

シャッタースピードを考慮した UAVによる反射率の算出について

小形幸平*・全炳徳**

Calculation of Reflectance by UAVs Considering Shutter Speed by

Kohei OGATA*, Byungdug JUN**

This paper discusses a method for calculating the reflectance of a UAV considering the shutter speed, especially for calculating the reflectance accurately from a single image taken by DJI's MAVIC 3M UAV. First, vignetting correction must be performed as a pre-processing step. Next, lens distortion was corrected, and reflectance was calculated based on the irradiance and shutter speed values finally.

Key words : UAV, Remote Sensing, Mavic-3M, Image Processing

1. 序論

近年, UAV(Unmanned Aerial Vehicle)を応用したリモートセンシング技術が農業分野において利用されている. 特に, 代表的な指数として, 以下の正規化植生指標 (Normalized Difference Vegetation Index: NDVI)式(1)が用いられる.

$$NDVI = \frac{NIR_{Ref} - RED_{Ref}}{NIR_{Ref} + RED_{Ref}} \quad (1)$$

ここで, NIR_{Ref} と RED_{Ref} はそれぞれ, 近赤外線バンドと赤バンドによる反射率である. これらからもわかるように, 植生などの把握を行うためには, 計測対象物の反射率を正確に測る必要がある.

一般的な手法として, UAVの画像の反射率補正を行うには, 藤原ら¹⁾が行っているように, 標準反射板(グレー色)を使って, 放射率を補正する手法などが提案されてある. その他, 複数枚のドローン画像を用いて, オルソ画像を生成する手法もある. 村井ら²⁾は, UAV撮影時におけるBRF(bidirectional reflectance distribution factor)を考慮したオルソ画像を構築する手法を発表している.

本論ではこれらの手法とは異なるもので, カメラの機能として捕らえたセンサー情報を直接用いて植生指数を算出するものである. 本研究の目的は, マルチスペクトルカメラを搭載したUAVにおいて, 地

上の例えば茶園などを撮影した際に, マルチバンドデータのシャッタースピードが固定できない技術的欠点を補うためのものである. この手法により標準反射板やオルソ画像の構築において, 複雑な作業が省力化できる.

実際, 近年のUAVの撮影技術の向上により, 様々なセンサーが取り付けられている. それらを用いることで, 付帯設備の標準反射板や複数枚の画像などを使わずに, 一枚の画像で正確な反射率を測ることができるのではないかと考えた.

本論では, 農業の航空測量のために開発されたUAVである「MAVIC 3M」(DJI社)で撮影された画像から反射率を算出する方法を論じる. 本論においては, DJI社が提供している「Mavic 3M Image Processing Guide」³⁾を参考にして作成している.

2. 材料と方法

本研究において, 使用したUAVは「MAVIC 3M」(DJI社)である. 「MAVIC 3M」は, RGBの可視光のカメラと4つのマルチスペクトルカメラを搭載している. 4つのマルチスペクトルカメラが対応している波長はグリーン(560 ± 16 nm), レッド(650 ± 16 nm), レッドエッジ(730 ± 16 nm), そして近赤外(860 ± 26 nm)の4種類である. また, 本研究における解析では, 主にコンピュータビジョン向けライブラリであるOpenCVを使用している.

令和5年12月18日受理

* 工学研究科博士後期課程 (Graduated School of Engineering)

** 総合生産科学域 (Institute of Integrated Science and Technology)



図 1 MAVIC 3M のレンズ配置

本手法の大まかな流れについて説明する。まず、「MAVIC 3M」に搭載されているセンサーの情報を Exif 情報から読み取る。Exif 情報を参照して、レンズ周辺減光(optical vignetting)の処理を行い、次に、レンズ固有の歪みを補正する。その後、シャッタースピード(exposure time)を考慮した反射率を算出する。

ここで、「MAVIC 3M」の搭載しているセンサーから得られる Exif 情報について、本論で特に用いた値について、以下に簡単に紹介しておく。

表 1 本論で扱う主な Exif 情報の値

【XMP】drone-dji		
名称	型	意味
Calibrated Optical CenterX	float	画像上における中心の X 座標
Calibrated Optical CenterY	float	画像上における中心の Y 座標
Vignetting Data	string	レンズ周辺減光の補正係数
DewarpData	string	レンズに固有の補正パラメーター
Irradiance	float	放射照度
Sensor Gain	float	マルチスペクトルカメラに固有の補正係数
Sensor Gain Adjustment	float	標準 NIR モジュールに対するマルチスペクトル画像センサーの補正係数
ExposureTime	Integer	シャッタースピード
Black Current	integer	ブラックレベル

2.1 レンズ周辺減光(optical vignetting)の補正

レンズ周辺減光とは、画像の中心から周辺にかけての輝度値が減少するビネッティング(Vignetting)と呼ばれる現象である。これを補正するには、「MAVIC 3M」における[Vignetting Data]の補正係数を下記の式(2)に用いることで算出できる。

$$I_X(x,y) = I(x,y) \times (k[5] * r^6 + k[4] * r^5 + k[3] * r^4 + k[2] * r^3 + k[1] * r^2 + k[0] * r + 1.0) \quad (2)$$

