DATA ACQUISITION AND IMAGE PROCESSING FOR SOLAR IRRADIANCE FORECAST

Guillermo Terrén-Serrano  
Department of Electrical and Computer Engineering  
The University of New Mexico  
Albuquerque, NM 87131, United States  
guillermoterren@unm.edu

Manel Martínez-Ramón  
Department of Electrical and Computer Engineering  
The University of New Mexico  
Albuquerque, NM 87131, United States  
manel@unm.edu

November 26, 2020

ABSTRACT

The energy available in Micro Grid (MG) that is powered by solar energy is tightly related to the weather conditions in the moment of generation. Very short-term forecast of solar irradiance provides the MG with the capability of automatically controlling the dispatch of energy. To achieve this, we propose a method for statistical quantification of cloud features extracted from long-wave infrared (IR) images to forecast the Clear Sky Index (CSI). The images are obtained using a data acquisition system (DAQ) mounted on a solar tracker. We explain how to remove cyclostationary bias in the data caused by the devices in the own DAQ. We investigate a method to obtain the CSI, after the detrending of Global Horizontal Irradiance (GHI) measurements. We propose a method to fusion multiple exposures of circumsolar visible (VI) light images. We implement a method for extracting physical features using radiometric measurements of the IR camera. We introduce a model to remove from IR images both the effect of the atmosphere scatter radiation, and the effect of the Sun direct radiation. We explain how to model of diffuse radiation of the IR camera window, which is produce by water spots and dust particles stack to the germanium lens of the DAQ enclosure. The frames, that were used to model the camera window, are selected using an atmospheric condition model. This model classifies the sky four different categories: clear, cumulus, stratus, and nimbus. We introduce a geometric transformation of the size of the pixels to their actual dimension in a plane of the atmosphere which is at a given height. This transformation is performed according to the elevation angle of the Sun and field of view (FOV) of the camera. We compare the error between the transformation and anapproximation of transformation.

Keywords  Sky Imaging · Machine Learning · Long-wave Infrared Camera · Solar Forecasting

Contents

1 Technical Specification
1.1 Infrared Sensor ........................................... 2
1.2 Visible Sensor ........................................... 3
1.3 Solar Tracker ............................................ 3
1.4 Pyranometer ............................................. 4

2 Data Acquisition
2.1 Sun Position ............................................... 5
2.2 Noise Attenuation ........................................ 7
2.3 Exposure Switching

3 Data Processing
3.1 Physical Model of Global Solar Irradiance
3.2 Pyranometer Bias Correction
3.3 Visible Image Fusion

4 Girasol Datasets Description

5 Infrared Radiometry
5.1 Moist-Adiabatic Lapse Rate
5.2 Cloud Base

6 Atmospheric Radiation Model
6.1 Solar Radiation Models
6.2 Optimal Parameters
6.3 Model of the Parameters
6.4 Atmospheric Model Application

7 Window Model
7.1 Atmospheric Condition Model
7.2 Persistent Window Model
7.3 Window Model Application

8 Perspective Transformation
8.1 Rectilinear Lens
8.2 Flat Earth Approximation
8.3 Great Circle Approach
8.3.1 Length of a Segment Approximation

1 Technical Specification

We built a system composed of Visible (VI) and Infrared (IR) solar radiation cameras, a tracking system, and a pyranometer, to collect the data. A weatherproof enclosure contains the two USB cameras, and it is mounted on top of two servomotors. The cameras are connected to a motherboard that runs a solar tracking algorithm. This motherboard is placed inside of a different weatherproof enclosure together with a router, a servomotors control unit, a data storage system, plus their respective power supplies. The enclosure’s degree of protection is IP66. This is an international standard utilized for electronic equipment that provides protection against dust and water.

1.1 Infrared Sensor

The IR sensor used to capture the images is a FLIR© Lepton 2.5 camera\(^1\), which is mounted on Pure Thermal 1 board\(^2\) manufactured and distributed by Group Gets. The Lepton 2.5 sensor produces thermal imaging by measuring long-wave infrared. It captures IR radiation with a nominal response wavelength band from 8 to 14 microns.

\(^1\)http://www.flir.com
\(^2\)https://groupgets.com
The dimensions of a Lepton 2.5 are 8.5 x 11.7 x 5.6 mm. It has 51° HFOV and 63.5° diagonal with lens type f1.1 silicon doublet. The resolution is 80 horizontal x 60 vertical active pixels. The thermal sensitivity is < 50 mK. This module model integrates digital thermal image processing functions that includes automatic thermal environment compensation, noise filter, non-uniformity correction and gain control. This sensor can produce on stream an image in < 0.5 seconds. It operates at 150 mW nominally, and has a low power standby mode. The Pure Thermal 1 board adds functionalities to the IR sensor such as thermal video transmission over USB which works with USB Video Class (UVC) library on Windows, Linux, Mac and Android. It also includes on-board image processing without needing for an external system. The microprocessor is a STM32F411CEU6 ARM. It also includes a thermopile contactless temperature sensor to manually calibrate a Lepton module. The operational temperatures for Flat Filed Correction (FFC) function range from −10°C to 65°C. The captured images are digitalized in Y16 format. It is a single channel format that only quantifies intensity levels.

1.2 Visible Sensor

The VI camera is a 5 megapixels USB 2.0 color sensor manufacture by ELP©. The sensor is an OV5640 with maximum resolution of 2592 horizontal x 1944 vertical pixels and 170° FOV with fisheye type lens that is adjustable within the range of 2.1 to 6 mm. The pixel’s size is 1.4 x 1.4 microns, and the image area is 3673.6 x 2738.4 microns. The image stream rate is 30 frames per seconds at 640 x 480 pixels’ resolution and using MJPEG compression format. The communication protocol is UVC and its interface is USB 2.0 high speed. The dynamic range is 68 dB, and it has a mount-in shutter that can control frame’s exposure time. The camera-board has built-in functions that can be enable for Automatic Gain Control (AGC), Automatic Exposure Control (AEC), Automatic White Balance (AWB) and Automatic Black Focus (ABF). The sensor’s brightness, contrast, hue, saturation, sharpness, gamma, white balance, exposure and focus can be adjusted by software. Power consumption in VGA resolution is 150mW. The dimensions of camera board are 38 x 38 mm. The recommended operational temperatures for stable images range from 0°C to 60°C. The camera’s output format is YUYV for images and MJPEG for video. YUYV is 3 channels format in the color space of YUV format which represent luminance intensity, blue color and red color, respectively. MJPEG is lossy compression format. UVC drivers can operate in Windows, Linux, MAC and Android.

1.3 Solar Tracker

The device used for the tracking is a commercial model manufacture by FLIR© and distributed by moviTHERM. The model is PTU-E46-70. The tracking system is driven by two servo motors mounted to allow rotation on tilt and pan axes. The tracker is rated for a payload of 4.08 kg. Hence, the entire system must not weight more than this rate. Rotational speed of the tracker is 60° per second, and it has a resolution of 0.003°.

The degrees of freedom of the tilt servomotor allows vertical maneuvers of 78° with vertical limit from the horizontal of −47° to +31°. The pan servomotor has more degrees of freedom. This allows tracking of the sun throughout a day. Its range is ±150°. The motors’ velocity is adjustable up to 0.003° per second. The tracker requires an input voltage range of 12-30 VDC. It consumes 13 W in full power mode, 6 W low-power mode and 1 W in holding power off mode. The system is controlled via Ethernet from a host computer. The control unit is connected to the servomotors via a DB-9 female connector.
Figure 2: Tripod and junction box installed on the roof area of an UNM’s facility. Solar tracker’s servo motors mounted on the tripod, and screwed to a metallic structure which counterbalances the own enclosure’s weight. The pyranometer is adjusted to a 3D printed support placed on top of the junction box’s rack. Germanium and neutral density lenses for the IR and VI camera respectively, to filter the light beams and protect the cameras from weather hazards.

Regarding to the mechanical description of the tracker. The system weight is 1.36 kg, its dimensions are 7.6 cm height x 13 cm width x 10.17 cm depth. The control unit weight 0.227 kg, and its dimensions are 3.18 cm height x 8.26 cm width x 11.43 cm depth.

1.4 Pyranometer

As part of the development, a pyranometer sensor has been designed and tested to provide a device with high portability and accuracy. Here it is presented the circuit design for a signal conditioning of the pyranometer sensor model LI-COR LI-200, and it is summarized the architecture, components and software. The prototype was designed to send data to a laptop, Raspberry Pi or any other computer through USB communications, to which a simple python script needs to be added and executed. For portability purposes, the designed board is provided with a 5 V power supply, so that Raspberry Pi can be also connected to it.

The design objectives were to condition and measure the signal of a pyranometer sensor by converting it to a 12-bits digital signal and saving the data by timestamps in a csv file. This is done in a daily basis between 5 am to 9 pm. The system must be portable, easy to setup, and reliable.

The pyranometer output voltage range from 0 to 20 mV, where 10 mV represent a radiation power density of 1,000 W/m². A 120 V power outlet is available to power the device. The system must be connected to the internet and it has logging capabilities.

Figure 3: Electronics components layout. This Print Circuit Board (PCB) was developed using EAGLE, and it shows what is the actual position of the electronic components on the manufactured board once it has been already assembled.
The designed solution uses a differential amplifier with 13 dB gain that conditions the signal to a DAC. A microprocessor is used to measure the signal output from the ADC, and transmits it by USB port. The board is connected to a Raspberry Pi to save the signal’s data in files. This Raspberry Pi is connected to the internet, so it is also able to transmit the data. The custom-built electronic board is shown in Figure 1, and the circuit schematics are shown in Figure 2.

2 Data Acquisition

The tracking system and DAQ software is currently operative. The software was programmed in single Python 2.7 script. The system is placed on the top roof area of the Mechanical Engineering building at UNM central campus (35.0821, -106.6259). The DAQ sessions can be visually monitored through a webpage. This webpage contained real-time data acquired from IR and VI sensors, the Sun’s position algorithm and measures from the pyranometer. All devices were interconnected via a LAN network built by DHCP server.

