

光の照度が異なるライン上でドローンの自律飛行を可能にする ラインカラー認識方法の提案と評価

2020SC085 鈴木龍也 2020SC102 上島快斗

指導教員：石原靖哲

1 はじめに

近年、あらゆる場面においてロボットによる自動化が進められている。[1]によると、産業用ロボットの導入台数が2020年は39万台であり、2021年は52.6万台に増加している。[1]は、2026年に718万台と予測している。世界中でロボット化が進む中、農業や物流業界で自律移動するロボットが採用される。自律移動は、ロボット自身がセンサにより環境をリアルタイムで認識し、あらかじめ与えられた移動アルゴリズムや情報に基づき、判断することで可能となる。人間の作業では付加価値を生まない様々な業界で労働力不足解消が期待される。

現在実用化されている自律移動ロボットの多くは、車両型のロボットである。車両型ロボットは地面に接しているため、安定した移動が可能で、重量のあるものを運搬するのに適している。しかし、不均一な地形や障害物に制約を受けるといった問題がある。そこで本研究では、三次元空間を移動することができるドローンに注目する。ドローンは、飛行時間が限られていることや天候による制約がある一方で、不均一な地形や障害物による影響を受けにくいという利点がある。

ドローンを移動させるには、移動する方向を入力する方法の他、GPSを用いた方法や物体を追尾する方法などがあるが、本研究では、ライントレースによる自律移動に注目する。ライントレースとは、ラインに沿ってドローンが飛行することであり、あらかじめ指定した色のラインを辿っていくものである。位置情報や周囲の環境情報を必要とせず、ラインのみを必要するため、実装が比較的容易である。しかし、同じ色のラインにおいても日光や照明などの光の当たり具合により、色が大きく変化してしまい、検出できなくなってしまうおそれがある。そこで、本研究では、HSL変換を用いて光の検出をし、さらにいくつかの異なる光の強度に基づいたしきい値を用いることで、光の当たり具合に左右されないラインの認識方法の提案と評価を行う。

2 先行研究

佐藤[2]は、色、太さ、素材が異なる20種類の粘着テープの中から、適合率が0.96以上の15種類の粘着テープを選定し、さらにその15種類の中から再現率の最も高い、幅5cmでツヤのない赤色の粘着テープを実験に採用している。この実験において、同じ色であってもより太く、ツヤの無い線の方がより高い適合率、再現率を得ている。ここで、適合率とは、正事例と予測したもののなかで真の値

が正事例の割合を表す指標であり、再現率とは、真の値が正事例のものなかで正事例と予測した割合を表す指標である。それらは、以下の式で表される。

$$\text{適合率} = \frac{\text{正解領域} \cap \text{抽出領域}}{\text{抽出領域}} \quad (1)$$

$$\text{再現率} = \frac{\text{正解領域} \cap \text{抽出領域}}{\text{正解領域}} \quad (2)$$

この結果はツヤによる光の反射で本来とは別の色に見える部分があったり、細い線によって、ドローンのカメラが、指定した色を認識できる面積が少なくなっているからであると考えられる。また、佐藤らの研究では、ライン上に障害物があり、ラインが隠されている状態でも、正常に動作していた。しかし、ここで二値化された画像は、ラインが遮断されてしまっている。これは、光がラインに当たることによる色の変化からも想定される。そこで、本研究では、光の当たり具合によって再現率が変わらないラインの認識方法の提案と評価を目指す。

3 ライントレース

ライントレースは、ロボティクスや自動制御の分野でよく用いられる技術の一つである。この技術では、ロボットや自動車などの移動体が、あらかじめ設定された経路や線に沿って移動するように制御される。カメラを用いたライントレースでは、取得した画像から画像処理技術を活用して線を検出し、ロボットの制御に利用する。ドローンにおけるラインの認識も同様に、内蔵されたカメラで動画を撮影し、それを1フレームごとに、HSV変換、二値化、ラベリング、面積と重心の計算をすることで行う[3]。

3.1 HSV変換

HSV色空間は色相(Hue)、彩度(Saturation)、明度(Value)の3要素からなる。色相は赤、緑、青などの色の種類を定義する要素である。色相環であるため、0-360度の範囲で表される。彩度は、色相で定義された色の濃さを定義する要素である。0-100%の範囲で表され、値が大きくなるにつれ、色が濃くなる。明度は、色の明るさを定義する要素である。0-100%の範囲で表され、0%は黒になり、値が大きくなるにつれ、色が明るくなる。RGB(赤、緑、青)空間では3色を重ね合わせて表現するため、特定の色の抽出が困難である一方で、HSV色空間では、それぞれの値の範囲を指定することにより、特定の色を抽出しやすいという特徴がある。RGBをHSVに変換するとき用いる式は(3)、(4)、(5)式である。 R 、 G 、 B には各要素

