Hyper-spectral to DEM automatic alignment and projection algorithm for Change Analysis of Mountainous Terrain

Stefano Dell'Anna

May 17, 2023

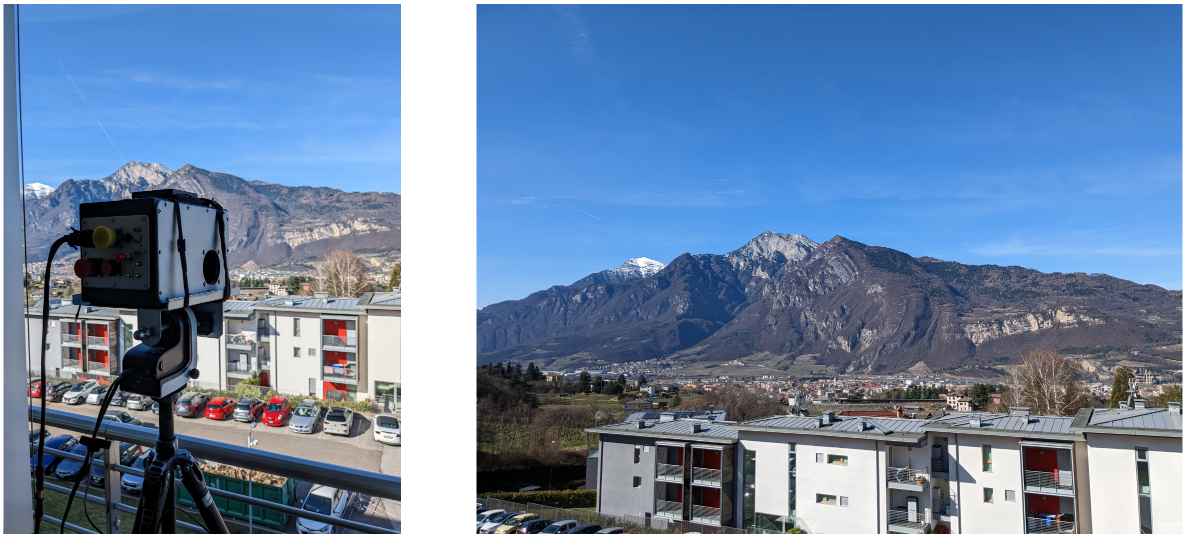
1 Introduction

When discussing applications for change detection, satellite-acquired data represents one of the most valuable assets. They provide consistent data over time in terms of spatial and spectral resolution, as well as the acquisition geometry, which makes it easier to process and detect changes.
However it is also important to consider the acquisition frequency of said platforms, taking PRISMA for example, the system has a re-look capability of 7 days, the frequency of acquisition can thus result too low in the investigation of fast changing phenomena, when meteorological factors like cloud cover are taken into account, this figure can further decrease to several weeks or even months for a particular region.
For the aforementioned reasons, these systems are often combined with higher-resolution and more flexible acquisition systems, an example could be airborne sensors to acquire data on-demand, however these systems come with logistical challenges and high operational costs and are more suited for medium scale operations.
When acquiring a limited area, for example a mountain or a valley, one solution can be to acquire data from a ground based sensor and project this information onto the corresponding Digital Elevation Model (*DEM*) at a comparable resolution to satellite-acquired or airborne-acquired data for the specific application at hand.
One of the most problematic aspects of this solution is the condition number of the *DEM* to image alignment with respect to the position and orientation of the camera in the 3D space, for this reason a robust system to align the image with the *DEM* is needed.
To perform this task *GNSS* positioning could be used, this solution, while effective, is expensive, with the need for an high precision *GNSS* receiver and alignment suite, furthermore could still present some problems when the condition are not ideal, for example in low coverage and especially in high-latitude areas where the precision drastically decrease, the accelerometers and magnetometers to estimate the camera pose are also susceptible to drift and need to be accurately calibrated for every acquisition.
For this reason in this project an image-to-*DEM* alignment tool is developed[1], aiming at retrieving the position in the *DEM* Coordinate Reference System *CRS* and camera pose (azimuth, elevation and roll) that best align the content of the acquired hyper-spectral image for the projection.