2.1 Sun Position

The sun position is calculated to update the servomotors’ position every second. The following set of equations was used to calculate azimuth and elevation angles[^1]. These angles correspond to the servomotors’ pan and tilt axes. The tracking system must be aligned to the true North or South depending local, geographic coordinates define as $(\lambda, \phi)$, which are longitude $\lambda$, and latitude $\phi$. It also must be leveled with respect to the horizontal and vertical axis. The first step is to calculate the Local Standard Time Meridian (LSTM) which is the reference sun time between the Greenwich meridian and another time zone. LSTM is calculated per the following equation:

$$\text{LSTM} = 15^\circ \cdot \Delta T_{GT M},$$

(1)

where $\Delta T_{GT M}$ is the difference of the Local Time (LT) from Greenwich Mean Time (GMT) and it is measured in hours. Equation of Time (EoT) is an empirical equation that considers the eccentricity of earth’s orbit and its axial tilt. It is expressed in minutes and is described by,

$$\text{EoT} = 9.87 \cdot \sin(2B) - 7.53 \cdot \cos(G) - 1.5 \cdot \sin(B),$$

(2)

where,

$$B = \frac{360}{365} \cdot (d - 81).$$

(3)

$B$ units are degrees and $d$ corresponds to the days since the beginning of the year. The variation of the LST in a local effect due to a different longitude is account by the Time Correction Factor (TC), is which quantified in minutes, and its equation is

$$\text{TC} = 4 \cdot (\lambda - \text{LSTM}) + \text{EoT},$$

(4)

the constant 4 is related to the rotation of the earth, which rotates 1° per 4 minute, and $\lambda$ is the longitude. Local Solar Time (LST) is calculated by using the previous correction and Local Time (LT). Following equation is expressed in hours

$$\text{LST} = \text{LT} + \frac{\text{TC}}{60}.$$ 

(5)

The Earth rotates 15° per hour, so each hour away from the solar noon corresponds to an angular position. The Hour Angle (HRA) converts LST to the degrees that the sun moves on the sky. Before noon, angles are negative and after, the angles obtained are positive. The expression for this is:

$$\text{HRA} = 15^\circ \cdot (\text{LST} - 12).$$

(6)

The Sun’s declination angle $\delta$ varies seasonally, and it accounts for the angular difference between the equator and the Earth’s center. $\delta$ is obtained from this formula,

$$\delta = 23.45^\circ \cdot \sin \left(\frac{360}{365} \cdot (d - 8)\right),$$

(7)

where $d$ represents the day since the beginning of the year. The rotations on the tracker’s pan and tilt axes are given by Elevation $\epsilon$ and Azimuth $\alpha$ angles. The Elevation angle quantifies the Sun’s angular height, and the azimuth $\alpha$ is the Sun’s angular position on the horizon. The angles are calculate from these equation,

$$\epsilon = \sin^{-1} \left[ \sin \delta \cdot \sin \phi + \delta \cdot \cos \phi \cdot \cos(\text{HRA}) \right],$$

(8)

$$\alpha = \cos^{-1} \left[ \frac{\sin \delta \cdot \cos \varphi - \cos \delta \cdot \sin \varphi \cdot \cos(\text{HRA})}{\cos \xi} \right],$$

(9)

[^1]: http://www.pveducation.org/
Figure 4: ADC circuit’s schematics. The drawings show the connections for the all parts in the circuit. The analog circuit which measures the voltage level of the pyranometer is plotted on the top picture. On the bottom picture details the interconnections from the microprocessor to all circuit parts.
where $HRA$ is the hour angle, $\phi$ is latitude, and
\[
\xi = \sin^{-1} \left[ \sin \delta \cdot \sin \phi + \delta \cdot \cos \phi \cdot \cos(HRA) \right].
\] (10)

Alternatively, $\xi$ can be also calculated such as $\xi = 90^\circ + \phi - \delta$. The Zenith angle $\zeta$ is the difference between the Elevation angle $\epsilon$ and the vertical, so that
\[
\zeta = 90^\circ - \epsilon.
\] (11)

The tracking system initializes an hour after sunrise and stops an hour before sunset. This considers that sun’s elevation is too low to generate energy by PV systems. Inverters require a minimum consumption of power which is reached when it is above and below these time limits. The sunrise $\rho$ and sunset $\sigma$ are expressed in decimal hours and they are given by following equations,
\[
\rho = 12 - \frac{1}{15^\circ} \cdot \cos^{-1} \left( \frac{-\sin \phi \cdot \sin \delta}{\cos \phi \cdot \cos \delta} \right) - \frac{TC}{60^\circ},
\] (12)
\[
\sigma = 12 + \frac{1}{15^\circ} \cdot \cos^{-1} \left( \frac{-\sin \phi \cdot \sin \delta}{\cos \phi \cdot \cos \delta} \right) - \frac{TC}{60^\circ},
\] (13)

where $TC$ is the time correction.

\section*{2.2 Noise Attenuation}

The captures from the IR cameras are noisy. So, we propose to attenuate the noise by averaging $N$ consecutive frames taken at capture $k$. Our cameras fps are lower than the computer socket reading speed, so we need to assure that the frame $x^f$ read from the buffer is new, or it exits in our set. We define a capture as a set of frames $\mathcal{X} = \emptyset$. When we retrieve the first $x^f$, we add it to our set $\mathcal{X} = \{x^f\}$. We now read new $x^f$, and to find out if our frame $\exists x^i \in \mathcal{X}$, we calculate the normalized cross-correlation $\rho_{x^i, x^f}$ between each frames $x^i \in \mathcal{X}$, and $x^f$. Then, if $\{\rho_{x^i, x^f}\}_{i=1}^{f} \neq 1$, $\mathcal{X} = x^i \cup x^f$, $\forall i = \{1, \ldots, f\}$. We retrieve frames $x^f$ until $f = N$. The formula for the normalized cross-correlation $\rho_{x^i, x^f}$ is
\[
\rho_{x^i, x^f} = \frac{(x^i - \bar{x}) \cdot (x^f - \bar{x})}{\sqrt{(x^i - \bar{x})^2 \cdot (x^f - \bar{x})^2}},
\] (14)

where $\bar{x} = \frac{1}{N^D} \cdot \sum_{i=1}^{N} \sum_{j=1}^{D} x_{ij}$, which is the variable sample mean. Now, we define $\mathcal{I}^k$ as the set of different frames from capture $k$ such as
\[
\mathcal{I}^k \in \mathbb{N}[1, 2^{16}], \quad \forall i = 1, \ldots, N.
\] (15)

We average them together to attenuate the noise such as $\bar{I}^k = \frac{1}{N} \cdot \sum_{i=1}^{N} \mathcal{I}^k$. So now, we have a single frame that is defined as $\bar{I}^k \in \mathbb{N}[1, 2^{16}]$.

Figure 5: This figure shows the infrared images after applying the noise attenuation algorithm. The intensity of the pixels in the images is displayed in log-scale. The resolution of the images is $60 \times 80$.

\section*{2.3 Exposure Switching}

The scattering effect produced by the Sun radiation in the VI images avoids to detect the clouds in the circumsolar region, when the exposition time in capture is large. The light in the outer region of the image is dim when the exposition time is short. In order to sort this problem, we propose to switch the exposition time in each VI capture $x^f$ at the same
time that we attenuate the noise as we did with the IR captures. So, we have a set of $N$ frames with $E$ exposure times for each VI capture $k$ such as
\[
I_{i,j}^k \in \mathbb{N}^{[1,28]}, \quad \forall i = 1, \ldots, N, \quad \forall j = 1, \ldots, E.
\]
(16)
We average the $N$ frames for each $E$ exposition time to attenuate the noise
\[
\bar{I}_j^k = \frac{1}{N} \cdot \sum_{i=1}^{N} I_{i,e}^k.
\]
And, we obtain $E$ frames $\bar{I}_j^k \in \mathbb{N}^{[1,28]}$ per VI capture.

3 Data Processing

The pyranometer’s measurements has two error that we name as amplitude and shifting errors. The amplitude error is due to the calibration of the device, the pyranometer measures GSI at a higher $W/m^2$ than the real GSI. The shifting error is due to inclination of the pyranometer’s support platform. The pyranometer Li-200 is calibrated with incidence angle correction, so that functionality we were capable of recover the days that has a shifting in the measurements because of the artificial increment in real incidence angle that the support inclination produces.

3.1 Physical Model of Global Solar Irradiance

The GSI on a flat surface normal to the ground has 3 components [1],
\[
I_{GSI} = I_{Direct} + I_{Diffuse} + I_{Reflected}.
\]
(17)
The Direct component, when the body of reference is normal to the ground, is a function of the Sun’s elevation angle $\epsilon$,
\[
I_{Direct} = I_{DN} \cdot \sin \epsilon,
\]
(18)
and the Direct Normal (DN) component, that is
\[
I_{DN} = \theta_1 \cdot \exp \left( -\frac{\theta_2}{\sin \epsilon} \right),
\]
(19)
$\theta_1$ is the surface altitude with respect to the see level, and $\theta_2$ is the air mass coefficients.
The Diffuse component is the solar irradiance scattered by hitting particles in the atmosphere, as the water droplets forming clouds. The expression for a flat surface normal to the ground is
\[
I_{Diffuse} = \theta_3 \cdot I_{DN}^2.
\]
(20)
The coefficient $\theta_3$ defines the proportion of irradiance that is scattered for an atmospheric condition.
The Reflected component depends on the surfaces tilt angle. For our case, as the tilt angle is $0^\circ$, that means that is a flat surface normal to the ground, the expression is
\[
I_{Reflected} = \theta_4 \cdot I_{DN} \cdot (\theta_3 + \sin \epsilon),
\]
(21)$\theta_4$ is the reflective coefficient of the ground material where the surface is placed normal to it.
Finally, we can put together this three equations and obtained the solution for $I_{GSI}$ as function of the coefficients set $\theta^{(1)} = \{\theta_1, \ldots, \theta_4\} \in \mathbb{R}$ and the Sun elevation angle $\epsilon$, as
\[
I_{GSI} (\theta^{(1)}, \epsilon) = I_{DN} \cdot \sin \epsilon + \theta_3 \cdot \frac{I_{DN}}{2} + \theta_4 \cdot I_{DN} \cdot (\theta_3 + \sin \epsilon),
\]
(22)
where $I_{DN} = \theta_1 \cdot \exp \left( -\frac{\theta_2}{\sin \epsilon} \right)$.
Now, we obtain these coefficients $\theta^{(1)}$ from a set of theoretical formulae,
\[
\theta_1 = 1160 + 75 \cdot \sin \left( \frac{360}{N} \cdot (d - 275) \right),
\]
(23)
\[
\theta_2 = 0.174 + 0.035 \cdot \sin \left( \frac{360}{N} \cdot (d - 27) \right),
\]
(24)
\[
\theta_3 = 0.095 + 0.4 \cdot \sin \left( \frac{360}{N} \cdot (d - 27) \right),
\]
(25)
that are in function of the year day $d$, and the days on a year $N$ [1]. The coefficient $\theta_4$ amounts for the reflected radiation. As our pyranometer is mounted on a black surface, we assume that $I_{Reflected} = 0$, for that we set $\theta_4 = 0$. 

