DOI: 10.12700/AIS.2025.012

Analysis and correction of raw WREN satellite images

Gábor Nagy*
*Óbuda University, Alba Regia Faculty, Institute of Geoinformatics,
nagy.gabor@amk.uni-obuda.hu
ORCID ID: 0000-0001-9453-2291

Abstract—This paper describes some research related the sensor calibration of an optical remote sensing satellite. The WREN satellite acquires multispectral images, but these raw images need some correction before any application. The columns of the image, which was created by different sensors of the camera, does not contain comparable data, the images have a striped view.

The described studies analyze the histograms of the different columns of the images and unified the the distribution for creating a corrected image or calibrate the sensor.

Index Terms—WREN, satellite image, remote sensing

I. Introduction

The WREN-1 satellite has launched in 16 August 2024 by the Falcon 9 launcher of the Transporter 11 mission[1]. This six unit size cubesat was made for the WREN (Water Resources in Efficient Networks) project to take multispectral images [2], [3], [4] for a drought monitoring system.

II. The images

The WREN-1 satellite can take multispectral images from the Earth's surface in six bands. The image of a band contains 2048 columns and (in the examined cases) 5120 rows of 16 bit values.

The images are acquired in these bands:

- 01 Band, 445-515 nm (Blue)
- 02 Band, 505-585 nm (Green)
- 03 Band, 625-695 nm (Red)
- 04 Band, 770-790 nm (Near infrared)
- 05 Band, 890-910 nm (Near infrared)
- 06 Band, 445-805 nm (Panchromatic, not use in the described research)

When we look an image, we can see a striped pattern, because the sensors of the different columns are not uniform. (Figure 1.)

We need a method to unify the columns of the image.

There are more method for correcting the camera geometry[6], [7], but in this case we need a method for the radiometry correction of the sensors.

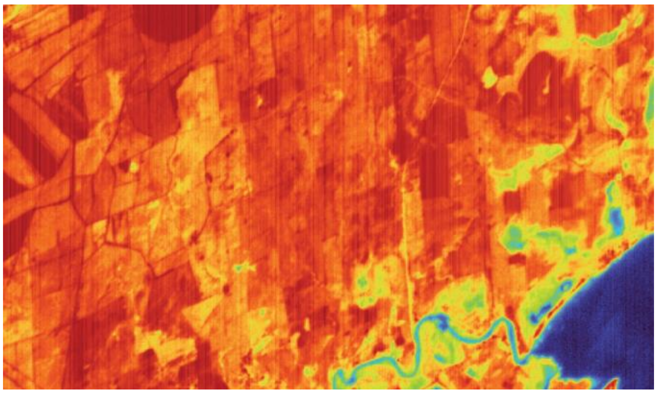

Figure 1. The view of a part of the original image in the 04 band which was taken 19 February 2025. The picture was created by QGIS [5], using a color scale, because the striping is more visible with this method.

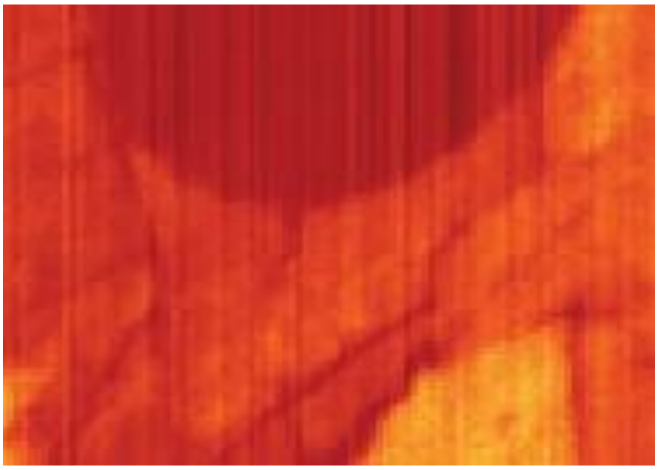

Figure 2. The enlarged view of the Figure 1.

III. A suggested correction method

The histograms can be calculated of the columns to study the differences. (Assume that columns close to each other are similar.) The following figures show the percentiles (the boundaries between the percentiles) with different colors in the columns of the image.