ここで、 $I_X(x,y)$ とは、 X の波長におけるレンズ周辺減光補正を施した座標 (x,y) における輝度値を表し、 $I(x,y)$ とは、任意のピクセル座標 (x,y) における輝度値を表している。また、 r とは、画像の中心から任意のピクセル (x,y) までの距離を表しており、下記の式(2)で表される。また、 $k[0] \sim k[5]$ は、Exif 情報にある[VignettingData]内から取得できる。

$$r = \sqrt{(x - \text{CenterX})^2 + (y - \text{CenterY})^2} \quad (3)$$

2.2 レンズ固有の歪みの補正

次に、レンズ固有の歪みを補正する。これは、レンズに固有の歪みが存在する場合に、歪みを補正するものである。こちらも Exif 情報内の[DewarpData]に値があるため、そちらを参照する。焦点距離 (fx, fy) と、光学距離 (cx, cy) を基に、カメラマトリックスを構築する。一般にカメラマトリックスは、下記の行列式(4)で表される。

$$\text{camera matrix} = \begin{bmatrix} fx & 0 & \text{CenterX} + cx \\ 0 & fy & \text{CenterY} + cy \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

また、Exif 情報内の[DewarpData]には、レンズ歪み補正に伴う $[k1, k2, p1, p2, k3]$ のパラメーターも記載してある。特に、 $k1, k2, k3$ は、ラディアル歪み係数といわれ、 $p1, p2$ は、切断歪み係数といわれている。これらのパラメーターを OpenCV の undistort 関数で用いることで、レンズ固有の歪みを補正する。

2.3 シャッタースピードを考慮した反射率の算出

次に、シャッタースピードを考慮する反射率の算出を行う。下記の式(5)で表される。

$$X_{Ref} = \frac{(I_X - I_{BlackLevel}) / (X_{gain} * \frac{X_{etime}}{10^6}) * \rho_{CamX}}{[Irradiance] * 2^{bitnum}} \quad (5)$$

なお、式(5)のそれぞれの変数は下記の通りである。

- X_{Ref} : X波長における反射率
- I_X : X波長における輝度値
- $I_{BlackLevel}$: X波長におけるブラックレベル
- ρ_{CamX} : マルチスペクトルカメラ固有の補正係数
- X_{gain} : 標準 NIR モジュールに対するマルチスペクトル画像センサーの補正係数
- X_{etime} : シャッタースピード
- [Irradiance] : 放射照度
- bitnum: 画像のビット数

3. 結果

本手法を用いて、宮崎県日南市酒谷地方にある「山口製茶」の圃場において、2023年4月8日に撮影した画像について適用した。まず、図2はレンズ周辺減光補正を行う以前のUAVによる撮影画像である。わかりやすいように輝度値をカラーバーで表示している。レンズ周辺減光補正を行った図3は中心から外側にかけての輝度値が上がっているのが確認できる。

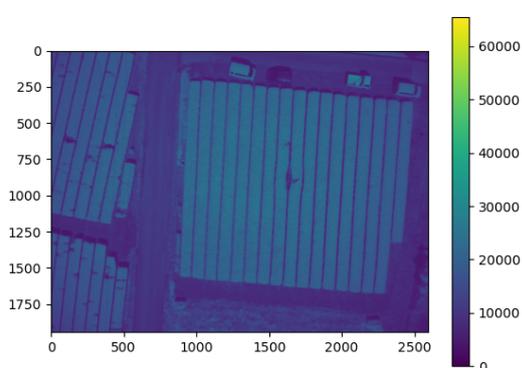


図2 レンズ周辺減光補正前の NIR の輝度値

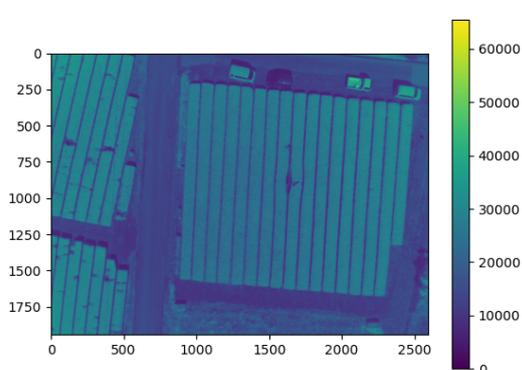


図3 レンズ周辺減光補正後の NIR の輝度値

次に、レンズ歪み補正を行ったが、レンズ周辺減光補正と変わらなかったため、画像は割愛する。最後に、シャッタースピードを考慮した反射率を計算した値を図4で確認することができる。最後に、同時刻に撮影されたRedバンドにも同様の処理を施し、NDVIを求めて図5に示す。

4. 結論

本論では、主に DJI 社が提供している「Mavic 3M Image Processing Guide」³⁾を参考にして、画像処理を進めてきた。本論で紹介した前処理の他にも、カメラの光学精度などの違いの補正をする手法や画像の位置合わせの手法が紹介されていたが、1枚の画像における反射率の計算であったため、本論では割愛した。今後の研究課題として、これらの手法を用いた反射率の処理結果と、正確な反射率を計測するマルチスペクトルフォトメーターなどの値との比較を行うことにより、処理をした画像の結果を検証する必要がある。今後の課題にしておきたい。