Figure 6: The graphs on the left column shows the GSI measurements from the pyranometer, the empirical coefficients in yellow, and the theoretical coefficients in red. The column on the right shows the GSI measurements, and theoretical model amplified by a $\sigma_d$. 
3.2 Pyranometer Bias Correction

• Pyranometer Amplitude Adjustment

The amplitude of the pyranometer’s measurements does not match the theoretical solar radiation and the measures are noisy at the maximum amplitude point, see Figure [6]. In order to overcome the problem of identifying this point, we propose to optimize the theoretical model to fit the measurements instead, this optimization problem is described in Physical Model Parameters’ Adjustment. In this way, we will obtain two smooth irradiance functions but with different amplitudes. We proceed to calculate the attenuation in the amplitude between the function evaluated with the theoretical coefficients \( \theta_d^{(1)} \), and the function evaluated with the optimized model coefficients \( \hat{\theta}_d^{(1)} \),

\[
\sigma_d = \frac{\max \left[ I_{GSI} \left( \hat{\theta}_d^{(1)}, \epsilon_d \right) \right]}{\max \left[ I_{GSI} \left( \theta_d^{(1)}, \epsilon_d \right) \right]}, \quad \forall d = 1, \ldots, D, \quad \sigma_d \in \mathbb{R}^{[1,2]}.
\]

(26)

The amplitudes samples \( \sigma_d \) are the dots in blue color in the left graph of figure [7]. We can observe that the amplitude bias follows a periodic function, so that we propose to model the bias by cycle-stationary sinusoidal function that is,

\[
C \left( \theta_1^{(2)}, d \right) = \theta_1 \cdot \sin \left( \theta_2 + \frac{2\pi d}{N} \right) + \theta_3,
\]

(27)

where \( N \) is the days on a year, \( d \) is the day of the year, and \( \theta^{(2)} = \{ \theta_1, \theta_2, \theta_3 \} \in \mathbb{R} \) is the parameters’ set.

We propose to correct the pyranometer’s measurements such as,

\[
R_{GSI} \left( \hat{\theta}_d^{(1)}, \epsilon_d, k \right) = \frac{y_{d,k}}{\sigma_d}, \quad y_{d,k} \in \mathbb{R}^+,
\]

(28)

and find the optimal set of parameters \( \theta^{(2)} \) that minimize the error function,

\[
\hat{\theta}^{(2)} = \arg \min_{\theta^{(2)}} E \left( \theta^{(2)} \right).
\]

(29)

We define the total sum of residuals as the error function,

\[
E \left( \theta^{(2)} \right) = \frac{1}{DK} \sum_{d=1}^{D} \sum_{k=1}^{K} |r_{d,k}|,
\]

(30)

and we calculate the residuals of our model’s fit as,

\[
r_{d,k} = \frac{y_{d,k}}{\mathcal{P} \left( \theta^{(2)}, d_d \right)} - I_{GSI} \left( \hat{\theta}_d^{(1)}, \epsilon_k \right), \quad r_{d,k} \in \mathbb{R},
\]

(31)

where \( y_d \) is the pyranometer’s measure, and \( \epsilon_d \) is the Sun’s elevation angle, and \( d_d = \{ d_d | d_d \in \mathbb{N}_{[1,N]}, d = 1, \ldots, D \} \) is the day of the year. We minimize the error function using the gradient w.r.t. \( \theta^{(2)} \) that is,

\[
\frac{\partial E \left( \theta^{(2)} \right)}{\partial \theta_i^{(2)}} = - \frac{1}{DK} \sum_{d=1}^{D} \sum_{k=1}^{K} \left[ I_{GSI} \left( \theta_d^{(1)}, \epsilon_d \right) \frac{r_{d,k}}{\left| r_{d,k} \right|} \frac{\partial \mathcal{P} \left( \theta^{(2)}, d_d \right)}{\partial \theta_i^{(2)}} \right].
\]

(32)

The results of model’s fit are displayed in the right graph of figure [7].

• Pyranometer Shifting Adjustment

The shifting caused by the inclination of the pyranometer’s support is corrected calculating the correlation between the theoretical GSI and the real GSI, and finding where is the maximum,

\[
t_{\max} = \arg \max \text{ Corr } \left( \mathcal{I} \left[ t \right], \mathcal{R} \left[ t \right] \right),
\]

(33)

so that \( \Delta t = \frac{t_N}{2} - t_{\max} \) is the displacement, and the sign \( \left( \Delta t \right) \) indicates the shifting direction in the time series.
Figure 7: In the left plot appears the estimated amplitude $\sigma_d$ for each clear day $d$ in blue. The proposed model $P(\theta^{(2)})$ is in red. In the right plot appears the piecewise model fit to shifts in the time series.

Once we know the displacement in the maxima, we shift the theoretical irradiance $I$ to detrend the pyranometer's measurements $R$ and obtain the CSI. If know multiply the CSI by the theoretical irradiance without shift, we have the corrected pyranometer's measurements $R'$:

$$R'[t] = \begin{cases} R[\Delta t : t_N] & \text{sign}(\Delta t) = 1 \\
R[1 : t_N - \Delta t] & \text{sign}(\Delta t) = -1 \end{cases} \quad \forall t = 1, \ldots, t_N - \Delta t \quad (34)$$

Despite the fact of the correction, there is information loss. The correct time series $R'$ is $\Delta t$ shorter. However, this do not suppose as problem as we only use information when $\epsilon > 15^\circ$.

We approximate the shifts in clear sky days by a piecewise model, see Figure 7. The piecewise model has 4 constant levels that are the mean of the shifts at the same level. We process the measures in the dataset to have a shift according to this model.

### Physical Model Numerical Parameters Adjustment

To adjust the physical model, we have a data set of $D$ days of clear sky. In each day, the data set is composed by $K_d$ measurements of irradiance $y_d = \{y_{d,k} | y_{d,k} \in \mathbb{R}^+, k = 1, \ldots, K_d\}$, and their corresponding elevation angles $e_d = \{\epsilon_{d,k} | \epsilon_{d,k} \in [0, \pi/2], k = 1, \ldots, K_d\}$. We solve the following optimization problem to estimate each day optimal set of a parameters $\hat{\theta}_d^{(1)}$:

$$\hat{\theta}_d^{(1)} = \arg \min_{\theta_d^{(1)}} \mathcal{E}(\theta_d^{(1)}), \quad \forall d = 1, \ldots, D. \quad (35)$$

The function that we propose to minimize is the Root Mean Square Error (RMSE),

$$\mathcal{E}(\theta_d^{(1)}) = \frac{1}{K} \sum_{k=1}^{K} \sqrt{(y_{d,k} - I_{GSI}(\theta_d^{(1)}, \epsilon_{d,k}))^2}. \quad (36)$$

The error function’s gradient w.r.t $\theta_{d,i}$ is,

$$\frac{\partial \mathcal{E}(\theta_d^{(1)})}{\partial \theta_{d,i}} = \frac{1}{K} \sum_{k=1}^{K} \left[ \frac{y_{d,k} - I_{GSI}(\theta_d^{(1)}, \epsilon_{d,k})}{y_{d,k} - I_{GSI}(\theta_d^{(1)}, \epsilon_{d,k})} \frac{\partial I_{GSI}(\theta_d^{(1)}, \epsilon_{d,k})}{\partial \theta_{d,i}} \right]. \quad (37)$$

### Steepest Descent

We propose to estimate the each parameters’ set $\theta^{(j)} = \{\theta^{(1)}, \theta^{(2)}, \theta^{(3)}\}$ by Steepest Descent [2]. So, we start by a random initialization of the parameters such as $\theta_0^{(j)} \sim \mathcal{U}(0, 1)$, and we update the parameters following this formula that uses the function gradient,

$$\theta_{i,n+1} = \theta_{i,n} - \eta \cdot \frac{\partial \mathcal{E}(\theta^{(j)})}{\partial \theta_{i,n}}, \quad \forall n = 0, \ldots, N, \quad (38)$$
Figure 8: The graphs in the left show the displacement in the pyranometer’s time series with respect to the theoretical GSI. The graphs in the right show the results after correcting the time series.
where $\eta \in \mathbb{R}^+$ is a very small number. Therefore, we might have a series of parameters’ updates that converge to a function minimum such as,

$$
\hat{\theta}_N^{(j)} = \arg \min_{\theta^{(j)}} \left\{ \mathcal{E} \left( \theta_n^{(j)} \right) \geq \mathcal{E} \left( \theta_{n+1}^{(j)} \right), \ldots, \mathcal{E} \left( \theta_N^{(j)} \right) < \mathcal{E} \left( \theta_{N+1}^{(j)} \right) \right\}.
$$

(39)

### 3.3 Visible Image Fusion

We introduce a method to fusion images with different exposition to generate a High Dynamic Range (HDR) of 16 bits, that combines information of the clouds from captures with different exposure time. The images noise was attenuated before hand as explained. To merge the images we first regularized the RGB components in each exposition to avoid division by zero when light is too dim,

$$
\mathbf{I}_{e,c}^k + \lambda, \quad \mathbf{I}_{e,c}^k \in \mathbb{R}^{D \times D}
$$

and we convert the images to gray scale by a weighted sum of the image RGB channels with corresponding Luma coding system coefficients that are $\beta_e = \{0.299, 0.587, 0.114\},$

$$
\mathbf{I}_e^k = \sum_{c=1}^{C} \mathbf{I}_{e,c}^k \cdot \beta_c, \quad \forall c = 1, 2, 3.
$$

(41)

The method propose is based on the distance from a pixel to the Sun. We know that the Sun is always centered in the frames thereby we define our fusion mask as a function of the radial distance of the Sun to a pixel such as,

$$
\mathcal{M}(r) = \left[ (\mathbf{X} - x_{sun})^2 + (\mathbf{Y} - y_{sun})^2 \right]^{1/2} \leq r, \quad \mathcal{M} \in \mathbb{2},
$$

(42)

notice that we are using boolean algebra. We define a mask $\mathbf{M}_e$ for each exposition time $r$ that as a constant distance to the Sun $\rho_e$.

$$
\mathbf{M}_e = \mathcal{M}(\rho_e), \quad \forall \rho_e = \{\rho_2, \cdots, \rho_E\}
$$

(43)

the set of fusion mask is $\mathbf{M}^m = \{0, \mathbf{M}^2, \cdots, \mathbf{M}^E, 1\}$. We apply a Gaussian function in sliding window,

$$
\mathcal{G}(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right),
$$

(44)

to blur the masks’ pixel in edges with the objective of smoothing out the transition between images regions in the resulting merged image,

$$
\mathbf{M}_{m,j} = \frac{1}{C} \sum_{x} \sum_{y} \left[ \mathbf{M}_{m,x,y} \odot \mathbf{G}_{x,y} \right], \quad \forall x, y \in \mathbf{W}_{N \times N}, \quad \forall m \in \{2, \ldots, E\},
$$

(45)

where normalization constant $C = \sum_{x=1}^{x=N} \sum_{y=1}^{y=N} \mathbf{G}_{x,y}$. Notice that the masks $\tilde{\mathbf{M}}$ are not longer Boolean $\tilde{\mathbf{M}}^m = \mathbb{R}^{D \times D}$.