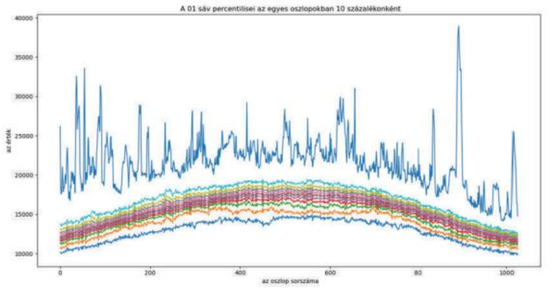

Figure 3. Distribution by percentiles in the 01 band (Blue, 445-515 nm).

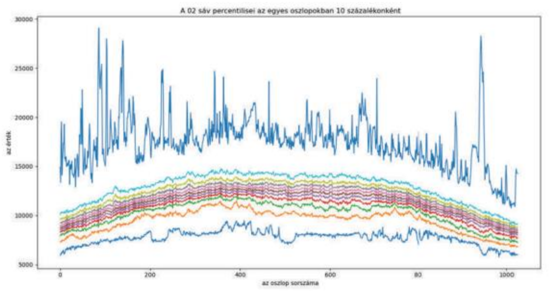

Figure 4. Distribution by percentiles in the 02 band (Green, 505-585 nm).

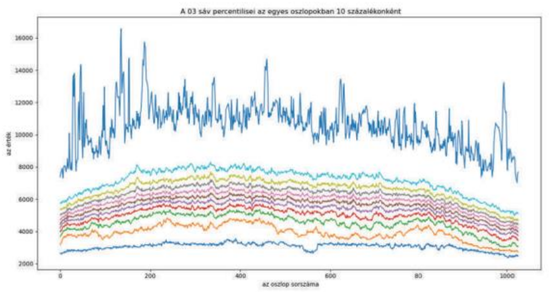

Figure 5. Distribution by percentiles in the 03 band (Red, 625-695 nm).

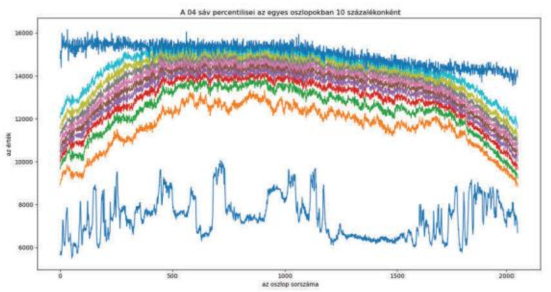

Figure 6. Distribution by percentiles in the 04 band (NIR 770-790 nm).

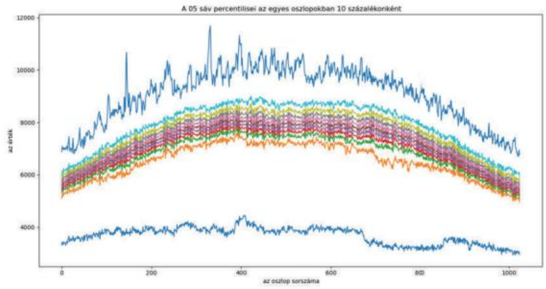

Figure 7. Distribution by percentiles in the 05 band (NIR 890-910).

We can zoom in on a part of the diagram to clearly see the differences between the columns of the image.

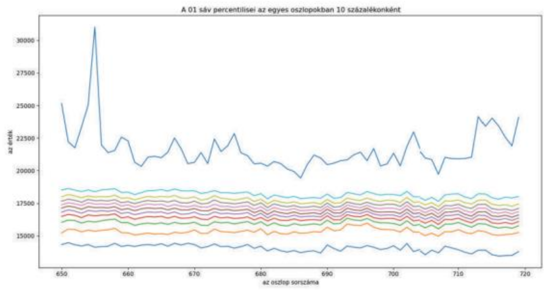

Figure 8. Distribution by percentiles in the 01 band. (zoomed part)

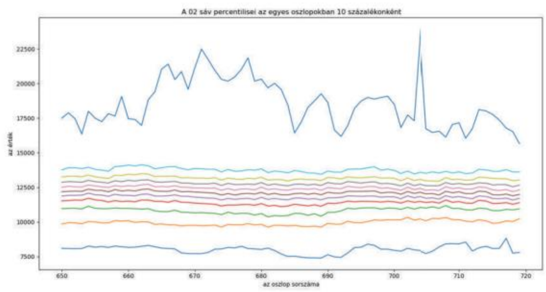

Figure 9. Distribution by percentiles in the 02 band. (zoomed part)

