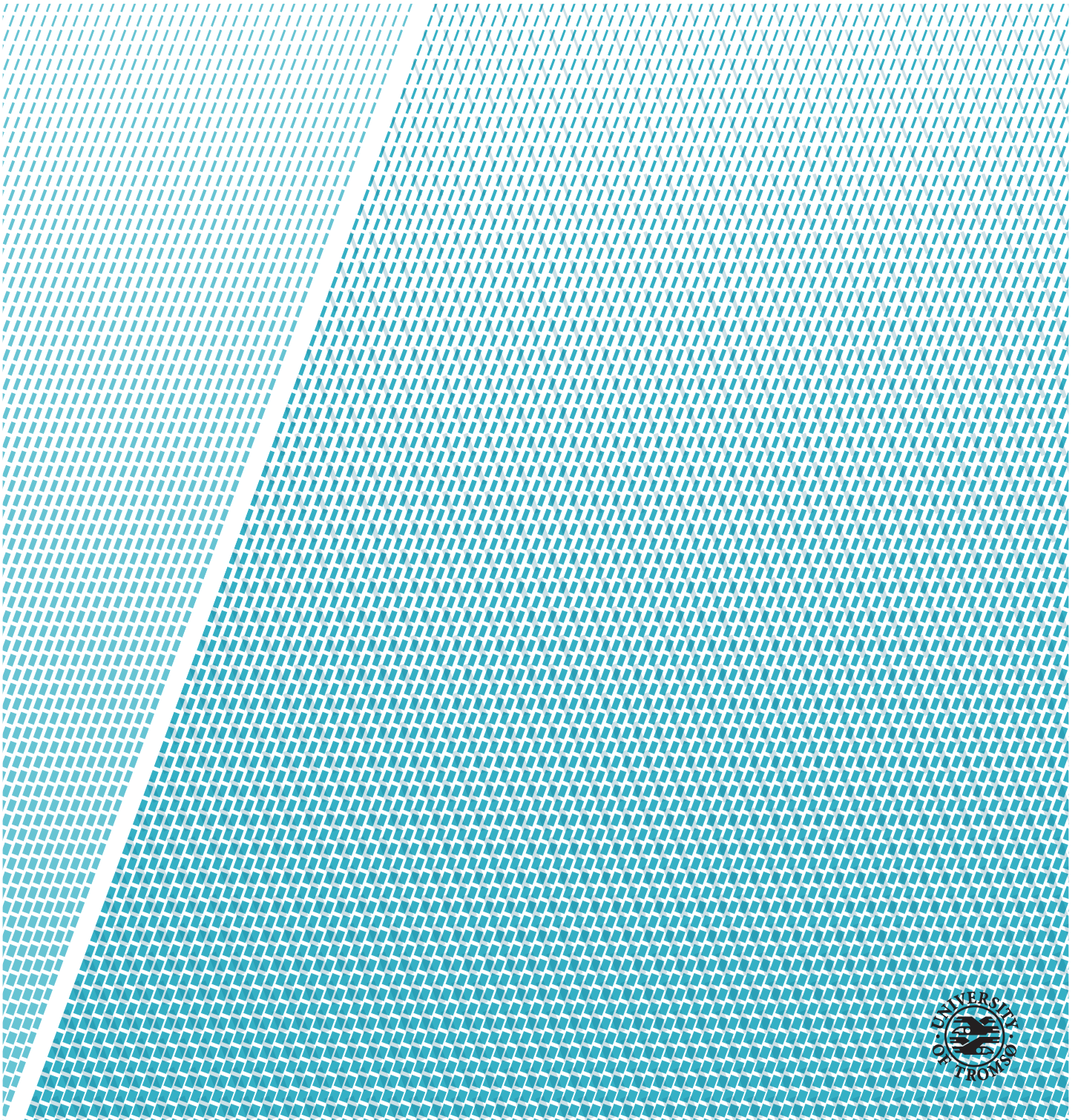


Planetary auroral imaging

Yoshiaki Nagoshi

FYS-3900 Master's thesis in Physics 60ECTS ... July 2021



Abstract

The interaction of the solar wind and the Earth magnetosphere cause auroras. The energetic electrically charged particles, mostly electrons, accelerate along the Earth magnetic field lines into the upper atmosphere, where they collide with gas atoms, causing them to emit light. Some planets also have auroral emission in their characteristic environment. The Jovian magnetosphere is the largest magnetosphere of the solar system and its system is different from the Earth. The auroras on Jupiter can be studied with high sensitivity and resolution by the Hubble Space Telescope (HST) Ultraviolet (UV) and far-ultraviolet Space Telescope Imaging Spectrograph (STIS) and Advanced Camera for Surveys (ACS) instruments. I present the planetary auroral imaging techniques, geometrical transformation and subtraction the airglow model, which can be used as a pre-processing to the image before further process by the VOronoi Image SEgmentation (VOISE) algorithm [6]. VOISE is a dynamic and self-organising algorithm which creates a partition of an image pixel into Voronoi diagram (VD) regions according to prescribed homogeneity criteria. The Jovian auroral image was selected from the APIS database [8]. Using a planetary model [7], the geometric transformation was performed to get the polar projection, build the airglow model and subtract it from the original to make a clear auroral representation in the two dimensional image.

Acknowledgements

I am grateful to Professor. Patrick Guio for helpful discussions and the Planetary model section is based on his note. I couldn't do this thesis work without his help.

Contents

Abstract	i
Acknowledgements	iii
List of Figures	vii
1 Introduction	1
1.1 Objective of this project	1
1.2 Jovian aurora and magnetosphere	2
1.2.1 Earth's magnetosphere and aurora	2
1.2.2 Jovian magnetosphere	5
1.3 Coordinate and latitude system	6
2 Planetary auroral imaging	7
2.1 APIS database (Auroral Planetary Imaging and Spectroscopy service)	7
2.2 Determination of the center of the planet	8
2.3 Right ascension and declination	9
2.4 Geometric transformation	10
2.4.1 Orthographic projection	10
2.4.2 Rotation of the image	11
2.4.3 Polar projection	13
2.5 Airglow model and subtract it from the image	15
2.5.1 Cosine illumination model	15