の値を代入し, max , min には, R , G , B の各要素の最大値, 最小値を代入する.

$$H = \begin{cases} 60(G - B)/(max - min) & max = R \\ 60(B - R)/(max - min) + 120 & max = G \\ 60(R - G)/(max - min) + 240 & max = B \\ 0 & min = max \end{cases} \quad (3)$$

$$S = \frac{max - min}{max} \quad (4)$$

$$V = max \quad (5)$$

本研究の画像処理で用いる OpenCV では, H , S , V それぞれの値は, 0-180, 0-255, 0-255 の範囲でしきい値を設定し, 特定のラインの色を抽出する. そのため, H は 1/2 倍, S は 255 倍する.

3.2 二値化

画像処理において, カラー画像は色の情報があるため, 情報量が多くなってしまい, 処理が遅くなる. そこで, カラー画像に対して, 情報量が 0 と 1 の 2 つのみとなる二値化を用いる. これにより, 処理コストが少なく, 画像処理が高速になる. 特にライントレースのように, 映像から 1 フレームごとに特定の色のみを抽出する必要がある場合には, この方法が有効である. ドローンのカメラで撮影した映像に対して, 上記で述べた HSV 色空間で特定の色を抽出した後, 二値化処理をすることで, トレースをするラインとそれ以外の白黒画像へ変換が可能である.

3.3 ラベリング

ラベリングとは, 上記の方法で二値化された白黒画像において連結している画素を 1 つのグループとして, 複数の領域をまとめ, 各グループに 0, 1, 2, ... と識別番号を与えることである. 画像の黒い部分の番号を 0, それ以外の連結部分ごとに 1, 2, ... と番号を割り振る. 抽出する色以外の部分である, 黒い部分は, ライントレースでは不要な部分であるため, データの削除を行い, ラベルの個数を一つ減らす. 次に, 面積が最大となっているグループの x 座標, y 座標, 幅, 高さ, 面積, 重心座標 mx , my を取得する. これらの処理により, 検出したいものをデータとして認識することができる. そして, そのデータから得られた重心座標に向かって飛行することでライントレースが可能となる [4].

4 提案する手法

本研究で提案する手法は次の 3 通りである. 手法 1 は, HSV 変換し, 光の照度ごとに設定されたしきい値を用いて二値画像を得てそれらを合成する方法である. 手法 2 は, HSV 変換し, 光の照度ごとに設定されたしきい値を用いて二値画像を得てそれらを合成する方法である. これは光が強く当たっている部分に対して有効であると考えられる.

法 3 は, HSV 変換と HSL 変換の併用によって画像処理を行う方法である. どの手法でも二値画像を合成するため, 1 色のラインに対して照度の異なる複数の光に対応できると考える.

4.1 手法 1: HSV 変換のみによる二値画像の合成

HSV 変換は, 従来ライントレースを行う際に用いられる手法である. この手法では, 3 通りの異なる照度の光を当て, それぞれの場合における HSV の値を求め, そこからしきい値を設定する. そこで得られた 3 種類のしきい値それぞれを用いて, 画像の二値化を行い, 検出された領域の和をとってからラベリングする.

4.2 手法 2: HSL 変換のみによる二値画像の合成

この手法についても, 4.1 節で述べたのと同様に, 3 通りの異なる照度の光を当て, それぞれで得られた二値画像の合成を行う. ここで, HSL とは, 「色相 (Hue)」「彩度 (Saturation)」「輝度 (Lightness)」の 3 要素で表現する方式である. 輝度は, 明度同様に色の明るさを定義する要素である. HSV 色空間では赤, 緑, 青の原色が明度 100% となっているが, HSL 色空間では, 原色の輝度は 50% になる. そのため, 光が当たっているとき, 輝度は光が当たっていない場合より大きくなり, 光を検出することができる. ここで, RGB を HSL に変換するとき用いる式は (6), (7), (8) 式である. R , G , B には各要素の値を代入し, max , min には, R , G , B の各要素の最大値, 最小値を代入する. また, 式 (6) において, マイナスの値が出た場合は, 360 を足して正の値にする.