2 Acquisition system

In order to understand the processing chain from the hyper-spectral image acquisition to the data projection onto the *DEM*, an overview of the reference systems used is needed.
In the processing chain a reference unitary sphere, called Sphere Reference Frame (*SRF*), is used to represent the acquisition system and allow the algorithm to correctly map the hyper-spectral image samples in the image coordinate system (*ICS*) to the corresponding direction of arrival (*DoA*) in order to perform the inverse operation during the re-projection onto the *DEM*.

[1] https://github.com/Stedll/HyDAAP



(a) Acquisition system. (b) View of the acquired scene.

Figure 1: Acquisition scenario.

2.1 Star tracker

To properly acquire the data the acquisition system is mounted onto a Sky-Watcher AZ-GTI mount, this is used in combination with a custom serial interface in order to control and acquire the angular position data to correctly register the line-wise acquisition of the hyper-spectral camera.
The star-tracker possess two different coordinate systems:

- **internal reference system** (*IRS*): The azimuth and elevation angle of the plate to which the hyper-spectral camera is mounted with respect to it's base, the head doesn't have a roll axis.
- **external reference system** (*ERS*): The position of the plate to which the hyper-spectral camera is mounted in the world reference system given by the cardinal position in the *DEM* coordinate system and the $zy'x''$ Tait-Bryan angles (yaw, pitch and roll) that describe the camera pose.

The distinction between these two different systems is essential when taking into account a *IRS* non zero-elevation acquisition.
In these scenarios the trajectory of the hyper-spectral sensor center does not lay on the geodesic of the *SRF* and the two systems are then not commutable.
The the algorithm need to map the samples position to the *SRF* using the data of the position of the head in *IRS*, the *SRF* will be then be roto-translated, according to the data of the optimization and correlation processes to the estimated *ERS* coordinates.

2.2 Hyper-spectral camera

For the acquisitions a HySpex Mjolnir v1240 hyper-spectral, shown in fig. 1a camera is used, acquiring data in 200 3-nm bands with a vertical resolution of 1240 pixel in a pushbrooom acquisition configuration.
The camera is mounted onto the tracker plate with the scanner vertical with respect to the tracker base in zero-elevation and zero-azimuth direction according to the *IRS*, such that during the acquisition with a pure azimuthal rotation the system can be represented as a cylindrical sensor.
In both zero or non-zero elevation cases, the acquisition is projected onto the *SRF* according to the sensor and lens specification to obtain the correspondences between the samples and their *DoA*.

3 Skyline extraction

For this particular project the choice was to extract the main skyline, corresponding to the edge between ground and sky in the image, for a better match and in order to align better the image to the *DEM* multiple ridges can be extracted to increase the reliability and robustness of the technique.

3.1 DEM skyline

To extract the skyline from the *DEM* this is re-sampled in a cylindrical fashion to avoid the use of computational heavy viewshed's algorithms.
After the re-sampling the points are mapped to spherical coordinates, using then a simple sorting in each angular bin with respect to the elevation angles associated to each sample the algorithm extracts the one with the highest viewing angle.
This allow the code to extract the sample correspond to the maximum height of the terrain visible in a particular direction, repeating this process on each bin the whole skyline is extracted.
The re-sampling process is adaptive to the specified angular resolution for the skyline, this step is needed in order to avoid, if converting directly from Cartesian to spherical coordinates, to extract samples that don't belong to the real skyline but exist behind the ridge and are not visible from the specified camera position, due to the fact that some angular bins can include only samples that do not belong to the real skyline.
The extracted points are plotted in a 2:1 image where every pixel correspond to a spherical direction in $[-\pi, \pi]$ for azimuth and $[-\pi/2, \pi/2]$ for elevation, sampled at the same angular resolution.