2.5.2 Minnaert model	17
2.5.3 Li model	19
3 Result and discussion	23
3.1 Result of airglow models	23
3.2 Discussion and conclusion	26
4 Appendix 1: Planetary model from Professor Patrick Guio's note	27
4.1 Planetary model	27
4.2 Limb projection	29

4.3	Terminator projection	33
4.4	Cusp points and terminator determinant	36
4.5	Find visible and hidden terminator	39
5	Appendix 2: Voronoi diagram and VOISE algorithm	41
5.1	Voronoi diagram	41
5.2	Delaunay triangulation	42
5.3	VOISE algorithm	43
5.4	Initialization phase	44
5.5	Dividing phase	44
5.6	Merging phase	46
5.7	Regularization phase	47
6	Appendix 3: Minnaert model Matlab code	49
7	Appendix 4: Li model Matlab code	53

List of Figures

- 1.1 Schematic illustration of the Earth's magnetosphere. The Earth's magnetic field lines are shown as modified by the interaction with the solar wind. The solar wind, whose flow speed exceeds the speeds at which perturbations of the field and the plasma flow directions can propagate in the plasma, is incident from the left. The pressure exerted by the Earth's magnetic field excludes the solar wind. The boundary of the magnetospheric cavity is called the magnetopause, its nose distance (black arrow) being R_m Sunward (upstream) of the magnetopause. The bow shock shows the incident flow, and the perturbed solar wind plasma between the bow shock and the magnetopause is called the magnetosheath. Antisunward (downstream) of the Earth, the magnetic field lines stretch out to form the magnetotail. In the northern portion of the magnetotail, field lines point generally sunward, while in the southern portion, the orientation reverses. These regions are referred to as the northern and southern lobes, and they are separated by a sheet of electrical current flowing generally dawn to dusk across the near-equatorial magnetotail in the plasmasheet. Low-energy plasma diffusing up from the ionosphere is found close to Earth in the region called plasmasphere whose boundary is the plasmapause. The dots show the entry of magnetosheath plasma that originated in the solar wind into the magnetosphere, particularly in the polar cusp regions. Inset is a diagram showing the three-dimensional structure of the Van Allen belts of energetic particles that are trapped in the magnetic field and drift around the Earth. Source: *The New Solar System* (eds. Kelly Seatty et al.), CUP/Sky Publishing. Credit: Steve Bartlett; Inset: Don Davis..[15] . . . 4