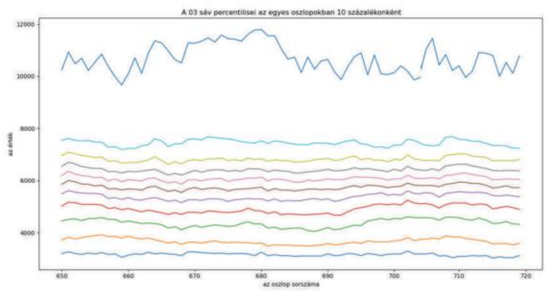

Figure 10. Distribution by percentiles in the 03 band. (zoomed part)

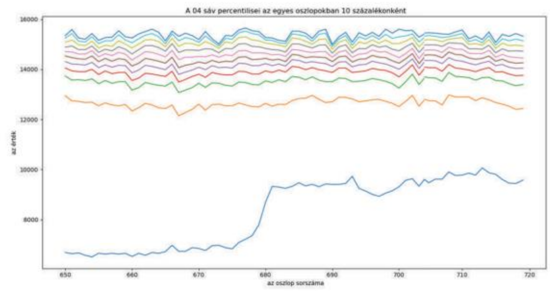

Figure 11. Distribution by percentiles in the 04 band. (zoomed part)

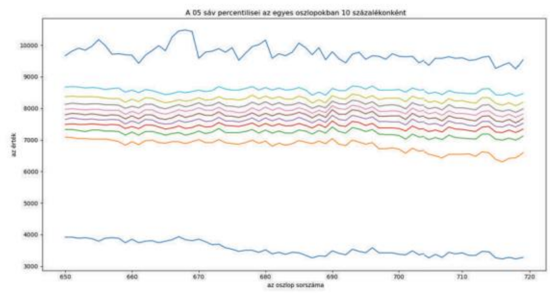

Figure 12. Distribution by percentiles in the 05 band. (zoomed part)

In the Figure 13. there are smoothing lines in dotted line. The smoothing lines created by linear regression from the neighbors (±7) of the columns. The program fit a regression line to these 15 values, and estimate a smoothed value with the regression line in the inspected point. The program use Sector Based Linear Regression [8] which is a robust regression, the generalized process of the Fitting Disc Method [9] which developed for LiDAR data processing.

The robust regressions (like the used SBLR) can provide an appropriate result in that case too, if the dataset contains some outlier. The SBLR was used in the described analyses for this reason.

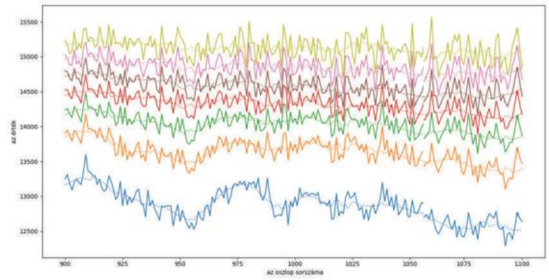

Figure 13. Distribution by percentiles in the 04 band (zoomed part) with smoothing lines (dotted lines).

The figures was created by a Python [10] program with NumPy [11] and Matplotlib [12], [13]. The source code of the program in the APPENDIX.

The program calculated a linear regression between the original and the smoothed percentiles (use also SBLR, between the 9 key values of the distributions), and apply this linear function to calculate the corrected values of the pixels of the column. This method can provide a corrected image.

IV. Conclusion

The result of the studied method is in the Figure 14.

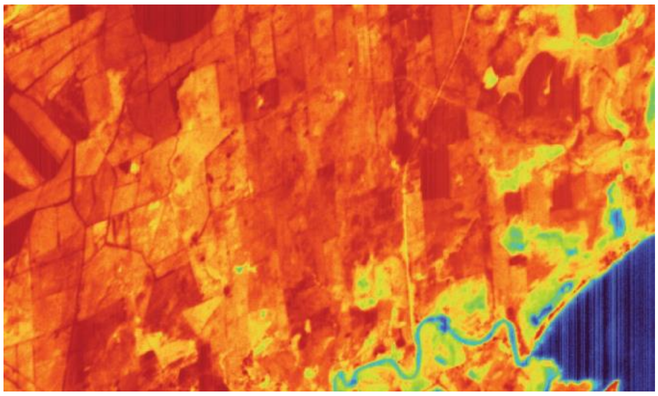

Figure 14. The corrected image (the same part and same color scale than the Figure 1.)

