではなく，4：3の920×720です．最近のフルHD（1920×1080）や4K（3840×2160）に比べれば画像サイズは小さいです．

Core i7クラスのCPUと大容量のメモリを持ったPCであれば，920×720のサイズでもリアルタイムに画像処理結果を画面表示できます．しかし，ラズベリー・パイ4のグラフィックス処理の能力では，このサイズの画像では，激しい遅延が起こってしまいます．

遅延が大きいとリアルタイムな画像フィードバック処理はうまくいきません．認識してから行動を開始しようとしても，既にドローンの機体は大きく動いてしまっているからです．例えば2秒前の画像で障害物を認識したとしても，現実では既に衝突しているでしょう．

本稿ではラズベリー・パイの性能を考慮し，処理する画像を元画像に対して4分の1の480×360にします（**図1**）．このサイズでもCPU負荷率は90%を超えてきます．

実際のロボットでも，一昔前は320×240で画像処理することは普通でした．後述するDonkeyCarでも160×120の画像を集めて機械学習を行っています．480×360でもまだ大きい方だと思っておいてください．

図2 ラズベリー・パイのWi-Fi設定

画像処理をしているとCPU負荷率が90%近くになるので，小さなヒートシンクでは温度が80℃を超えてしまうこともあります．この温度では，半導体の寿命も縮んでしまいます．ラズベリー・パイを強制冷却する空冷ファンは必須です．ちなみに飛行するTelloの吹き下ろし風を当てても，結構冷却されます．

1：まずはスケルトン・プログラムで動かしてみる

Telloを自律移動させるには，カメラ映像を画像処理し，処理結果をTelloの動きにフィードバックさせる必要があります．すなわち入力，制御，出力が必要になります．この処理のファースト・ステップとして，スケルトン・プログラムを作ります．

スケルトン・プログラムとは，簡単に言えば，DJITelloPyのクラスを使ってOpenCVで処理するための準備が完了した最低動作プログラムです．

2値化，ラベリング，顔検出，ARマーカなどの発展的な処理は，スケルトン・プログラムをベースに作ります．

● ラズパイとドローンTelloをWi-Fi接続する

ラズベリー・パイがWi-FiでTelloに接続していることを確認します．

図2を参考に，ラズベリー・パイに192.168.10.x（xは任意の数字）というIPアドレスが割り振られたことを確認します．