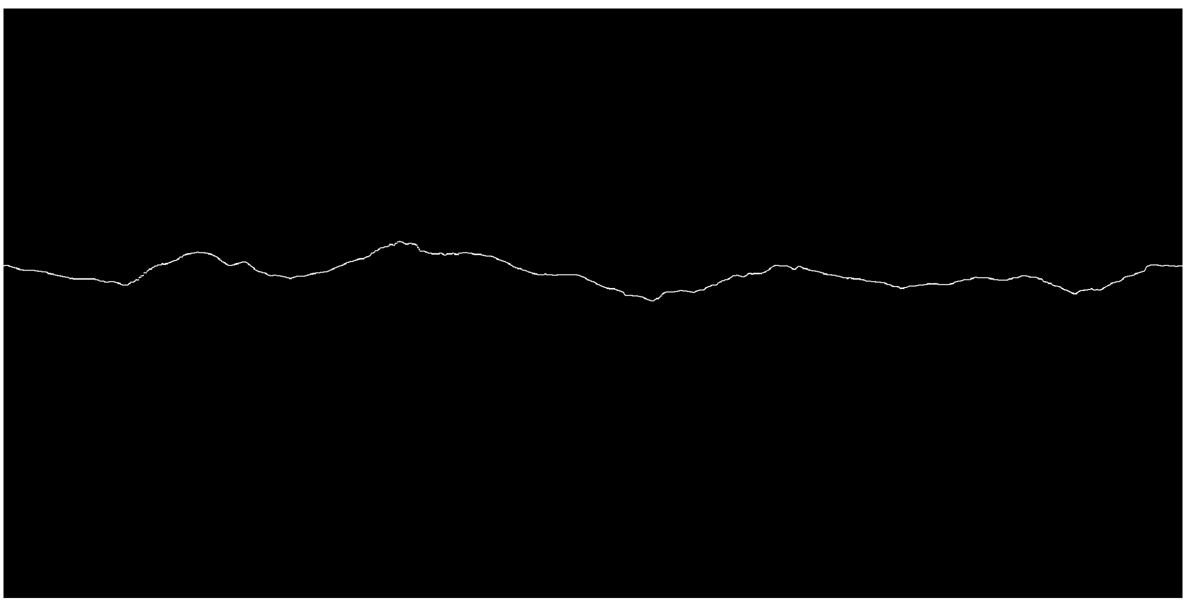


Figure 2: Extracted *DEM* skyline.

3.2 Image skyline

The extraction process of the skyline from the image can vary depending on the data, in general, when talking about hyper-spectral acquisitions, different bands can be used in order to differentiate the sky from the ground, computing the mask of the last and extracting its upper edge the algorithm can easily plot the skyline.
In this particular implementation two near-infrared bands and a visible blue band are used, in combination with a series of filtering steps, to obtain a gray-scale image with good contrast between the sky and the ground, a binary threshold is then applied, followed by an edge-extraction algorithm and a filtering process to extract the main skyline from the image.
When considered multi-spectral or RGB images the process is similar, depending on the available bands a more sophisticated technique before the masking process might be needed in order to better discriminate between ground and sky.
After the extraction of the skyline edge, this need to be re-projected in *SRF*, using the camera specification it is possible to compute the *DoA* of every pixel in the original image.
In the case of the hyper-spectral acquisitions described in 2 a simple look-up process for each skyline pixel is enough, having already mapped each pixel to the corresponding *DoA*.
As for the skyline extracted from the *DEM* described in 3.1 the skyline points are plotted in a 2:1 image with the same specifications, this allow the two skylines to be compared later.