The striped effect are reduced in the corrected image, but has not eliminated fully, mainly the most highest values (red in the Figure 14.). In the most lowest values of the image (typically water surfaces, blue in the Figure 14.) the result are worse than in the original image.

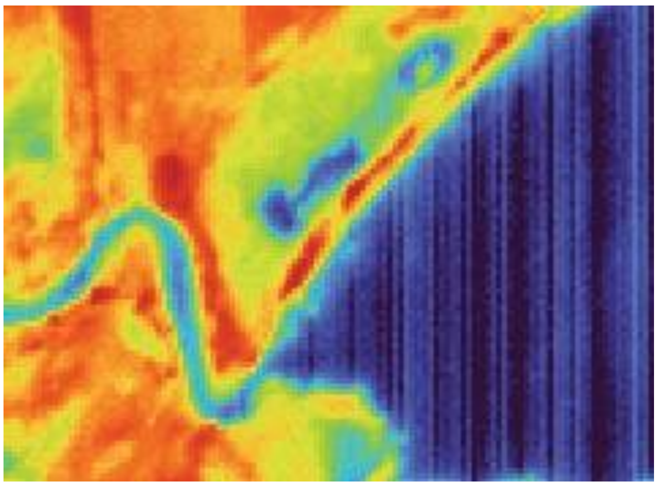

Figure 15. The enlarged view of the Figure 14.

The linear regression is not suitable for the correction in the highest and lowest parts of the histogram. In the future I can use other regression (for example polynomials) or a different linear regressions in the highest and lowest values.

An usable solution for the described problem need more research.

The usable calibration of the sensor also needs more research in the future.

References

[1] S. Gyányi, R. Szabolcsi, P. J. Varga, G. Horváth, P. Horváth, and T. Wührl, "Micro-vibrations analysis in leo cubesats using mems accelerometers," *Sensors*, vol. 25, no. 18, p. 5917, 2025.

[2] M. V. Wojtaszek *et al.*, "Comparison of three image classification methods in urban environment," *Journal of Geoscience and Environment Protection*, vol. 3, no. 02, p. 54, 2015.

[3] X. Zhou, T. Jancso, C. Chen, M. Verone *et al.*, "Urban land cover mapping based on object oriented classification using worldview 2 satellite remote sensing images," in *International scientific conference on sustainable development & ecological footprint*, vol. 2, 2012.

[4] Q. Meng, C. Wang, X. Gu, Y. Sun, Y. Zhang, R. Vatseva, and T. Jancso, "Hot dark spot index method based on multi-angular remote sensing for leaf area index retrieval," *Environmental Earth Sciences*, vol. 75, pp. 1–11, 2016.

[5] QGIS Development Team, *QGIS Geographic Information System*, QGIS Association, 2023. [Online]. Available: https://www.qgis.org

[6] Z. Tóth, G. Mélykúti, and Á. Barsi, "Digitális videokamera kalibrációja," *GEOMATIKAI KÖZLEMÉNYEK / PUBLICATIONS IN GEOMATICS*, vol. 8, pp. 297–302, 2005. [Online]. Available: https://m2.mtmt.hu/api/publication/2201237

[7] Z. Kugler, Z. Toth, Z. Szalay, D. Szagri, and A. Barsi, "Supporting microclimate modelling with 3d uas data acquisition," *IDOJARAS / QUARTERLY JOURNAL OF THE HUNGARIAN METEOROLOGICAL SERVICE*, vol. 123, pp. 279–294, 2019. [Online]. Available: https://m2.mtmt.hu/api/publication/30672653

[8] G. Nagy, "Sector based linear regression, a new robust method for the multiple linear regression." *Acta Cybernetica*, vol. 23, no. 4, 2018.

[9] G. Nagy, T. Jancsó, and C. Chen, "The fitting disc method, a new robust algorithm of the point cloud processing," *ACTA POLYTECHNICA HUNGARICA*, vol. 14, no. 6, pp. 59–73, 2017.