$$H = \begin{cases} 60(G - B)/(max - min) & max = R \\ 60(B - R)/(max - min) + 120 & max = G \\ 60(R - G)/(max - min) + 240 & max = B \\ 0 & min = max \end{cases} \quad (6)$$

$$L = \frac{(max + min)}{2} \times \frac{100}{255} \quad (7)$$

$$S = \begin{cases} \frac{max - min}{max + min} \times 100 & 0 \leq L \leq 50 \\ \frac{(max - min)}{510 - (max + min)} \times 100 & 51 \leq L \leq 100 \end{cases} \quad (8)$$

4.3 手法 3: HSV 変換と HSL 変換の併用による二値画像の合成

正確な二値化を目指すために, HSV 変換で得られた合成画像と, HSL 変換で得られた合成画像の共通領域をとってからラベリングする方法である. この手法では, それぞれを単独で適用した場合に見られる不具合を補完できると考える.

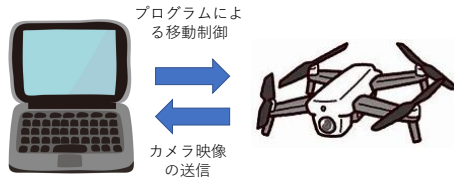


図1 システムの全体構造



図2 ラインとライトの位置関係



図3 粘着テープの一部を照らす様子

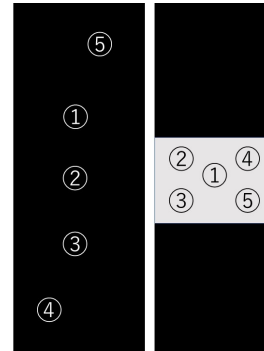


図4 ラインの測定箇所

5 実験

5.1 実験環境

ドローンのシステム全体の構造を以下の図1である。本研究では、Ryze Technology 社の Tello というドローンを使用する。Tello のカメラ性能は HD 画質 720p である。また、解像度は、16:9 の 1280 × 720 ではなく 4:3 の 920 × 720 である。カメラが前方についており、ライン以外も二値化してしまうため、本研究では、画像サイズを (0, 250)–(479, 359) にする。

Tello は、小型かつ軽量であり、複雑なアルゴリズムを実行するための十分な計算能力を持っていない。そのため、Wi-Fi を介して PC と通信を行い、PC からコマンドを受信することで制御が可能となる。実験は、安全のため、ドローンにプロペラガードを装着し、縦 2.5m、横 2.5m、高さ 2.65m のテント内で行う。テント内では照度が低くなってしまうため、工場内での照度を想定し、300–500lx となるようにライトを設置する (図2)。

5.2 実験手順

ライントレースで使用するしきい値の設定を行う。白、黒、赤の3種類の粘着テープを用意し、Tello で撮影する。撮影した画像に対して、[5] のツールを用いて、RGB を測定し、式 (3), (4), (5) を用いて HSV 変換し、ライントレースに用いるしきい値を求める。そして、それぞれの最大値と最小値をしきい値とする。次に、光が当たっている部分において、3種類のラインに対して測定した RGB か

ら弱、中、強の HSV を求める。また、HSL も同様に式 (6), (7), (8) を用いて求める。

光は、各3種類の粘着テープに、図3のように粘着テープの一部を照らし、約 1000lx(弱)、約 4000lx(中)、約 10000lx(強) の照度となるようにして、Tello で撮影する。撮影した画像に対して、[5] のツールを用いて、RGB を測定する。測定する箇所は、光が当たっていない場合は、図4の左側の通りであり、光が当たっている場合は、図4の右側の通りである。そして、求めたそれぞれの HSV 及び HSL の最大値と最小値をしきい値として設定する。

求めたしきい値を用いて、次の実験手順を白、黒、赤の各粘着テープにおいて実行する。まず光が当たっていない場合のしきい値を用いて二値化 (図5左上) を行い、再現率と適合率を求める。次にそれぞれの提案手法を用いて図5左上の画像と、光の当たっている部分をいて二値化した画像 (図5右上) を合成して、(図5左下) の画像を作成し、再現率と適合率を求める。

5.3 実験結果と評価

初めに測定した HSV のしきい値のみを用いて行った結果を調整前、これと弱、中、強の HSV のしきい値を用いて行った結果を手法1、同様に HSL のしきい値を用いて行った結果を手法2、HSV と HSL のしきい値を併用して行った結果を手法3として、得られた再現率を表1、表2、表3に、得られた適合率を表4、表5、表6に示す。