We define ring-shape masks $\mathbf{R}_e^1$ and $\mathbf{R}_e^2$. We use $\mathcal{M}(r)$ function as we did before, but applying logical-sum operation with a circular mask with a bit smaller and larger radii $\epsilon = 2$, to produce the ring such as,

$$
\mathbf{R}_e^1 = \mathcal{M}(\rho_e) \oplus \mathcal{M}(\rho_e + \epsilon), \quad \forall \rho_e = \{\rho_2, \cdots, \rho_E\}
$$

(46)

$$
\mathbf{R}_e^2 = \mathcal{M}(\rho_e) \oplus \mathcal{M}(\rho_e - \epsilon), \quad \forall \rho_e = \{\rho_1, \cdots, \rho_E\}.
$$

(47)

The objective of the ring-shape masks is to select only adjacent pixels between regions different regions to weigh each image and produce smooth transitions. The weights are computed for each image applying the ring shape masks,

$$
\alpha_{e+1}^k = \alpha_e^k \frac{\sum_{i} \sum_{j} \mathbf{I}_e^k \odot \mathbf{R}_e^2}{\sum_{i} \sum_{j} \mathbf{I}_{e+1}^k \odot \mathbf{R}_e^2}, \quad \forall e \in \{1, \cdots, E - 1\}, \quad \alpha_e^k \in \mathbb{R}.
$$

(48)

We obtain a set of weights $\alpha_e^k = \{\alpha_1^k, \cdots, \alpha_E^k\}$ for each frame $k$. And, we proceed to merge the images with this formula,

$$
\mathbf{X}^k = \sum_{e=1}^{E} \frac{1}{e} \left[ \tilde{\mathbf{M}}_{e+1} \odot (1 - \mathbf{M}_e) \odot \left( \sum_{i=1}^{e} \alpha_e^k \cdot \mathbf{I}_i^k \right) \right], \quad \mathbf{X}^k \in \mathbb{R}^{D \times D}
$$

(49)
Figure 9: Each column shows an different image that were captured using different exposure times, that are displayed by column. The last row shows the images after computing the fusion algorithm. The resulting images are displayed in logarithmic-scale because of the large intensity of the pixels in the circumsolar region. Their size is 450 × 450, pixels without information are not displayed.
The image obtained in not normalized, so we normalize it by diving the image by a large number and multiply by the number of bits we want in the the image.

\[ I^k = \left( \frac{X^k}{225} \right) \cdot 2^{16}, \quad I^k \in \mathbb{R}^{D \times D} \]  

(50)

as we want a 16 bits image, we have to make sure there are not pixel above \(2^{16}\), and we convert the real number of the pixels to the natural numbers encounter on a 16 bits image, such as

\[ I^k_{16\text{bits}} = \begin{cases} I^k_{i,j} & I^k_{i,j} \leq 2^{16} \\ \text{null} & I^k_{i,j} > 2^{16} \end{cases}, \quad \forall i,j = 1, \cdots, D, \quad I^k_{16\text{bits}} \in \mathbb{N}^{2^{16}}. \]  

(51)

The outer circular region in the fusioned image is set to an intensity value of 0, It displays artifacts produce by own lens’ support and case. Therefore has not relevant information for the prediction.

4 Girasol Datasets Description

The repository has available recordings from 240 days of 3 years days, that amount for 120Gb of data in total. The sampling interval of the cameras is 15 seconds when the Sun’s elevation angles is \(> 15^\circ\), so there are approximately 1,200 to 2,400 captures per day from each camera depending of the year day.

- VI images are 16 bits with resolution 450x450 with only intensity channel. Approx. 240Kb per frame. Between 200Mb to 400Mb per day depending on amount of clouds. Images are save in .png compress-less format in the directory */infrared.
- IR images are 16 bits with resolution 60x80 only intensity channel. Approx. 8Kb per frame. 20Mb per day roughly constant. Images are save in .png compress-less format in the directory */infrared.
- The pyranometer samples as many measures as it can in a second. This ranges from 4 to 6 samples per seconds. The measures are save in the directory */pyranometer in a .csv file with their date as name (yyyy-mm-dd), approx. 4,500Kb to 7,500Kb per file. The files contain unix time in the first column and GSI in \(W/m^2\) in the second column.
- The Sun position files are generate out of the time recorded in the pyranometer files. The positions are save in the directory */position in a .csv file with their date as name (yyyy-mm-dd), approx. 6,500Kb to 11,500Kb per file. The files contain the unix time in the first column, elevation angle in the second column, and azimuth angle in the third column.

5 Infrared Radiometry

The Lepton\(^4\) camera with Radiometry has a wavelength from 8 to 14 micros, and provides an uniform thermal image as the output. When applying the radiometry functionality, the camera recordings are turned into temperature measurements. To obtain the corresponding values of temperature from the intensity of a pixels, they have to be operated such as,

\[ T = \frac{I}{100} [^\circ K] \]  

(52)

for international system units, or as

\[ T = \frac{I}{100} - 273.15 [^\circ C], \]  

(53)

to obtain the temperature in Celsius degrees.

5.1 Moist-Adiabatic Lapse Rate

The temperature of a particle in the Troposphere is a function of the height. In contrast with the temperature in the Tropopause that can be considered approximately constant \(^3\). There is also a height of slightly increasing in temperature in the Stratosphere that will be deprecated in this investigation, as cloud phenomena is restricted to the

\(^4\)https://www.flir.com/
Troposphere. Above the Stratosphere range of height to the Exosphere the content of water vapor particles will be considered zero.

The presence of water in the Troposphere implies that, in the process of convection, a parcel of air which rises at the same time cools, and eventually saturates [4]. At this point, as the temperature continues decreasing with the height the water vapor condenses and forms clouds realizing heat [5]. This process is known as the water vapor adiabatic lapse, and while the dry adiabatic lapse rate is constant, the moist adiabatic lapse is not [6, 7]. This fact is important because in most climates the relative humidity is nonzero. The equation for the moist adiabatic lapse is,

$$\Gamma_{MALP} = g \frac{1 + \frac{L_v r_v}{R T}}{c_{pd} + \frac{L_v r_v}{R T^2}} = g \frac{R_{sd} T^2 + H_v r_v T}{c_{pd} R_{sd} T^2 + H_v^2 r_v \epsilon} \left[ \circ K/m \right]$$  \hspace{1cm} (54)

where:

- Earth’s gravitational acceleration, \( g = 9.8076 \text{ m/s}^2 \).
- Heat of vaporization of water \( H_v = 2501000 \text{ J/kg} \).
- Specific gas constant of dry air \( R_{sd} = 287 \text{ J/kg} \circ K \).
- Specific gas constant of water vapour \( R_{sw} = 461.5 \text{ J/kg} \circ K \).
- Dimensionless ratio of the specific gas constant of dry air to the specific gas constant for water vapour \( \epsilon = \frac{R_{sd}}{R_{sw}} = 0.622 \).
- Water vapour pressure of the saturated air, \( e = \epsilon \cdot \exp(\frac{7.5 \cdot T_{dew}}{273.3 + T_{dew}}) \cdot 100 \). The original formula is in hPa but this is in Pa.
- Mixing ratio of the mass of water vapour to the mass of dry air \( r_v = \frac{e - e}{p - e} \).
- Specific heat of dry air at constant pressure, \( c_{pd} = 1003.5 \text{ J/kg} \circ K \).

Therefore, the moist adiabatic lapse rate is fully described knowing the air temperature \( T \), the atmospheric pressure \( p \), and the dew point \( T_{dew} \).

### 5.2 Cloud Base

The base of a cloud can be estimated from ground measurements of air temperature, and dew point or relative humidity. This calculation of the lifted condensation level is based on the spread, that is the difference between the air temperature \( T \) and the dew point \( T_{dew} \),

$$h_{base} = 304.8 \cdot \frac{(T - T_{dew})}{2.5} \left[ \text{m} \right]$$  \hspace{1cm} (55)

The spread has to be divided by 4.4 (if temperatures are in \( \circ F \)) or 2.5 (if temperatures are in \( \circ C \)), and multiplied by 304.8 (if the heights is sought in meters) or by 1000 (if it is sought in feet). Some networks of weather stations can be remotely assessed to read real-time measurements. In fact, there are available either time series [5] or spatio-temporal weather data [6].

### 6 Atmospheric Radiation Model

The techniques to estimate the motion vectors in an image are sensitive to gradient in the pixels’ intensity. Therefore, it is necessary to remove the gradient which is produced by the solar direct radiation, and the atmospheric scatter radiation in order to extract dynamic features from the cloud. The diffuse or scatter radiation are emitted by water molecules or dust particles in the atmosphere. In particular, the scatter radiation from the atmosphere dome has systematical pattern that appear on the images in the course of a year.

#### 6.1 Solar Radiation Models

The Sun direct radiation is scattered by aerosol and water particles floating in the atmosphere [8]. The scatter radiation depends on the atmosphere air mass and clearness conditions. The air mass varies with the altitude, because the

---

5https://www.wunderground.com
6https://www.windy.com
gravity attracts particles concentrating them near the surfaces. The clearness depends on the weather conditions such as wind that disperse the aerosol particles. The effects produced by the scatter radiation from atmosphere, and the direct radiation, appear on a IR image.

The aim is to remove the scatter and direct radiation from the IR images, so that only the reflected radiation from clouds appears on the images. We define each pixel in a frame, as a point \( x = \{ x, y \} \) in a Cartesian coordinates system such \( \mathbf{X} = \{(1, \ldots, N) \mid x_i \in \mathbb{N}, i = 1, \ldots, M, j = 1, \ldots, N\} \), and \( \mathbf{Y} = \{(1, \ldots, M) \mid y_i,j \in \mathbb{N}, i = 1, \ldots, M, j = 1, \ldots, N\} \). We propose to model effect from the background atmospheric radiation using an exponential function with scale \( \sigma_1 \) and length-scale \( \lambda_1 \), centered in the Sun’s coordinates \( x_0 = \{ x_0, y_0 \} \) such as

\[
S(y_{i,j}; y_0, \sigma_1, \lambda_1) = \sigma_1 \exp \left( \frac{y_{i,j} - y_0}{\lambda_1} \right), \quad \sigma_1, \lambda_1 \in \mathbb{R}. \tag{56}
\]

The effect produced on the image by Sun’s direct radiation can be modeled by a bivariate quadratic exponential function that depends on the weather conditions, the date, and the Sun’s position in the horizon. In order to approximate the model, we have to define sufficient parameters to achieve an adequate fit in the frames recorded during all the seasons of a year, and at different times in a day. Because that, the parameters of the atmospheric radiation model are expected to be variables of a function that is a function of parameters’ set \( \theta = \{\sigma_1, \lambda_1, \sigma_2, \lambda_2\} \). It models the deterministic component of the radiation in the IR.

\[
\mathcal{A}(x_{i,j}; y_{i,j}; x_0, \theta) = \theta_1 \exp \left\{ \frac{y_{i,j} - y_0}{\theta_2} \right\} + \ldots \tag{59}
\]

\[
\ldots + \theta_3 \exp \left\{ \frac{\theta_4^2}{\sqrt{x_{i,j} - x_0)^2 + (y_{i,j} - y_0)^2 + \theta_4^2}} \right\}. \tag{60}
\]

This function is more flexible in the function height and tails. The multivariate Cauchy distribution function is

\[
\mathcal{C}(x_{i,j}; y_{i,j}; x_0, \sigma_2, \lambda_2) = \frac{1}{2\pi} \exp \left\{ \frac{\lambda_2}{\left[(x_{i,j} - x_0)^2 + (y_{i,j} - y_0)^2 + \lambda_2^2\right]^2} \right\}, \quad \lambda_2 \in \mathbb{R}. \tag{57}
\]

This distribution is transformed to a Lorentzian function multiplying by a scale parameter \( \sigma_2 \), that controls the density height. So, the function is now

\[
\mathcal{D}(x_{i,j}; y_{i,j}; x_0, \sigma_2, \lambda_2) = \sigma_2 \exp \left\{ \frac{\lambda_2^2}{\left[(x_{i,j} - x_0)^2 + (y_{i,j} - y_0)^2 + \lambda_2^2\right]^2} \right\}, \quad \sigma_2, \lambda_2 \in \mathbb{R}. \tag{58}
\]

This function is not a probability distribution, as its integral does not add to 1, but we do not intent to modeling probabilities, this is not important for us.