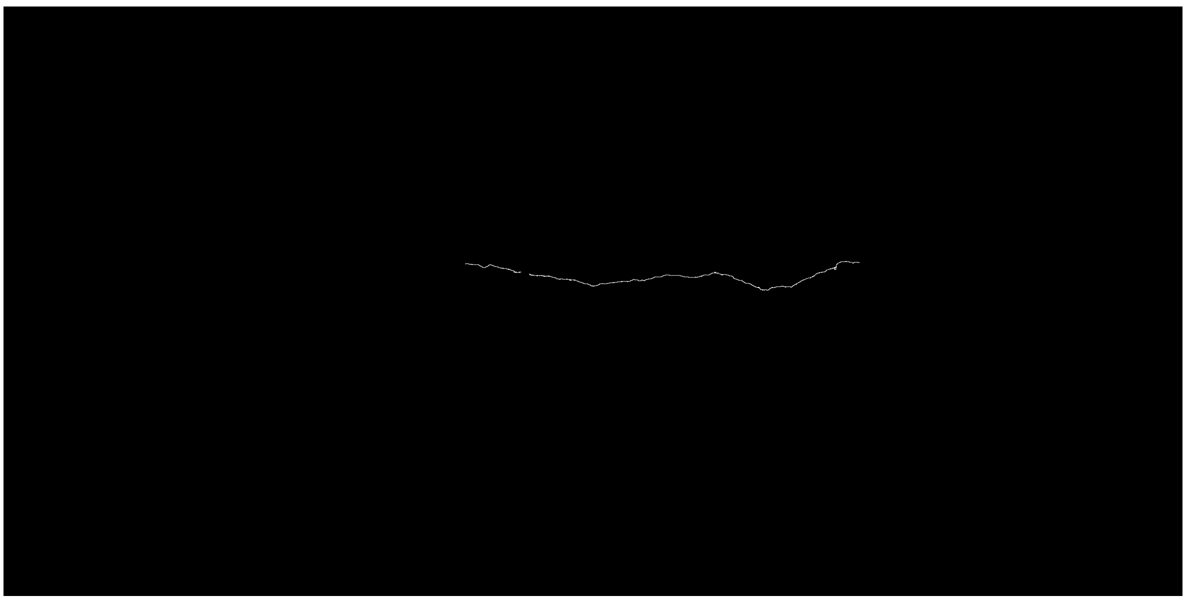


Figure 3: Extracted hyper-spectral skyline.

4 Camera pose estimation

The most important parameter to estimate is the camera pose, the three angles that represent the orientation in space of the *SRF* with respect to the *ERS*, these parameters are extremely delicate as an error of just one degree can lead to completely incorrect projections.

4.1 Spherical cross-correlation

Spherical cross correlation is a mathematical operation that measures the similarity between two functions defined on a sphere.
Given two functions f and g defined on the surface of a sphere, the spherical cross correlation measures how much the values of f and g overlap as the sphere is rotated.
Spherical cross-correlation can be seen as an extension of 2D cross-correlation to functions defined on spheres, and a third degree of freedom that allow to describe all possible rotations between the two functions.

4.1.1 Coefficients

The spherical coefficients that describe the f and g functions are extracted from the two spherical skyline images using PyShTools, a python wrapper for a collection of FORTRAN routines to work with spherical harmonics, the coefficients are then collected in a file to be used in the correlation process.

4.1.2 Correlation

SOFT [2][4] is a collection of C routines, based on FFTW3 in the used implementation, to perform coefficient extraction and correlation of spherical functions.
For the developed project, having already calculated the spherical coefficients of the functions, a custom function was developed to calculate the cross-correlation and extract the ZYZ Euler angles that can be converted to the wanted $zy'x''$ Tait-Bryan angles representing yaw, pitch and roll of the camera with respect to the *ERS*.

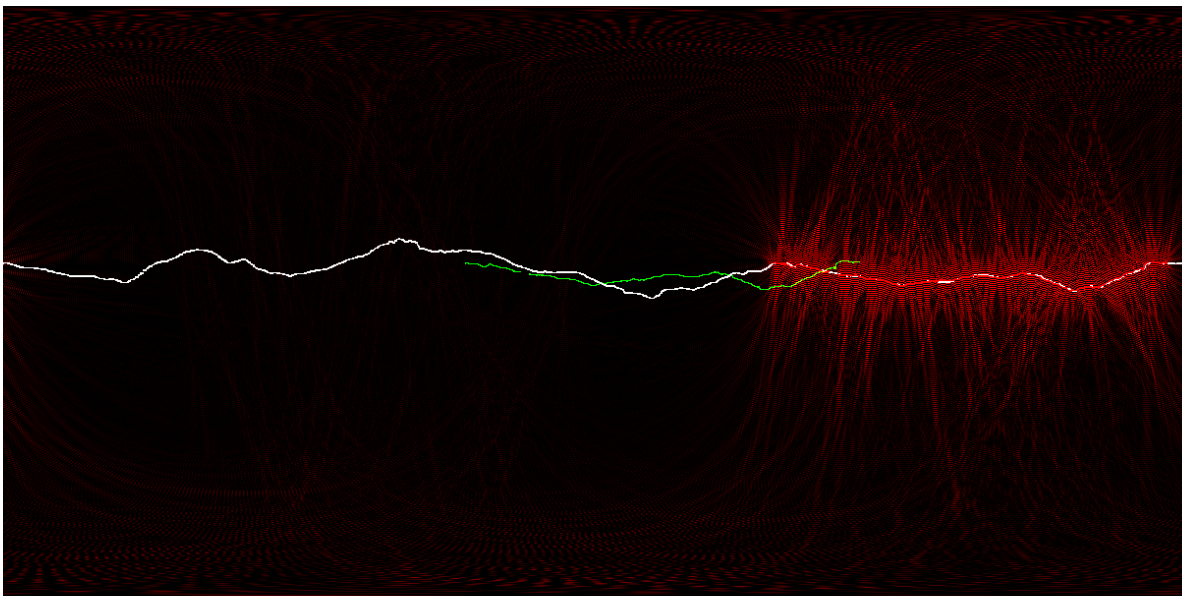


Figure 4: alignment of hyper-spectral skyline on *DEM* skyline before (green) and after (red) the matching process.