評価は、調整前と提案手法全体との比較、提案手法同士

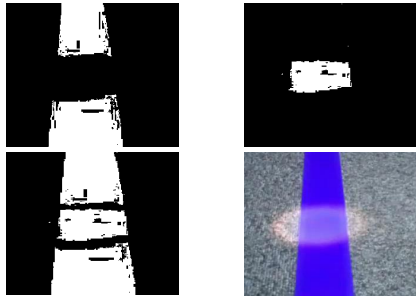


図5 二値画像の合成

表1 白の調整前, 後の再現率

白	調整前	手法1	手法2	手法3
光無し	0.862	0.886	1.00	0.882
弱	0.014	0.943	1.00	0.932
中	0.440	0.833	0.898	0.828
強	0.928	0.933	0.987	0.932

表2 黒の調整前, 後の再現率

黒	調整前	手法1	手法2	手法3
光無し	0.893	0.937	0.893	0.893
弱	0.561	0.775	0.696	0.696
中	0.526	0.671	0.591	0.546
強	0.427	0.477	0.489	0.463

表3 赤の調整前, 後の再現率

赤	調整前	手法1	手法2	手法3
光無し	0.864	0.864	0.935	0.864
弱	0.678	0.841	0.823	0.820
中	0.713	0.853	0.855	0.801
強	0.606	0.872	0.713	0.686

表4 白の調整前, 後の適合率

白	調整前	手法1	手法2	手法3
光無し	0.996	0.996	0.995	0.996
弱	0.723	0.979	0.975	0.981
中	0.797	0.845	0.837	0.846
強	0.832	0.823	0.825	0.830

表5 黒の調整前, 後の適合率

黒	調整前	手法1	手法2	手法3
光無し	0.909	0.913	0.909	0.909
弱	0.880	0.890	0.892	0.892
中	0.814	0.849	0.829	0.822
強	0.858	0.868	0.868	0.867

表6 赤の調整前, 後の適合率

赤	調整前	手法1	手法2	手法3
光無し	0.974	0.974	0.921	0.975
弱	0.925	0.847	0.899	0.930
中	0.928	0.931	0.909	0.922
強	0.955	0.711	0.948	0.947

の比較の2つにより行う。まず調整前と提案手法全体を比較すると、再現率は増加している。この増加は2種類に分けられ、黒と赤は光が当たっている部分の二値化が提案手法ではできているためであり、白は追加でライン上の、光が当たっていない部分の二値化ができているためである。次に提案手法同士の比較を行う。白では、手法2の再現率が高く、適合率も手法1、手法3とほとんど変わらない。手法2の再現率が高くなった理由は、前述のライン上の光が当たっていない部分の二値化ができているためである。黒においては手法1の再現率が高くなっており、適合率に関しても中では手法1が高く、他では差がない。手法1の再現率が高くなった理由は、光が当たっている部分の、正しく二値化された面積が大きいからである。赤では、再現率においてあまり差がなく、適合率は手法3が高い。これは白、黒と比べてHSV変換とHSL変換それぞれの正事例かつ真である領域が共通していたためである。

6 まとめと今後の課題

本研究では、光の照度を変えて実験を行ったが、ほぼ全ての照度において、佐藤の研究で得られた再現率より、高い値が得られた。同様に適合率においても佐藤 [2] の研究で基準としていた0.96に近づけることができた。また、色ごとに適した手法があることもわかった。

今後の課題としては、ラインの一部が暗くなっている場合のライントレースとノイズがある状態でのライントレースがあげられる。

参考文献

- [1] World Robotics 2023 Report: Asia ahead of Europe and the Americas. <https://ifr.org/ifr-press-releases/news/>, (参照 2023-09-26).
- [2] 佐藤孝太. ライントレースを使った屋内飛行運搬ロボットシステムの開発. 2015年度芝浦工業大学大学院理工学研究科機械工学専攻修士論文, 2016.
- [3] Interface編集部(編). 『Interface 2022年4月号 GPSや画像処理で挑戦! ドローン自律飛行』. CQ出版社, 2022.
- [4] 中村恭之, 小林正直, 上田悦子. 『OpenCVによるコンピュータビジョン・機械学習入門』. 講談社, 2017.
- [5] PEKO STEP. 画像から色のRGBを取得するポイントツール. <https://www.peko-step.com/tool/getcolor.html>, (参照 2023-09-25).