6.2 Optimal Parameters

The atmospheric model has to fit the radiation in frames recorded during all the seasons of a year, and at different times of the day. Because that, the parameters of the atmospheric radiation model are expected to be variables of a function that depends on the weather conditions, the date, and the Sun’s position in the horizon. In order to approximate the function that models the parameters, it is necessary to find which were the optimal parameters in frames previously recorded during different days of year, but it is also necessary that all of the frames were recorded in clear-sky conditions, so that only the radiation appears on the images.

If the intensity in a pixel is defined as \( I_{i,j}^{d,k} = \{ i_{i,j}^{d,k} \mid i_{i,j}^{d,k} \in \mathbb{N}^{[1,2^{16}], i = 1, \ldots, M, j = 1, \ldots, N} \), the Root Mean Squared Error (RMSE) function for sample frame \( k \) in a day \( d \) is

\[
\mathcal{E}(\theta_{d,k}) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sqrt{I_{i,j}^{d,k} - \mathcal{A}(x_{i,j}; y_{i,j}; x_0, \theta_{d,k})^2} \tag{61}
\]
The aim is to find the optimal set of parameters $\theta_k$ that minimizes the loss function,

$$\hat{\theta}_{d,k} = \arg\min_{\theta_{d,k}} \mathcal{E}(\theta_{d,k}), \quad \forall k = 1, \ldots, K, \forall d = 1, \ldots, D. \quad (62)$$

The gradient of error function $\mathcal{E}(\theta_{d,k})$ w.r.t. the parameters $\theta_{d,k}$ for any sample $d, k$ is,

$$\frac{\partial \mathcal{E}(\theta_{d,k})}{\partial \theta_{d,k}^i} = -\frac{1}{MN} \sum_{\ell=1}^{MN} \left[ \frac{\tilde{y}_{d,k}^\ell - A(\tilde{x}_{\ell}, \tilde{y}_{\ell}; x_{d,k}^0, \theta_k)}{\tilde{y}_{d,k}^\ell - A(\tilde{x}_{\ell}, \tilde{y}_{\ell}; x_{d,k}^0, \theta_k)} \frac{\partial A(\tilde{x}_{\ell}, \tilde{y}_{\ell}; x_{d,k}^0, \theta_k)}{\partial \theta_{d,k}^i} \right], \quad (63)$$

where $\tilde{y}_{d,k} = \text{Vec}(1_{d,k}^d)$, $\tilde{y} = \text{Vec}(Y)$, and $\tilde{x} = \text{Vec}(X)$ for notation simplification. The optimal parameters were found for a dataset consisting of 51 clear-sky conditions days selected out of a year long of samples. The number of sample-frames available in each day is different, because the number of day-light hours in each day are also different.

Figure 10: The graphs optimization results for the parameter $\theta^{(1)}$ in four different days $d$. The optimal $\hat{\theta}_{d,k}^{(1)}$ for a frame $k$ is displayed in color blue. The predicted $\hat{\theta}_{d,k}^{(1)}$ by the polynomial model is displayed in color red.

Figure 11: The graphs show the optimal parameter $\theta^{(2)}$ in four different days $d$. The $\hat{\theta}_{d,k}^{(2)}$ frame $k$ is displayed in color blue. The predicted $\hat{\theta}_{d,k}^{(2)}$ by the polynomial model is displayed in color red.

6.3 Model of the Parameters

Once the optimal set of parameters $\hat{\theta}_{d,k}$ are found for each sample $k$ in day $d$, the aim is to model these optimal sets of parameters $\hat{\Theta} = \{\theta_k \mid \theta_k \in \mathbb{R}^4, k = 1, \ldots, DK\}$ for any possible moment. In fact, these set of parameters has a physical interpretation. The parameter $\hat{\theta}^{(1)}$ is the average height of the Tropopause in the IR image. This height varies along the year as a function of the latitude and the weather conditions. The parameter $\hat{\theta}^{(2)}$ models the cycle-stationary pattern of the Earth curvature cross-section. The great circle of Tropopause is not perfectly spherical and influence by local and global climatic condition, so these patters are not easy to model as the IR images are spatial-temporal measurements.
The model's parameters of the Sun direct radiation are considered constant as the Sun's irradiance does not vary along the day in the IR images. The optimal values of \( \theta^{(1)} \) follows a normal distribution, and the ones of \( \theta^{(2)} \) are a uniform distribution. Therefore, the optimal value is computed as the sample mean, 

\[
\hat{\theta} = \frac{1}{D \cdot K_d} \sum_{d=1}^{D} \sum_{k=1}^{K_d} \theta_{d,k}.
\]  

(64)

In order to model \( \theta^{(1)} \) and \( \theta^{(2)} \), we know the Sun’s position is a function of the azimuth \( \alpha = \{ \alpha_k \mid \alpha_k \in \mathbb{R}^{[1,2\pi]}, k = 1, \ldots, DK \} \), and the elevation angle \( \varepsilon = \{ \varepsilon_k \mid \varepsilon_k \in \mathbb{R}^{[1,\pi/2]}, k = 1, \ldots, DK \} \). The cycle-stationary pattern correspond to the temporal sequence of days in a year \( \mathbf{d} = \{ d_k \mid d_k \in \mathbb{N}^{[1,365]}, k = 1, \ldots, DK \} \). The height of the Tropopause can be approximate by ground measurements, which can be obtained from a weather station that has available measurements of air temperature \( \mathbf{T}^{air} = \{ T_k^{air} \mid T_k^{air} \in \mathbb{R}, k = 1, \ldots, DK \} \) and dew point \( \mathbf{T}^{dew} = \{ T_k^{dew} \mid T_k^{dew} \in \mathbb{R}, k = 1, \ldots, DK \} \). Despite the fact that other variables, such as relative humidity or atmospheric pressure, can be available, no correlation was found with \( \theta_1 \) or \( \theta_2 \), consequently no further experimentation was carried out with them. The consistent features in the pattern vector for each parameters were found to be different, these feature vectors are \( \mathbf{x}_k^{(1)} = [T_k^{air} T_k^{dew} \varepsilon_k \alpha_k]^T \) and \( \mathbf{x}_k^{(2)} = [d_k \varepsilon_k \alpha_k]^T \) for any frame \( k \).

We propose to fit a model that applies a polynomial expansion of order \( n \) in the feature \( \mathcal{P} (\mathbf{x}_k^{(i)}) \), and that has coefficients \( \mathbf{w}^{(i)} \). The polynomial expansion formula implemented in the feature vector \( \mathbf{x}_k^{(1)} \), is,

\[
\mathcal{P} (\mathbf{x}_k^{(1)}) = \sum_{i,j,n,m}^P \mathbf{w}^{i,j,n,m} \cdot T_k^{air}^i \cdot T_k^{dew}^j \cdot \varepsilon_k^n \cdot \alpha_k^m, \quad \forall k = 1, \ldots, K \cdot D.
\]  

(65)

The expansion implemented in the feature vector \( \mathbf{x}_k^{(2)} \) has this formula,

\[
\mathcal{P} (\mathbf{x}_k^{(2)}) = \sum_{i,j,n}^P \mathbf{w}^{i,j,n} \cdot d_k^i \cdot \varepsilon_k^n \cdot \alpha_k^m, \quad \forall k = 1, \ldots, K \cdot D.
\]  

(66)

The formulation of our regression problem with a loss function applied to the polynomial expansion coefficients, which is a simplification of the Tikhonov’s regularization \([9]\), it is known as Ridge Regression (RR) when is implemented with the L-2 norm. This model’s objective function is,

\[
\min_{\mathbf{w}} \sum_{k=1}^{DK} \| \mathbf{y}_k - \mathcal{P} (\mathbf{x}_k) \|^2 + \lambda \cdot \| \mathbf{w} \|_2^2, \quad \forall k = 1, \ldots, K \cdot D.
\]  

(67)

The parameters \( \mathbf{w} \) that minimize loss function can be found analytically applying least squares,

\[
0 = \frac{\partial}{\partial \mathbf{w}} \left[ (\mathbf{y}_k - \mathcal{P} (\mathbf{x}_k))^T (\mathbf{y}_k - \mathcal{P} (\mathbf{x}_k)) + \lambda \cdot \text{tr} \left( \mathbf{w}^T \mathbf{w} \right) \right]
\]  

(68)

\[
0 = 2 \cdot (\Phi^T \mathbf{w} - \mathbf{y}_k) + \lambda \mathbf{w}
\]  

(69)

\[
\mathbf{w} = (\Phi^T \Phi + \lambda \cdot I)^{-1} \Phi^T \mathbf{y}.
\]  

(70)

The polynomial transformation of the features vectors is \( \varphi : \mathcal{X} \mapsto \mathcal{P}^n \), where \( n \) is the order of the polynomial expansion and the number of terms is \( \mathcal{P}^n = \frac{(n+d-1)!}{n!} \). When applying the transformation to a feature vector such as \( \mathbf{x}_k \mapsto \varphi (\mathbf{x}_k) \), \( \forall k \in (0,K] \), the resultant vector is transformed to a space of higher dimensions \( \Phi_k = [\varphi (\mathbf{x}_k)^1 \ldots \varphi (\mathbf{x}_k)^p] \in \mathbb{R}^{p \cdot n} \). The feature vectors \( \mathbf{x}_k^{(1)} \) or \( \mathbf{x}_k^{(2)} \) have a scalar dependent variable that is either \( \theta_k^{(1)} \) or \( \theta_k^{(2)} \).

The definition of the problem in matrix form contains the all training samples. The set of matrices is different for each model \( \mathcal{M}^{(i)} \), and those are,

\[
\mathbf{y}^{(i)} = \begin{bmatrix} \hat{\theta}_1^{(i)} \\ \vdots \\ \hat{\theta}_K^{(i)} \end{bmatrix}, \quad \Phi^{(i)} = \begin{bmatrix} \varphi (\mathbf{x}_1^{(i)})^1 & \cdots & \varphi (\mathbf{x}_1^{(i)})^p \\ \vdots & \ddots & \vdots \\ \varphi (\mathbf{x}_K^{(i)})^1 & \cdots & \varphi (\mathbf{x}_K^{(i)})^p \end{bmatrix}, \quad \mathbf{w}^{(i)} = \begin{bmatrix} \mathbf{w}_1^{(i)} \\ \vdots \\ \mathbf{w}_p^{(i)} \end{bmatrix}.
\]  

(71)
The prediction of parameter $\hat{\theta}_s(i)$ for a new observation $x_s(i)$ is computed independently using each models parameters such as,

$$\hat{\theta}_s(i) = \bar{w}^{(i)}, \varphi \left( x_s^{(i)} \right).$$

Leave-One-Out (LOO) Cross-Validation (CV) method was implemented to find the optimal polynomial expansion order $n$, and regularization parameter $\lambda$. This routine was independently computed for $\theta_1$ and $\theta_2$, since their are modelled using different feature vectors. The dataset contains 51 days with no drops of CSI. These days were selected from a year long set of weather and IR image recordings. 20% of those days were reserved for testing proposes, whilst the remaining set were the individual samples left aside for LOO validation during the training of the models. Notice that each sample set is a whole day $d$ that itself has $K_d$ recorded weather and inferred samples.