4.2 DEM position optimization

The data needed in the whole processing chain are the *DEM* of the acquired area, the hyper-spectral image and the corresponding tracker information to correct it, and a rough estimation of the acquisition location, like a *GNSS* estimation or the knowledge of where the acquisition was made.
Due to the nature of it, the movement on the *DEM* surface, corresponding to the real life position of the system during the acquisition, can hugely impact the alignment [1] of different topographic elements that coexist in the acquired view if they lay at different distances from the sensors due to parallax effect.
To optimize this parameter a gradient ascend method is used, with the correlation value between the hyperspectral and the *DEM* skylines as the metric, at convergence or at an arbitrary number of iteration the best position is selected in combination with the corresponding angles as the coordinates in *ERS* as the best metric.
In the particular acquisitions for this project the optimization process performed good enough, deliberately setting the starting position at around 200 meters from the ground truth, the optimization moved the estimated location to less than 80 meters from the real one.
Possible reasons for this results are discussed in sec. 6.

Figure 5: Optimization results vs ground truth.

5 Data projection

All the processes described until now come together to fulfill the descried project scope and to obtain a new *DEM* with the associated spectral information for each visible point.

5.1 DEM to reference sphere

The PRACTISE [3] package is a set of MATLAB routines originally developed to perform image-to-*DEM* projection and classification for snow-covering applications [5].
For this project the original PRACTISE code has been heavily modified and stripped down to the basic viewshed-computation subroutine and the transformation to map the visible points in the *DEM* onto the *SRF*.

5.2 reference sphere to hyper-spectral

The information provided by the PRACTISE run, as described in sec. 5.1, are saved in an array and parsed in the main Python process.
Having then the association between the visible points in the *DEM* and the *SRF*, the inverse projection process as the one described in sec. 2.2 is used to map the *DoA* to the nearest pixel in the hyper-spectral image.

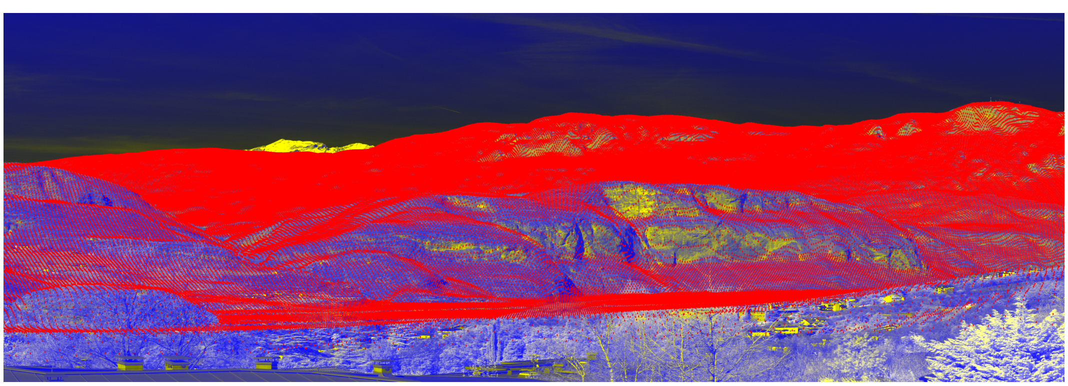


Figure 6: Projection of the visible *DEM* points on a false-color view of the hyper-spectral image.

With this information and after writing the corresponding hyper-spectral values in different layers of the *DEM* GeoTiff's file the procedure is complete.

6 Results and conclusions

The developed software for automatic projection of hyper-spectral images data onto a digital elevation model has demonstrated its capability to achieve a correct alignment with the precision allowed by the acquisition setup.
The results obtained through this alignment process provide a valuable foundation for further analysis and interpretation of the data.

Figure 7: RGB bands re-projected on the *DEM*, viewed from above the acquisition location.