1.2	(Left) Illustration of the System III coordinate system. The z-axis is defined by the spin axis of Jupiter. The x-axis is defined by 0° latitude on the System III longitude $\lambda_{III} = 0^\circ$ which is the prime meridian of Jupiter. The y-axis completes the orthogonal left-handed system. Latitude β defined from the equator. (Right) Illustration of the planetocentric (Φ') and planetographic (Φ) latitudes.	6
2.1	The image of Jupiter taken by HST 2007-02-20 (Dataset: j9rlb0fxq_drz). This is the raw data which has no extensions and less information than the processed one. The scale of the image is in pixel.	8
2.2	The image of celestial sphere and right ascension and declination.	9
2.3	A vertical perspective from infinity with parallel rays. This is the sketch of the orthographic projection. The planetary model which has (x, y, z) -coordinate system is projected into the tangent plane (x_s, y_s) -coordinate. Orthographic image is the same image we see in the picture of the planet. The origin of the (x_s, y_s) -coordinate is always at the center of the ellipse (limb) of the planet. The center point is corresponding to the sub-Earth point of the planet to the Earth observer.	10
2.4	Rotation of the planetary image. The planetary model grid is rotated with the angle of $(\alpha - \alpha_{HST})$ where α is the position angle between celestial north (direction normal to the ecliptic) and the axis of the Jupiter. α_{HST} which is the orientation angle between celestial north and the V3 axis of HST camera.	11
2.5	The rotated image of Jupiter with grid lines (Dataset: j9rlb0fxq_drz). This is the same image of Fig 2.1 but plotted with the planetary model grids and lines of limb and terminator generated by the algorithm of Appendix 1. The center of the planet is checked with SPICE tool kit and IMCCE query service. Then the center point is chosen manually corrected one. The scale of the image is transformed pixel to arcsec.	12
2.6	Sketch of transformation of two dimensional planetary image into polar projection.	13
2.7	The image of Jupiter in polar projection (Dataset: j9rlb0fxq_drz) with the planetary model grid. The azimuth of 0° is the CML line. The image scale is transformed arcsec to the scale of Jovian radius R_J . The intensity of the image converted into auroral brightnesses (1 kR represents a photon source flux of $10^9 \text{cm}^{-2} \text{s}^{-1}$ radiating into 4π steradians) using the the conversion factor $1 \text{ kR} = 1.473 \times 10^{-3} \text{counts s}^{-1} \text{ pixel}$ [11] of total H_2 emission over the 80-170 nm spectral range.	14

2.8	The illustration of the observer zenith angle θ and the cosine.	16
2.9	S scatter plot of $\ln(I\mu)$ as a function of $\ln(\mu\mu_0)$ with 1 degree resolution. The asymmetry of the sunlit side and the terminator side is clearly seen in the latitude bin close to the pole and the equator. The asymmetry in the polar latitude bin is considered to be caused by the auroral emission.	18
2.10	There is a difference in day and night side mean intensity function of MLAT. The range between the red vertical lines is determined as the auroral oval region following criteria. The outside of the red vertical lines is the non-auroral region. . .	20
2.11	Peak auroral region intensity in MLAT range of -87° to -66° . There are seen some peaks of auroral emissions.	20
2.12	Non auroral region pixel intensity vs cosine of Solar Zenith Angle (SZA) and Observer Zenith Angle (OZA).	21
3.1	The polar projection of subtracted image. The image scale is the Jovian radius R_J . The azimuth of 0° is the CML line. The intensity of the image is converted into auroral brightnesses (1 kR represents a photon source flux of $10^9 \text{cm}^{-2} \text{s}^{-1}$ radiating into 4π steradians) using the the conversion factor $1 \text{ kR} = 1.473 \times 10^{-3} \text{counts s}^{-1} \text{pixel}$ [11] for total H_2 emission over the 80-170 nm spectral range.	23
3.2	(Top) Original image (Dataset: j9rlb0fxq_drz). The intensity unit is the numbers of photons count per Pixel (Middle) calculated Minnaert model from the original image with the Minnaert model with 1 degree latitude bins, (Bottom) The subtracted image. The scale of the image is in pixel.	24
3.3	(Top) Original image (Dataset: j9rlb0fxq_drz). The intensity unit is the numbers of photons count per Pixel (Middle) calculated Li model from the original image with the Minnaert model with 1 degree latitude bins, (Bottom) The subtracted image. The scale of the image is in pixel.	25