[10] G. Van Rossum *et al.*, "Python Programming Language." in *USENIX Annual Technical Conference*, vol. 41, 2007.

[11] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[12] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, May-Jun 2007.

[13] G. Nagy, "Matplotlib in the geospatial analyzes," in *12th International Symposium on Applied Informatics and Related Areas (AIS 2017)*, G. T. Orosz, Ed. Óbudai Egyetem, 2017, pp. 90–93.

APPENDIX

```
#!/usr/bin/python3
"""
Értékek eloszlása oszloponként és percentilisenként

     \      /    I----    I----    I\  I
      \    /     I----    I----    I \ I
       \/\/      I  \     I----    I  \I

A programot Nagy Gábor készítette a WREN keretében
"""

import numpy as np
from osgeo import gdal
from osgeo.gdalconst import *
import matplotlib.pyplot as plt
from sblr import sblr

def beolvas(felvetel, savok):
    """A felvétel beolvasása sávonként egy szótárba"""
    ret={}
    for sav in savok:
        kepfile=gdal.Open(felvetel+'/HICAM/band_'+sav+'.tiff',    GA_ReadOnly)
        ret[sav]=kepfile.ReadAsArray()
    return ret

def ticklist(rnum, cmin, cmax, ticknum=7, unit=1.0):
    """Tickerek és címkéik elöállítása a Matplotlib számára"""
    rlog=np.log10((cmax-cmin)/(ticknum-1))
    if rlog%1<=0.1505149978319906: #np.log10((1*2)**0.5)
        rsz=1
    elif rlog%1<=0.34948500216800943: #np.log10((2*2.5)**0.5)
        rsz=2
    elif rlog%1<=0.54845500650402823: #np.log10((2.5*5)**0.5)
        rsz=2.5
    elif rlog%1<=0.84948500216800937: #np.log10((5*10)**0.5)
        rsz=5
    else:
```

```
        rsz=10
    tickdist=int(rsz*10**int(rlog))
    tickcoords=np.arange(tickdist*np.ceil(cmin/tickdist), cmax+0.0001, tickdist)
    if tickdist%1==0.0:
        ticklabels=[str(int(x*unit)) for x in tickcoords]
    else:
        ticklabels=[str(x*unit) for x in tickcoords]
    ticks=[rnum*((x-cmin)/(cmax-cmin)) for x in tickcoords]
    return (ticks, ticklabels)

szinek=['tab:gray', 'tab:blue', 'tab:orange', 'tab:green', 'tab:red', 'tab:brown', 'tab:pink',

adat=beolvas('2025_02_19T13_10_u1', ['01', '02', '03', '04', '05'])

sav='04'
kezd_osz=900
veg_osz=1100
ih=7
pctn=8
pctq=[i*(100/pctn) for i in range(pctn+1)]

nsor=adat[sav].shape[0]
nosz=adat[sav].shape[1]
pct=np.zeros((pctn+1,nosz))
pctsim=np.zeros((pctn+1,nosz))
for osz in range(nosz):
    for i in range(pctn+1):
        pct[i][osz]=np.percentile(adat[sav][:,osz], pctq[i])
for osz in range(ih, nosz-ih):
    for i in range(pctn+1):
        regegym, regegyb = sblr(list(zip(np.arange(osz-ih, osz+ih+1), pct[i,osz-ih:osz+ih+1]))
        pctsim[i, osz] = regegym*osz+regegyb
        if osz==ih:
            for osze in range(ih):
                pctsim[i, osze] = regegym*osze+regegyb
        if osz==nosz-ih-1:
            for oszu in range(osz+1,nosz):
                pctsim[i, oszu] = regegym*oszu+regegyb

```

```
kezd_osz=900
veg_osz=1100

for i in range(1,pctn):
    plt.plot(pct[i][kezd_osz:veg_osz+1], color=szinek[i], linestyle='-')
    plt.plot(pctsim[i][kezd_osz:veg_osz+1], color=szinek[i], linestyle=':')
plt.title('A '+sav+' sáv percentilisei az egyes oszlopokban 10 százalékonként')
plt.xlabel('az oszlop sorszáma')
plt.ylabel('az érték')
tckx,tckl=ticklist(veg_osz-kezd_osz, kezd_osz, veg_osz)
plt.xticks(tckx, tckl)
plt.show()
```