Figure 12: The validation of the model for $\theta_1$ shows that, the minimum error $\text{RMSE} (\theta_1) = 220.56$, is accomplish by a polynomial expansion of order $n = 2$, which uses as features the air temperature, dew point, and elevation angle.

Figure 13: The results of the validation perform in the model for $\theta_2$. The minimum error in test is $\text{RMSE} (\theta_2) = 143.14$ for a polynomial expansion of order $n = 6$. The optimal features were day of the year, elevation, and azimuth angles.

The atmospheric radiation can be approximated for a new observation $x_s$, knowing the set of parameters $\hat{\theta}_s$ and the Sun’s position is $x_0$, with the model $\hat{I}_s = \mathcal{A} \left( X, Y; x_0^k, \hat{\theta}_s \right)$, where $\hat{I}_s \in \mathbb{R}^{[1, 2^{16}]}$.

When a sequence of images is defined as space-time series such as $\mathbf{I}_k = \{ \mathbf{I}_k : k \in [1, \infty) \} \in \mathbb{R}^{[1, 2^{16}]}$, the atmospheric model for scatter and direct radiation is the deterministic component of the solar irradiance,

$$\mathbf{I}_k = \left\{ \mathcal{A} \left( X, Y; x_0^k, \hat{\theta}_s \right) : k \in [1, \infty) \right\} \in \mathbb{R}^{[1, 2^{16}]}.$$ (73)

Therefore, we propose to implement this model to detrend the 2D-dimensional time series, so it can be obtained an image that contains only information from the reflected radiation emitted by the clouds, see figure 14.
6.4 Atmospheric Model Application

The Sun has always constant intensity $\tau_0$ in the images unless a cloud is occluding it, so that the direct radiation has to be removed only when it is on the images. Therefore, when the Sun is detected on the image, the atmospheric radiation model $A(x_0, \theta^*)$ is applied, otherwise the scatter radiation model $S(y_0, \theta^*)$ is applied,

$$\bar{I}_k = \begin{cases} I^b_k - S(y_0, \theta^*) & \text{if } \max[\bar{I}_k] < \tau_0 \\ I^b_k - A(x_0, \theta^*) & \text{otherwise} \end{cases}$$ \hspace{1cm} (74)

When the model $A(x_0, \theta^*)$ is subtracted to the frames can be errors, see figure 15. The inaccuracies are due to the low resolution of the IR sensor, plus any other systematic errors added by the tracking system.

The interpolation of the pixels’ intensity which Euclidean distance is $\leq r_0$ from the Sun $x_0 = \{x_0, y_0\}$, can attenuate the effects produced by errors. The interpolation is performed using the remaining pixels in the image $x = \{(x_{i,j}, y_{i,j}) \in \mathbb{N} | i = 1, \ldots, M, j = 1, \ldots, r_{i,j} \geq r_0\}$, to find which one of these are the nearest neighbors,

$$\hat{x}_{i,j} = \arg\min \left| x^1 - x_{i,j} \right|, \quad \hat{y}_{i,j} = \arg\min \left| x^2 - y_{i,j} \right|, \quad \forall i, j \in \mathbb{N} \cap x_{i,j} \land y_{i,j} \leq r_0.$$

To the intensity of those pixels, that are in the circumsolar region defined as $x_{i,j} \land y_{i,j} \leq r_0$, is assigned their nearest neighbour’s intensity [10],

$$\bar{I}_k(x_{i,j}, y_{i,j}) = \bar{I}_k(\hat{x}_{i,j}, \hat{y}_{i,j}), \quad \forall i, j \in \mathbb{N} \cap x_{i,j} \land y_{i,j} \leq r_0.$$

where $r_0 = 3$ is the radius that we apply in the interpolation, see the results in figure 15.

The frame, after removing the direct and scatter atmospheric radiation, is a measure of the reflected radiation. This reflected component is the radiation emitted by objects such as clouds, streetlights, or building structures, but it emitted by particles attached to the window protecting the IR camera. The window is made out of germanium because is a material that allows the transmission of the IR radiation. In next section, we will explore how to numerically model the transmittance of a germanium window.

7 Window Model

The window of the camera gets dirty with water spots produced by dried droplets containing dust particles after rainy days, and dust particles suspended in the air that accumulate on the window over time. The window cannot be routinely cleaned because of difficult access to the DAQ localization. Therefore, we propose a persistent model to remove the effects produced by the dust particles and water spots on the window.

7.1 Atmospheric Condition Model

The aim is to classify the sky conditions in four possible categories: clear sky, cumulus, stratus, or nimbus cloud. This classification determine when the image has to be segmented, when the sky is clear, we do not need to segment, and
Figure 15: The graphs in each row correspond to a different day. In the left column, the graphs of the background model. The graphs in the middle display the raw pixels image. The right column graphs show the results after removing the constant background from the raw image.
when there are stratus or nimbus, the sky is entirely cover. Therefore, the segmentation has to be applied only if there are cumulus clouds. The classification is necessary when thin stratus cover the sky. In this situation, optimal segmentation is difficult to achieve, however we know that the entire image is cover with a thin layer a cloud.

We propose a persistent linear Support Vector for Classification (SVC) for this tasks due to the large number of clouds $\mathbf{X} = \{\mathbf{x}_i \in \mathbb{R}^D \ | \ \forall i = 1, \ldots, N\}$, and $\mathbf{y} = \{y_i \in [-1, +1] \ | \ \forall i = 1, \ldots, N\}$ is,

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\| + C \sum_{i=1}^{N} \xi (\mathbf{w}; \mathbf{x}_i, y_i)$$

(78)

where $C > 0$ is the penalty term, and $\xi (\mathbf{w}; \mathbf{x}_i, y_i)$ is the loss function that in our model is the L-2 for a SVC, so we have this,

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|_2 + C \sum_{i=1}^{N} \left( \max \left(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i\right) \right)^2.$$  

(79)

The solution in the dual formulation is the following constrained quadratic programming problem,

$$\min_{\alpha} \frac{1}{2} \alpha^\top \mathbf{Q} \alpha - \mathbf{1}^\top \alpha$$

(80)

s.t. $0 \leq \alpha_i \leq C, \ i = 1, \ldots, N$

(81)

where $Q_{i,j} = y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$, $\mathbf{1} = [1, \ldots, 1]^\top$, and $N$ is the number of samples.

The standard solution of a liner SVC is defined for two possible categories. We have four categories, so this is a multi-class classification problem. We propose to solve it implementing the one-versus-all scheme. Hence, there will have a different set of labels for each one of the linear SVC that is,

$$\hat{y}_{i,\ell} = \begin{cases} +1 & y_i = \ell \\ -1 & \text{Otherwise} \end{cases} \quad \forall i = 1, \ldots, N, \ \forall \ell = 1, \ldots, L.$$  

(82)

Notice that the bias is included in the weights, so for a new sample $k$ we obtain a classification such as,

$$\hat{y}_k = \arg \max_{\ell \in \mathcal{L}} \mathbf{w}_\ell^\top \mathbf{x}_k.$$  

(83)

The features vector includes the atmospheric pressure $p_k$ and the CSI $i_k$ collected by a weather station. In addition to a set of statistical metrics that are the mean, variance, kurtosis, and skewness. These were computed using the pixels’ of the radiometric measures of temperature in the IR image $\mathbf{T}_k$, and the magnitude of the velocity vectors $\mathbf{M}_k = \sqrt{\mathbf{U}_k^\top + \mathbf{V}_k^\top}$. The feature vector for each sample $k$ is,

$$\mathbf{x}_k = \left[p_k, i_k \ \mathbb{E} (\mathbf{T}_k) \ \mathbb{V} (\mathbf{T}_k) \ \mathbb{S} (\mathbf{T}_k) \ \mathbb{K} (\mathbf{T}_k) \ \mathbb{E} (\mathbf{M}_k) \ \mathbb{V} (\mathbf{M}_k) \ \mathbb{K} (\mathbf{M}_k) \ \mathbb{1}^\top \right]^\top,$$

(84)

the skewness of the magnitude is not included as the model under-performs when is included.

A polynomial expansion is applied to each vector such as $\mathbf{x}_k \rightarrow \varphi (\mathbf{x}_k) \in \mathbb{R}^{pn}$, where $n$ is the polynomial expansion of order, and the atmospheric condition classes are such as $y_k \in \{0, 1, 2, 3\}$. The dataset in matrix form is,

$$\mathbf{X} = \begin{bmatrix} \varphi (\mathbf{x}_1) \\ \vdots \\ \varphi (\mathbf{x}_N) \end{bmatrix}, \ \mathbf{X} \in \mathbb{R}^{N \times pn}, \ \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \ \mathbf{y} \in \mathbb{N}^N.$$  

(85)

In the experiments, we aim to find the polynomial expansion order $n$ and together with the combination of weather features that produces the highest accuracy in the atmospheric condition classification. We implement a leave-one-out (LOO) Cross-Validation (CV) to find $C$ for each $n$, and we compare the accuracy in validation of the linear SVC different models.

The different pre-classified atmospheric condition sequences that composed the dataset were divided into batches of the same size to be each of the samples in LOO-CV implementation. The samples were not scramble to preserve the time structure in the sequences. The sequences of samples belong to nonconsecutive 21 days out of a year of recordings, that
Figure 16: The graph shows the accuracy obtained in the classification during the validation of the polynomial expansion order and the weather features. The SVC without polynomial expansion and that uses all the weather features has 89.51% accuracy.

Figure 17: The graph shows the confusion matrix obtained in the test of the SVC without polynomial expansion, and using all the features.

were previously selected as the most suitable for the training due to containing artifacts (trees, structures, buildings, light posts, water marks in the window, etc...).

Histograms instead of the statistics from the pixels’ temperature and magnitude was found to under-perform and consequently discarded from further experiments. The weather station measures that were found to be uncorrelated with the classification are: air temperature, dew point, relative humidity, the Sun’s elevation and azimuth.

The accuracy of the classification model implies that there will be miss-classifications, but as transition between type of sky conditions is slow, we propose to implement a persistent classification so sporadic errors due to artifacts can be avoided,

\[
\hat{y}_k = [\hat{y}_k, \ldots, \hat{y}_{k-T}],
\]

where \( T \) is the lag in the persistent time series of consecutive classifications. We classify frame \( k \) such as,

\[
\hat{y}_k' = \text{Mode} (\hat{y}_k).
\]

Notice that the persistent classification vector of classes does not contain the modes, is updated with the output of the model classification.