4.1 (From [7]) Sketch of the geometry of the planet and the observer. The eccentricity of the planetary ellipsoid is exaggerated for clarity. The figure is a cut through the planet that contains the planet's rotation axis z and the observer direction δ_{obs} . The x -axis is contained in the equatorial plane of the planet and in the plane of the figure. The direction \hat{n}_s is the local normal of the ellipse perpendicular to the observing direction δ_{obs} . The plane of the limb is contained in a plane perpendicular to the figure and its intersection with the plane of the figure is the dashed line segment. \hat{n}_L is the direction normal to the plane of the limb and pointing toward the observer. The angle between \hat{n}_L and the x -axis is $\beta_{\hat{n}_L}$. Finally (x_s, y_s) are the axes for the plane of the sky as seen by the observer. 28

5.1 An example of Voronoi diagram. Dots are seeds, dashed lines and solid lines are bisector between associate seeds. Each region with a point separated by the lines is VR. VD appears with a form of dashed and solid lines. 42

5.2 An example of the appearance of Voronoi graph and the Delaunay triangulations. 43

5.3 (from [6]) The image of how the weights effect on seed placement. Seeds are added to the central hexagonal region (upper-left panel) recursively for three iterations, adding 18 seeds. The weights are $w_s = \frac{1}{4}$, $w_v = \frac{3}{4}$ (upper-right), $w_s = \frac{1}{2}$, $w_v = \frac{1}{2}$ (lower-left), $w_s = \frac{3}{4}$, $w_v = \frac{1}{4}$ (lower-right), 45



Introduction

1.1 Objective of this project

The goal of this project is to develop a planetary auroral image processing tool around the Voronoi Image SEgmentation (VOISE) algorithm [6]. The application of VOISE has been presented in some astronomical imaging. One is the semi-automatic planetary disc parameters detection which contains the limb and terminator equations and fitting algorithm[4]. Also the method for auroral feature detection by combined image segmentation and clustering analysis[3]. For this project, our interest is to understand the planetary auroral physics from images. The images are often captured by both ground- and space-based instruments, and have been studied. Planetary auroral images have become a particularly useful diagnostic tool for morphological characterizations of the planetary aurora and its magnetosphere. During the last two decades, the Hubble Space Telescope (HST) has provided high sensitivity and high resolution images of Jupiter and its moons, Saturn and Uranus in the ultraviolet (UV) spectral region space observation required. The images capture spectacular auroral phenomena occurring in the polar region of the gas giants as a result of strong magnetospheric particles raining down onto the planet's upper atmosphere. At Jupiter and Saturn, the dominant atmospheric species is hydrogen, which emits in the UV when excited by auroral electron impact. Combining remote imaging with in situ data allows the study of magnetospheric processes and how they affect the planet's upper atmosphere, and ionosphere via the planet's magnetic field, and the footprint auroral emission of satellites. However, before getting into the more interpretative physics of the planetary aurora, the techniques of the image processing have to be defined. For example, before to any detailed investigation, we need to get a

set of parameters for these images related to each particular observation. In addition, geometrical transformation technique is required so that we can understand the polar region aurora from a 2 dimensional image. Moreover, the faint or noisy features of the image need to be removed, for further processing by VOISE. For this project, we chose the Jovian auroral images. The important steps for planetary auroral imaging are introduced in the second chapter of Planetary auroral imaging.

1. Introduce the data source of images with basic information and tools for identifying the astronomical parameters to each data set. The motivation for this is to know the exact geometry of the planetary aurora. This step is in sections 2.1 to 2.3.
2. Geometric transformation to 2 dimensional image by using planetary model provided by Professor Patrick Guio. Because, we want to look down and study the polar region aurora, polar projection to the 2 dimensional image is required. This step is in section 2.4.
3. Program two airglow model to isolate the intrinsic auroral emissions. This is a pre-processing method before further image processing by VOISE. This step is in section 2.5.