While the achieved precision is decent, the true alignment is impacted by several factors, here briefly discussed:

- Skyline extraction algorithm: for both the *DEM* and the hyper-spectral image the algorithm aims at the extraction of only the true skyline, this is computationally easier but obviously the overall accuracy during the spherical matching process is lower with respect to a multi-ridge extraction from both of them, in conditions where the majority of skyline targets lay at similar distances from the acquisition location the parallax effect may simply be not enough to correctly optimize the position location of the camera, with multi-ridge extraction instead also the multiple ridges ad different heights and horizontal locations would play a role in determining the correct view angles and thus lead to a better location optimization result.

- *IRS* elevation data: The tracker described in sec. 2.1 provides two types of measures for *IRS* angles, an absolute one, with an angular resolution of approx. 0.33°, and a relative one, with an angular resolution of approx. 0.0002°.
  In the acquisition process the absolute one was used, believing that the angular resolution was enough for the problem at hand and avoiding the troublesome process of correctly calibrating the relative one to achieve better results with respect to the absolute one, the relatively low angular resolution translated to a slightly off angular direction on the *SRF*, this caused an imperfect alignment between the re-projected *DEM* and the hyper-spectral image, for the scope of this project this offset was empirically estimated, the solution while effective is still highly sub-optimal and the goodness of the correction was evaluated by sight on the plot in fig. 6, a better calibration step can be developed to avoid this problem.

- Spherical cross-correlation resolution: Due to hardware constraints during the project development the spherical coefficient computation was limited to order 300, with an higher maximum order correlation a more fine camera pose estimation can be achieved, with a subsequent better alignment between the *DEM* and the image.

- *DEM* position optimization: The optimization step of the position on the *DEM* was performed using a simple gradient-ascend algorithm, computing the gradients in a 10m radius around the current position and optimizing accordingly, a better search method, with a better estimation of the gradient to avoid optimizing over noise can be implemented to improve the performances of the process, anyhow the baseline presented with this project seems to suggest the possibility to use such technique to perform a good match from a coarse estimation of the position.

- Correct altitude data: the probable main cause of the optimization results presented in sec. 4.2 is the wrong on-*DEM* altitude estimation, in the code an offset is set over the estimated *DEM* altitude in the specified location, for the tests this was set to 1.5m, in reality the true offset with respect to the *DEM* is unknown, having performed the acquisition at the second floor of the university building this figure can be greatly skewed depending on the particular *DEM* used, an optimization process could also take into account this value in a specified range to obtain a better alignment.

- Loss function in skyline extraction and matching: In this implementation the skyline extraction produce a binary image with 0 value for non-skyline pixels and 255 for skyline pixels, this during the matching step cannot take into account how close a part of the skyline is if the alignment is not perfect, especially during the optimization step a loss function that can also take into account almost-alignment of the skyline (for example an inverse exponential function around the pixels) can improve the performances, smoothing out the correlation values for the positions in the *DEM* allowing for a better optimization process overall.

Overall the performances show the feasibility of the discussed technique to perform automatic image to *DEM* alignment and spectral information re-projection to acquire multi- or hyper-spectral data at specific raster resolution for fast acquisition of environmental data in mountainous terrain, the *DEM* can also be selected to achieve the same spatial resolution as satellite-based multi-spectral sensors to perform change detection using comparable data acquired in different ways to allow for a more flexible and dynamic approach.

6.1 Possible improvements

- Multiple crest extraction from *DEM* and Image for more robust correlation.
- GPU accelerated, filtered spherical cross-correlation for better performances.
- More efficient computation of *DEM* viewshed.
- More sophisticated *DEM* position optimization algorithm.
- Spectral correction for hyper-spectral data.
- More stable *IRS* elevation estimation.
- Optimization of height offset on *DEM*.

References

[1] Lionel Baboud, Martin Cadik, Elmar Eisemann, and Hans-Peter Seidel. Automatic photo-to-terrain alignment for the annotation of mountain pictures. *CVPR 2011*, Aug 2011.

[2] ccpem. Ccpem/soft · gitlab, Jan 2019.

[3] Stefan Härer. Shaerer/practise: V2.1.2, Dec 2015.

[4] Peter J. Kostelec and Daniel N. Rockmore. Ffts on the rotation group. *Journal of Fourier Analysis and Applications*, 14(2):145–179, 2008.

[5] SebBuchelt. Sebbuchelt/georef_webcam: Python tool to georeference camera images & project them into map coordinates, May 2020.