7.2 Persistent Window Model

We have the set \( I \) that contains up to \( L \) selected frames. These frames had previously applied the atmospheric model so that is only remaining the reflected radiation from the clouds, and other particles, a part from other objects that can be
on the images and that were not accounted for,
\[ I_k = \{ \bar{I}_1, \ldots, \bar{I}_L \}, \quad \bar{I}_k \in \mathbb{R}^{D \times N}. \]  
(88)

The objective is that as the tracking system rotates the particles accumulated on the surface of the window stay constant over short periods of time, and consequently is possible to model them.

In addition, we have a vector \( i_k \) that contains the sequence of last \( \ell \) measures of CSI,
\[ i_k = [i_{k-\ell}, \ldots, i_k], \quad i_k \in \mathbb{R}^+. \]  
(89)

When the Sun is totally or partially occluded, the absolute value of the drop in CSI \( i_k \) is expected to be high,
\[ r_k = |1 - \frac{1}{\ell} \sum_{i=0}^{\ell} i_{k-i}|, \quad r_k \in \mathbb{R}^+. \]  
(90)

The total amount of CSI \( r_k \) is another indicator of when there was not stratus, nimbus, or any other type of large cloud. The aim is to add robustness to atmospheric conditions classification, so that a new frame is added to the set iff the classification is clear sky and the drop on the CSI was under a threshold,
\[ I_k = \begin{cases} I_k \cup \bar{I}_k & C_k = 0 \land r_k \leq 0.05 \\ I_k \cap \bar{I}_k & \text{Otherwise}, \end{cases} \]  
(91)

so that \( C_k = 0 \) and the robustness condition \( r_k \leq 0.05 \) are always fulfill. The window’s artifacts model is updated when there are enough frames \( L \) with clear sky conditions to attenuate the variance of the noise,
\[ \omega_{i,j} = \text{Median} (I_k), \quad \text{iff} \ |I_k| \geq L, \]  
(92)

where \(|\cdot|\) the cardinally of a set. When the DAQ session is finished for the day, an algorithm randomly selects \( L \) elements in the set form sampling an uniform distribution to initialize the set that will be used in the next day.

Figure 18: The graphs show the effects of window produced in the radiation measure by the IR sensor. Although these effects vary along the day, the changes are most visually appreciable after heavy rain. This algorithm requires \( L \approx 250 \) selected frames to learn the artifacts on the window. This task can be performed either during the night or day time.

### 7.3 Window Model Application

In order to avoid any window artifacts that can affect computation of the clouds motion vector, and the quality of the segmentation, the window model is applied to image after the atmospheric model subtraction,
\[ \bar{I}_k = I_k - \omega_{i,j}, \quad \forall i = 1, \ldots, M, \quad \forall j = 1, \ldots, N, \quad \bar{I}_k \in \mathbb{N}^{[1,9700]}. \]  
(93)

The resulting image is the reflect radiation from the clouds.

The fact of having the cloud’s intensity in the same interval facilitates the segmentation, and thereby the clouds labelling. To normalize an infrared image of 16 bits to 8 bits is necessary to consider that the average distance from the sea level to the Tropopause is \( \sim 12 \text{ km} \) at \( 36^\circ \) latitude north\footnote{https://www2.acom.ucar.edu/news/cloud-tops-and-tropopause} but Albuquerque is at \( 1,641 \text{ km} \) above the sea level. Then, if the air temperature decreases by \( 9.8^\circ \text{ km} \), the maximum feasible intensity that a cloud can have, should be at around: \( 9.8 \cdot (11,5 - 1,6) \cdot 100 = 9.7 \times 10^3 \). The interval where the clouds are recognizable in the images \( \bar{I}_k \) is,
\[ \bar{I}_k = \frac{\bar{I}_k - \min \bar{I}_k}{9.7 \times 10^3} \cdot 2^8, \quad \bar{I}_k \in \mathbb{N}^{[1,2^8]}. \]  
(94)

The minimum intensity \( \min \bar{I}_k \) can vary between consecutive images, and it can have a value either above zero (this is most in cloudy condition) or below zero. It is a offset or a floating point in the signal cause by the noise introduced in errors of the atmospheric and window models. Nevertheless, as the normalization is only for segmentation purposes, not for feature extraction, this is not an issue.
8 Perspective Transformation

The images are taken at an angle from the normal position of the camera in relation to the ground. This angle is the Sun’s zenith angle. This causes, that the relative distance of a object in horizon, to increase from the top to the bottom of an image. In order to account for this effect, we propose a geometrical transformation from the original euclidean frame of reference, to a non-linear frame of reference.

8.1 Rectilinear Lens

The image captured by an IR camera is a refraction of the black body radiation emitted by objects in a converging point. We consider that the radiant objects, which are the Sun and the clouds, are at an infinity distance from the camera’s germanium lens, so the radiation rays converge at the focal length. This can be shown by the thin lens equation, which is

\[
\frac{1}{f} = \frac{1}{z} + \frac{1}{d_i},
\]

where \( f \) is the focal length, \( z \) is the distance from the lens to the clouds, and \( d_i \) is the distance from the lens to converging point. If the equation factor is \( 0 \approx \frac{1}{z} \), then the rays converge at the focal length \( d_i = f \).

The relation between the diagonal Field of View (FOV), and the focal length \( f \) for a standard rectilinear lens is,

\[
\tan \frac{\text{FOV}}{2} = \frac{N_{\text{diag}}}{2f}
\]

where \( N_{\text{diag}} \) is the number of pixels in the diagonal of the sensor. As we know from the camera’s manufacture: the diagonal FOV is 63.75°, the horizontal FOV \( x \) is 51°, and the size of a pixel is \( \delta = 17\,\mu\text{m} \). Therefore, the focal length of our camera is,

\[
f = \sqrt{(N\delta)^2 + (M\delta)^2} / 2 \tan \frac{\text{FOV}}{2}
\]

that is a camera with resolution of \( N \times M \) pixels, and thus has a vertical FOV \( y \) of 38.25°.

8.2 Flat Earth Approximation

An approximate of perspective transformation can be easily found with the assumption that the Earth’s surface is flat. This consideration is practicable without large error due to small proportion recorded of the Earth’s atmosphere segment. In this approximation, the perspective transformation from the sensor plane to the camera plane, as described in figure [20] is a function of a cloud layer height \( h \) and the Sun’s elevation angle \( \varepsilon \),

\[
d_i = \frac{h}{\sin \varepsilon_i}.
\]

Applying the rectilinear lens approximation, the distance of the atmosphere projected into a pixel is \( x' = x \cdot \frac{\delta f}{f} \cdot d_i \). Therefore, knowing the pixels’ size and the focal length of the camera, the set of coordinates in the camera plane are given by the following formulas,

\[
x'_{i,j} = x_{i,j} \cdot \frac{\delta}{f} \cdot d_i = x_{i,j} \cdot \frac{\delta}{f \cdot \sin \varepsilon_i} \cdot h
\]
Figure 20: Schematic of the geometric transformation assuming that the recorded portion of the atmosphere is small enough to be viable to approximate it as flat. The upper plot shows the transformation in the FOV y-axis relative to the distance from the camera to a cloud at a height \( h \). The bottom plot shows the x-axis and y-axis differential increments calculated by the proposed transformation w.r.t. the camera plane.

and,

\[
y_{i,j}^{\prime} = y_{i,j} \cdot \frac{\delta}{f \cdot \sin \varepsilon_i} \cdot h
\]  

(100)

In order to transform the coordinates of the frame of reference from the camera plane to the cross-section plane, imagine a virtual pyramid delineated by the FOV of the camera when it intercepts a cloud’s layer. The pyramid defines a triangle when it is projected into a 2D plane, see upper schematic in figure 20, so we propose to compute the distance \( d_i \) for each row of pixels in a frame. Knowing that the FOV of a pixel is \( \Delta \alpha_y = \frac{\text{FOV}_y}{M} \) in our camera, and that these are the same for the x-axis and the y-axis, \( \Delta \alpha_y = \Delta \alpha_x = \Delta \alpha \). The vector of angular differential increments for each row of pixels is,

\[
\alpha = \left\{ \frac{i \cdot \Delta \alpha}{2} \middle| \alpha_i \in \mathbb{R}^{(0, 2\pi)} \right\}, \quad \forall i = 1, \ldots, M
\]

(101)

and when it is calculated w.r.t the Sun’s elevation angle \( \varepsilon_0 \), the elevation of each row of pixels in a frame is \( \varepsilon_i = \varepsilon_0 + (\alpha_i - \alpha_0) \).

The origin of coordinates is defined as the current position of the Sun for each vector in the x-axis,

\[
x_i'' = x_i'' - x_i'_{x_0}, \quad \forall i = 1, \ldots, M
\]

(102)
and the y-axis is,

$$Y'' = Y'' - y''_{x_0, y_0},$$  \hspace{1cm} (103)$$

where $x_0$ and $y_0$ are the Sun’s position index in the new coordinate system. Notice that the transformation of the camera to the cross-section plane are a linear function of the height.

![Figure 21](image)

**Figure 21:** These graphs show the transformation, when assuming that the surface is flat, of the original pixels’ Euclidian coordinates system decreasing with Elevation angle clockwise. The points in orange color are the pixels within the cross-section length when the camera is completely normal to the ground, which is at $\varepsilon = \frac{\pi}{2}$ rad. The tropopause is considered static at 12.5 km in all the graphs.

### 8.3 Great Circle Approach

The clouds phenomena is restricted to the troposphere, so the cloud bases can be found at any height from sea level, fog or mist, to the tropopause, alto-stratos or cirrus [13]. The height from the ground to the tropopause varies along the year, and it depends in the latitude. However, due to the fact that the cloud bases are found in a very small range compared to total atmosphere vertical distance from the ground up to the exosphere, we assume that the layers of clouds, are at constant height $h$ from the Earth surface, and we also assume that their surface is flat, for simplicity in finding the transformation that gives us the relative distance from a cloud to the camera on a frame. Therefore, the transformation from the sensor plane to the camera plane, as described in figure 22, is a function of a cloud layer height and the Sun’s elevation angle.

In order to transform the coordinates system from the camera plane to the cross-section plane, we have to compute the effects in each pixel due to the prospective in the image in differential increments as function of the distance. For that, we know that each pixels’ field of view is $\Delta \alpha_y = \frac{\text{FOV}_y}{M}$, and as the lens in our camera camera is a rectilinear, a pixel’s field of view is the same either in the x-axis, or in the y-axis, so that $\Delta \alpha_y = \Delta \alpha_x = \Delta \alpha$.

The pyramid formed by the camera’s FOV, defines symmetric triangles from the center of the image $x_0 = (x_0, y_0)$ to the sides of the plane when intercepts with a layer of clouds that is at a certain height $h$, see figure 22. Taking advantage of this fact, the actual cross-section plane can be estimated w.r.t. the camera plane, which is the projection of that plane in a frame. The differential increments in the y-axis of the camera plane are derived first, the differential increments in the x-axis are directly developed out from them. Notice that the x-axis in the Earth’s cross-section plane is the y-axis in the camera plane, and the x-axis in the camera plane is normal to Earth’s the cross-section plane, and it is obtained from the $z_i$ distance, see figure 22 and 23.