1.2 Jovian aurora and magnetosphere

1.2.1 Earth's magnetosphere and aurora

The term magnetosphere was introduced by T. Gold in 1959 to describe the region above the ionosphere in which the magnetic field of the Earth controls the motions of charged particles. The magnetic field traps low-energy charged particles and forms the Van Allen belts, torus-shaped regions in which high-energy ions and electrons, tens of keV and higher, drift around the Earth. The control of charged particles by the planetary magnetic field extends many Earth radii into space but terminates near 10 Earth radii in the direction toward the Sun. At this distance, the magnetosphere is confined by a low density magnetized plasma called the solar wind that flows radially outward from the Sun at supersonic speeds. Qualitatively, a planetary magnetosphere is the volume of space from which the solar wind is excluded by a planet's magnetic field. A schematic illustration of the terrestrial magnetosphere is given in Fig 1.1, which shows how the solar wind is diverted around the magnetopause, a surface that surrounds the volume containing the Earth, its distorted magnetic field, and the plasma trapped within the field[15]. This qualitative definition is far from precise. Some solar wind plasma finds its way in and dynamical phenomena give clear evidence of intermittent direct links between the solar wind and the plasmas governed by a planet's magnetic field. Auroras are the result of disturbances in the

magnetosphere caused by the solar wind. These disturbances alter the trajectories of charged particles in the magnetospheric plasma. These particles, mainly electrons and protons, precipitate into the upper atmosphere then they collide with gas atoms, causing the atoms to give off light.