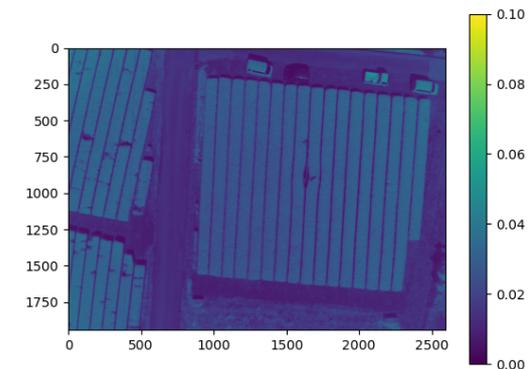


図4 シャッタースピードを補正した NIR の輝度値

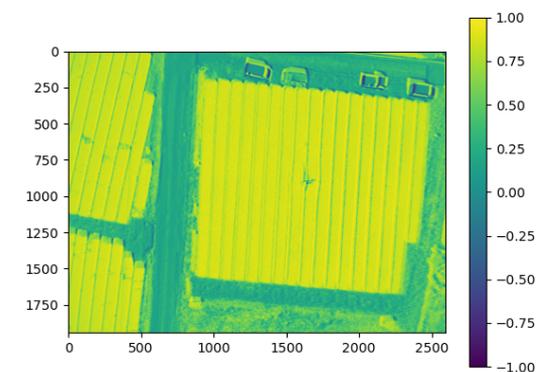


図5 本手法を適用した茶園における NDVI 結果

付録

ここでは、付録として補正を行った際のコードを記載しておく。但し、言語はPythonを用いている。

【Exif情報の取得】

```
from PIL import Image, ImageTk
from PIL.ExifTags import TAGS, GPSTAGS
image = Image.open(image_path)

def get_exif(x):
    exif_dict = x.getexif()
    exif = [TAGS.get(k, "Unknown")+ f":
{str(v)}" for k, v in exif_dict.items()]
    exif_str = "\n".join(exif)
    return exif_str
print(get_exif(image))
```

【レンズ周辺減光補正処理】

```
k = [-0.000016779, 1.386650e-06, -
4.019088e-09, 6.862371e-12, -5.235157e-
15, 1.481126e-18]
centerX = 1296.000000
centerY = 972.000000
height, width = dn_values.shape

distances = np.zeros((height, width))
for y in range(height):
    for x in range(width):
        distances [y, x] = np.sqrt((x -
centerX)**2 + (y - centerY)**2)
height_r, width_r = dn_values_r.shape

VignettingCollction_array =
dn_values*(k[5] *distances**6 + k[4]
*distances**5 + k[3] *distances**4 +
k[2] *distances**3 + k[1] *distances**2
+ k[0] *distances +1.0)
```

【レンズの固有の歪みの補正】

```
dewarp_data = "2022-08-
02;2170.000000000000,2170.000000000000,0
.000000000000,0.000000000000,0.000000000
000,0.000000000000,0.000000000000,0.0000
0000000,0.000000000000"
date_str, params_str =
dewarp_data.split(";")
```

```
params = [float(param) for param in
params_str.split(",")]
fx, fy, cx, cy = params[0], params[1],
params[2], params[3]
k1, k2, p1, p2, k3 = params[4],
params[5], params[6], params[7],
params[8]

camera_matrix = np.array([[fx, 0,
centerX + cx], [0, fy, centerY + cy],
[0, 0, 1]])

dist_coeffs = np.array([k1, k2, p1, p2,
k3])

undistorted_image =
cv2.undistort(VignettingCollction_array_
nir, camera_matrix, dist_coeffs)
```

【シャッタースピードを考慮した反射率の算出】

```
I_BlackLevel = 3200
nir_gain = 1.021
r_gain = 1.000
nir_etime = 1094
r_etime = 1514
nir_irradiance = 11467.438477
r_irradiance = 14887.082031
pCamNIR = 1.036728
pCamR = 0.535415

X_Ref = (I_X -
I_BlackLevel)*10**6*pCamX/(X_gain*X_etim
e*X_irradiance**2**16)
```

参考文献

- 1) 藤原洋一, 塚口直史, 長野峻介 : 普及型マルチスペクトルカメラによる水稻生育診断, 農業農村工学会, 87巻2号, p. IV_7, 2019.
- 2) 村井亮介, 高木方隆 : UAVを用いたBRFを考慮したオルソ画像生成手法, 日本写真測量学会, 62 (1), pp.22-29, 2023.
- 3) DJI : Mavic_3M_Image_Processing_Guide_EN, DJI, pp.1-10, 2023.
https://dl.djicdn.com/downloads/DJI_Mavic_3_Enterprise/20230829/Mavic_3M_Image_Processing_Guide_EN.pdf (参照 2023-12-17)