The great circle is the tropopause, and the small circle is the Earth. The tangent of Earth’s surface that intercepts the troposphere, is the chord defined as $AB$, see figure 25. The radius from center of the Earth to the tropopause, and the
Figure 22: Diagrams of the geometrical transformation in the y-axis and the x-axis respectively. Sensor and Lens are parts of the IR camera, and are physically separated by a focal length $f$. The velocity decomposition shows that the components of a cloud velocity has a perspective distortion either in the x-axis as in the y-axis due to the camera plane inclination of $\varepsilon$ degrees with respect to the normal. The distance $z$ from the camera to a layer of clouds at height $h$ is depicted in the upper graph. The FOV$_x$ and FOV$_y$ are $\alpha_x$ and $\alpha_y$. 
Figure 23: *Great* circle approach to find the distance $y$ between the chord of the small circle and small circle. The *small* circle radius is $r = r_{\text{earth}} + \rho$, where $\rho$ is the altitude of the localization. The *great* circle radius is $R = r + h$, where $h$ is the height of Troposphere, and it varies with respect to the latitude of the localization.

sagitta, which is the length from chord $CD$ to the troposphere top, are,

$$R = r + h$$

(104)

$$\ell_i = h - y_i.$$  

(105)

The relation between the sides of the triangle formed by the line that connects the Earth’s surface with the top of the tropopause with an inclination angle w.r.t. the normal $\frac{\pi}{2} - \varepsilon_i$ is,

$$\tan \varepsilon_i = \frac{y_i}{x_i}$$

(106)

$$x_i \cdot \tan \varepsilon_i = y_i$$

(107)

The sagitta of the chord $CD$ is related to the chord $AB$. The formula that describes the sagitta’s length of the chord $CD$, is a function of the connecting triangle formed by the intercepting line from $AB$ to $CD$, when the vertex angle $\varepsilon_i$ varies,

$$\ell_i = R - \sqrt{R^2 - x_i^2}$$

(108)

$$h - y_i = R - \sqrt{R^2 - x_i^2}$$

(109)

$$R^2 - x_i^2 = (x_i \cdot \tan \varepsilon_i + r)^2$$

(110)

$$R^2 - x_i^2 = x_i^2 \cdot \tan^2 \varepsilon_i + r^2 + x_i \cdot r \cdot \tan \varepsilon_i$$

(111)

$$0 = x_i^2 \cdot (1 + \tan^2 \varepsilon_i) + x_i \cdot (r \cdot \tan \varepsilon_i) - h \cdot (1 + 2 \cdot r)$$

(112)

The relation is explained by the quadratic equation that has the following coefficients,

$$\beta_i(y) = \begin{cases} 
  a_i = 1 + \tan^2 \varepsilon_i \\
  b_i = r \cdot \tan \varepsilon_i \\
  c_i = -h \cdot (1 + 2 \cdot r) 
\end{cases}$$

(113)

The side $x_i$ of the triangle is the positive result obtained from solving the quadratic equation with this formula,

$$x_i = -\frac{b_i \pm \sqrt{b_i^2 - 4 \cdot a_i \cdot c_i}}{2 \cdot a_i}, \quad x_i \in \mathbb{R}^+$$

(114)

The side $x_i$ of the triangle is the $y$-axis of the cross-section plane. To account for the curvature of the Earth, the side $x_i$ of the triangle, which is flat, is projected to the segment of the great circle with this formula,

$$y_i' = \theta_i \cdot R = \frac{\kappa_i}{2} \cdot \sin^{-1} \left( \frac{2x_i}{\kappa_i} \right)$$

(115)

$$\kappa_i = \ell_i + \frac{x_i^2}{\ell_i}$$

(116)
This gives the length of the segment formed by the normal of the camera, with each one of the pixel’s elevation angle.

The distance of segment, from the normal of the camera to the cross-section plane, is subtracted from each pixel,

\[ y'_i = y'_i - y_1, \]  

(117)

to calculate the relative distance between the extremes of the cross-section plane.

The vector of differential increments that transforms the camera plane to the cross-section plane in the y-axis, is obtained via numerical differentiation,

\[ \Delta y'_i = \Delta y'_i - \Delta y'_{i-1} \]  

(118)

so that \( Y' = y' \cdot 1^\top \), and analogously \( \Delta Y' = \Delta y' \cdot 1^\top \), where \( 1 \) is a vector of ones with dimensions \( M \times 1 \).

The projection of the camera plane, in the x-axis, at each pixel’s distance from the camera in the y-axis direction is,

\[ z_i = \sqrt{x_i^2 + y_i^2} \]  

(119)

\[ \alpha_j = (N + 2 - 2j) \cdot \nu \]  

(120)

\[ \hat{x}_{i,j} = 2 \cdot z_i \cdot \tan \frac{\alpha_j}{2} \]  

(121)

where \( \nu = \frac{\text{FOV}}{\sqrt{N^2 + M^2}} \cdot \frac{\pi}{180} \) is radians per pixel ratio of the camera.

The relation between the length of an arc segment, and the chord, is found through the sagitta. The formula which describes the relation between these variables is,

\[ R^2 = \left( \frac{\hat{x}_{i,j}}{2} \right)^2 + (R - \lambda_{i,j})^2 \]  

(122)

\[ - \left( \frac{\hat{x}_{i,j}}{2} \right)^2 = \lambda_{i,j}^2 - 2R\lambda_{i,j} \]  

(123)

\[-\lambda_{i,j}^2 + 2R\lambda_{i,j} - \left( \frac{\hat{x}_{i,j}}{2} \right)^2 = 0. \]  

(124)

The derived quadratic equation for computing the sagitta, has the following set of coefficients,

\[ \beta^{(x)}_{i,j} = \begin{cases} 
  a_{i,j} = -1 \\
  b_{i,j} = 2 \cdot R \\
  c_{i,j} = - \left( \frac{\hat{x}_{i,j}}{2} \right)^2 
\end{cases} \]  

(125)

In the x-axis, the great circle’s sagitta of the arc is the negative result obtained solving the quadratic equation,

\[ \lambda_{i,j} = \frac{-b_{i,j} \pm \sqrt{b_{i,j}^2 - 4 \cdot a_{i,j} \cdot c_{i,j}}}{2 \cdot a_{i,j}}, \quad \lambda_{i,j} \in \mathbb{R}^- . \]  

(126)

The segment length, which is calculated with the found sagitta, is the actual length of the x-axis cross-selection,

\[ x'_{i,j} = \frac{\kappa_{i,j}}{2} \cdot \sin^{-1} \left( \frac{\hat{x}_{i,j}}{\kappa_{i,j}} \right) \]  

(127)

\[ \kappa_{i,j} = \lambda_{i,j} + \frac{\hat{x}_{i,j}^2}{4\lambda_{i,j}} \]  

(128)

To compute the vector of differential increments that transform the camera plane to the cross-section plane in the x-axis, the numerical finite differential formula is applied only to the x-axis,

\[ \Delta x'_{i,j} = \Delta x'_{i,j} - \Delta x'_{i,j-1}, \]  

(129)

and it is symmetric to the center of the frame, so that \( X' = X' \cdot (\Lambda \cdot X')^\top \), and analogously \( \Delta X' = \Delta X' \cdot (\Lambda \cdot \Delta X')^\top \), where \( \Lambda \) is an anti-identity matrix of dimensions \( M \times M \).
Figure 24: These graphs show the non-linear transformation of the original pixels’ Euclidian coordinates system decreasing with Elevation angle clockwise. The points in orange color are the pixels within the cross-section length when the camera is completely normal to the ground, which is at $\varepsilon = \frac{\pi}{2}$ rad. The tropopause is considered static at 12.5km in all the graphs.

8.3.1 Length of a Segment Approximation

The segment of a circle formula dictates, that given the same angle of the great and small circle’s arc, the length are directly proportional,

$$s = \theta \cdot R$$  \hspace{1cm} (130)

$$\frac{s_1}{R_1} = \frac{s_2}{R_2}$$  \hspace{1cm} (131)

so that, when $R >> h$, the grid can be approximated at different heights applying the segment formula,

$$X'_i \approx X'_2 \cdot \frac{h_1}{h_2}$$  \hspace{1cm} (132)

$$Y'_i \approx Y'_2 \cdot \frac{h_1}{h_2}$$  \hspace{1cm} (133)

The origin of the non-linear coordinates system is the current position of the Sun, which is $(x_0, y_0)$. The Sun’s coordinates are given in the pixel’s index position in this new coordinates system.

References

[1] G.M. Masters and Wiley online library. Renewable and Efficient Electric Power Systems. Wiley - IEEE Series. Wiley, 2004.

[2] Jorge Nocedal and Stephen J. Wright. Numerical Optimization. Springer, New York, NY, USA, second edition, 2006.

[3] Iyyanki V. Muralikrishna and Valli Manickam. Chapter fourteen - air pollution control technologies. In Environmental Management, pages 337 – 397. Butterworth-Heinemann, 2017.

[4] John MWallace and Peter V. Hobbs. 3 - atmospheric thermodynamics. In Atmospheric Science (Second Edition), pages 63 – 111. Academic Press, second edition, 2006.

[5] C.R. Mechoso and A. Arakawa. Numerical models | general circulation models. In Encyclopedia of Atmospheric Sciences (Second Edition), pages 153 – 160. Academic Press, second edition, 2015.
Figure 25: Projection of the tropopause’s segment into the Earth surface. The Geographic Coordinates System (GCS) is in degrees. The camera is localized at the ME Building in the UNM central campus. The New Mexico maps, that organized clockwise, are simulations of the frames projections tracking the Sun along a day.

[6] Z. Zhang and J.C. Moore. Mathematical and Physical Fundamentals of Climate Change. Elsevier Science, 2014.
[7] James R. Holton and Gregory J. Hakim. Chapter 2 - basic conservation laws. In An Introduction to Dynamic Meteorology (Fifth Edition), pages 31 – 66. Academic Press, fifth edition edition, 2013.
[8] D. Lamb and J. Verlinde. Physics and Chemistry of Clouds. Cambridge University Press, 2011.
[9] Andrey N. Tikhonov and Vasiliiy Y. Arsenin. Solutions of ill-posed problems. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.
[10] H. C. Andrews and C. L. Patterson. Digital interpolation of discrete images. IEEE Transactions on Computers, C-25(2):196–202, 2 1976.
[11] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. J. Mach. Learn. Res., 9:1871–1874, June 2008.
[12] Chih wei Hsu, Chih chung Chang, and Chih jen Lin. A practical guide to support vector classification, 2010.
[13] R.A. Houze. Cloud Dynamics. Cloud Dynamics. Academic Press, 1993.
Figure 26: This figure shows the error maps between the full-prospective transformations and the transformation with flat surface assumption for the previously displayed elevation angles. The total error maps were computed from the squared residuals of each coordinate $E = \sqrt{\frac{1}{2} \cdot [E(X_1, X_2) + E(Y_1, Y_2)]}$, where the squared residuals are $E(X_1, X_2) = (X_1' - X_2')^2$.