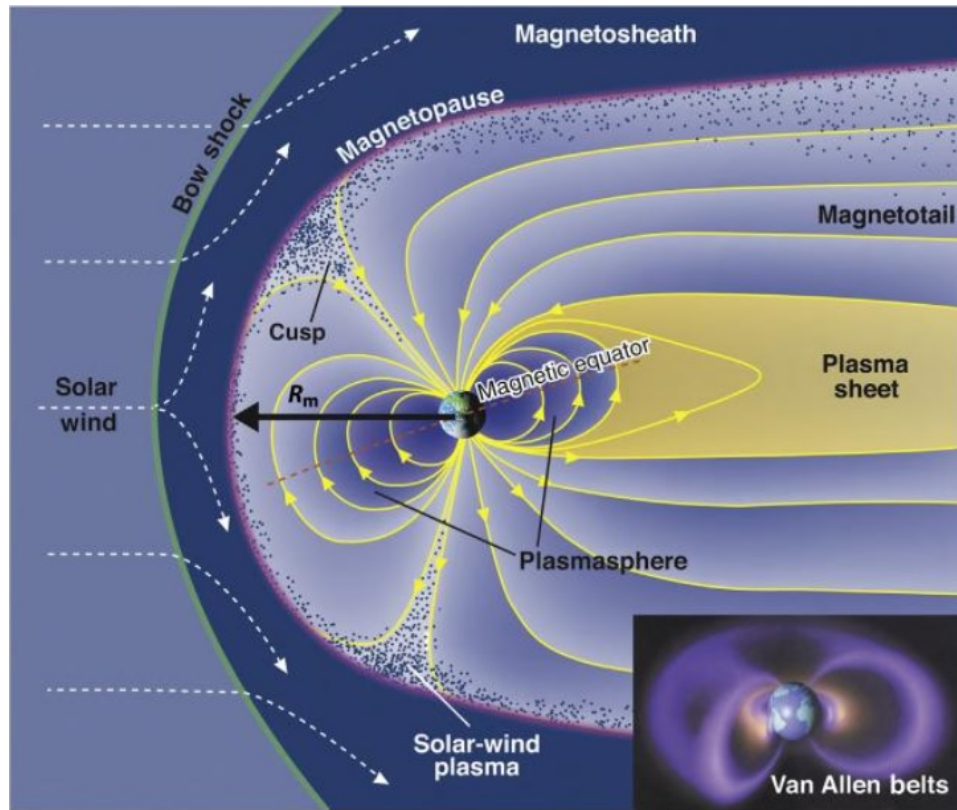


Figure 1.1: Schematic illustration of the Earth's magnetosphere. The Earth's magnetic field lines are shown as modified by the interaction with the solar wind. The solar wind, whose flow speed exceeds the speeds at which perturbations of the field and the plasma flow directions can propagate in the plasma, is incident from the left. The pressure exerted by the Earth's magnetic field excludes the solar wind. The boundary of the magnetospheric cavity is called the magnetopause, its nose distance (black arrow) being R_m Sunward (upstream) of the magnetopause. The bow shock shows the incident flow, and the perturbed solar wind plasma between the bow shock and the magnetopause is called the magnetosheath. Antisunward (downstream) of the Earth, the magnetic field lines stretch out to form the magnetotail. In the northern portion of the magnetotail, field lines point generally sunward, while in the southern portion, the orientation reverses. These regions are referred to as the northern and southern lobes, and they are separated by a sheet of electrical current flowing generally dawn to dusk across the near-equatorial magnetotail in the plasma sheet. Low-energy plasma diffusing up from the ionosphere is found close to Earth in the region called plasmasphere whose boundary is the plasmapause. The dots show the entry of magnetosheath plasma that originated in the solar wind into the magnetosphere, particularly in the polar cusp regions. Inset is a diagram showing the three-dimensional structure of the Van Allen belts of energetic particles that are trapped in the magnetic field and drift around the Earth. Source: *The New Solar System* (eds. Kelly Seatty et al.), CUP/Sky Publishing. Credit: Steve Bartlett; Inset: Don Davis.[15]

1.2.2 Jovian magnetosphere

The Jovian magnetosphere is the largest magnetosphere of the solar system. The distance from the planet center of the dayside magnetopause can reach to 100 Jovian radii (R_J), while the limit only reach to 10 Earth radii for Earth. Jupiter's internal magnetic dipole axis is tilted by 9.6° from the rotation axis of the planet and its rotation period is 9h55m. The Jovian magnetosphere structure is usually divided into 3 regions: the inner magnetosphere expands from 5 to 10 R_J , the middle magnetosphere spans from 10 to 40 R_J , and the outer magnetosphere ranges from 40 R_J to magnetopause. The Jovian magnetosphere differs from the Earth magnetosphere in two main aspects, one is the plasma source and another is the energy source. The Earth magnetosphere is a solar wind-driven magnetosphere in which both the plasma and the energy mainly originate from the solar wind. Jupiter's magnetosphere is an internally-driven magnetosphere and the available energy is mainly provided by the rapid rotation of the planet's strong magnetic field. Additionally, the magnetospheric plasma essentially originates from an internal source, Jovian moon Io's volcanism. There are four particularly big moons which are called Galilean satellites, Io, Europa, Ganymede, and Callisto. The moon Io orbits around Jupiter from about 6 R_J and Io is exposed to large tidal forces and heat. The interior of Io liberates from its volcanoes of the order of 1 tonne s^{-1} of sulphur and oxygen plasma into the neighbourhood of its orbit. The inner magnetosphere (5 to 10 R_J) is mainly controlled by the Jovian internal magnetic field. The main features are the inner radiation belts and the Io's plasma torus. The plasma feels the fast rotation of Jupiter's magnetic field and is accelerated to the same rotation rate as the planet. This fast rotation causes it to diffuse away from Jupiter due to the centrifugal force. The migrating plasma from the inner magnetosphere is slowed down and expands creating a vast spinning middle and outer magnetosphere of around Jupiter. The strong magnetic field keeps all this plasma rotating at the same angular velocity. This enforcement is moderated by an electric current system and a part of this system drives the electrons to upper-atmosphere and create the main auroral ovals. Jupiter exhibits two more other classes of auroral features which are the moon footprint and the polar auroras. The moon footprints are spots of auroral emission that are magnetically linked to the Galilean moons. The brightest spot is that associated with Io, and it also exhibits a tail, or wake that traces in the direction of rotation of Jupiter.

1.3 Coordinate and latitude system

Jupiter is a gaseous planet, as far as the polar aurorae are concerned, the most relevant coordinate system is the System III. The illustration of System III is shown in Fig 1.2. Since the main auroral features are approximately fixed in this reference frame. This coordinate system is bound to the Jovian magnetic field. This coordinate system is left-handed, the longitudes increase from dusk to dawn through noon. The System III longitude at the intersection point of the Earth-Jupiter line with the Jovian surface is called the Central Meridian Longitude (CML).

Since the shape of Jupiter is an ellipse, with the equatorial radius is about 71492 km and the polar radius is about 66854 km. Two latitude systems co-exist, planetocentric and the planetographic latitudes. The planetocentric latitude (Φ') refers to the angle between the line joining the planet center to the considered point and the equatorial plane, while the planetographic latitude (Φ) refers to the angle between the normal to the surface at the considered point and the equatorial plane. These two angles have a relation with the semi-major axis a and semi-minor axis b .

$$\tan(\Phi') = \frac{b^2}{a^2} \tan(\Phi). \quad (1.1)$$

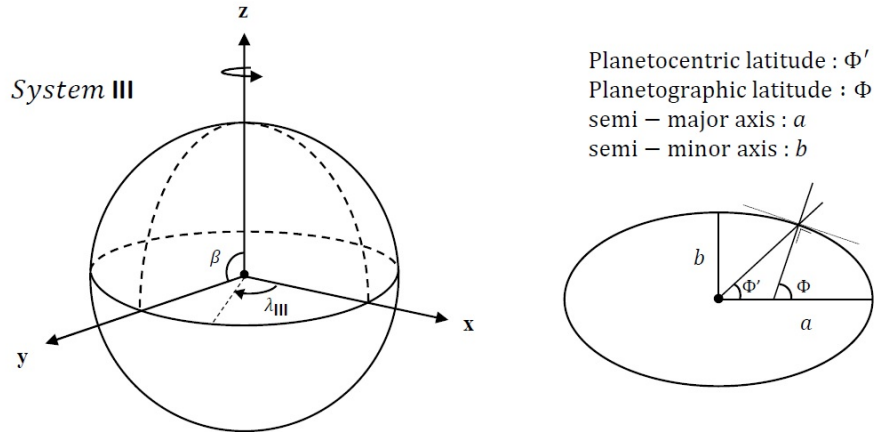


Figure 1.2: (Left) Illustration of the System III coordinate system. The z-axis is defined by the spin axis of Jupiter. The x-axis is defined by 0° latitude on the System III longitude $\lambda_{III} = 0^\circ$ which is the prime meridian of Jupiter. The y-axis completes the orthogonal left-handed system. Latitude β defined from the equator. (Right) Illustration of the planetocentric (Φ') and planetographic (Φ) latitudes.

/2

Planetary auroral imaging

2.1 APIS database (Auroral Planetary Imaging and Spectroscopy service)

Firstly, introduce the data source of images with basic information which are from the Auroral Planetary Imaging and Spectroscopy service (APIS)[8] database. The APIS service started in 2015. It provides an open access to processed auroral observations of the outer planets and their satellites. Their high resolution far ultraviolet (FUV) images are a wealth of information on planetary environments. The planetary image data has been provided by HST many campaigns before the APIS service began, the information of the image are not beginner friendly, however, because of their complexity. The APIS database allows beginners easy access to a high level processed data, built from public HST observations. For the FUV imaging, the instruments are restricted to the mostly used, such as Space Telescope Spectrograph (STIS)[16] and the Advanced Camera for Surveys (ACS). APIS provides a set of value-added data corresponding to the level of processing for each of images in *fits* [13] files. The images are provided under three levels of processing. The image data are chosen from the APIS database. The image processing tool needs to be built for both processed and raw auroral data. The processed data has abundant astronomical ephemeris information and a set of extensions. The extensions are necessary for limb darkening process, but the raw data does not have those data. To fill the lack of information, we need to get the raw image data close to the processed one. Using the planetary model in Appendix 1 provided by Professor Patrick Guio, with the accurate center of the planet we can get similar data equivalent to the processed

data.

Fig 2.1 is the image of Jupiter taken by ACS of HST on 20th of February, 2007 (Dataset: j9rlb0fxq_drz) from APIS. I used this dataset for further image processing in this project.

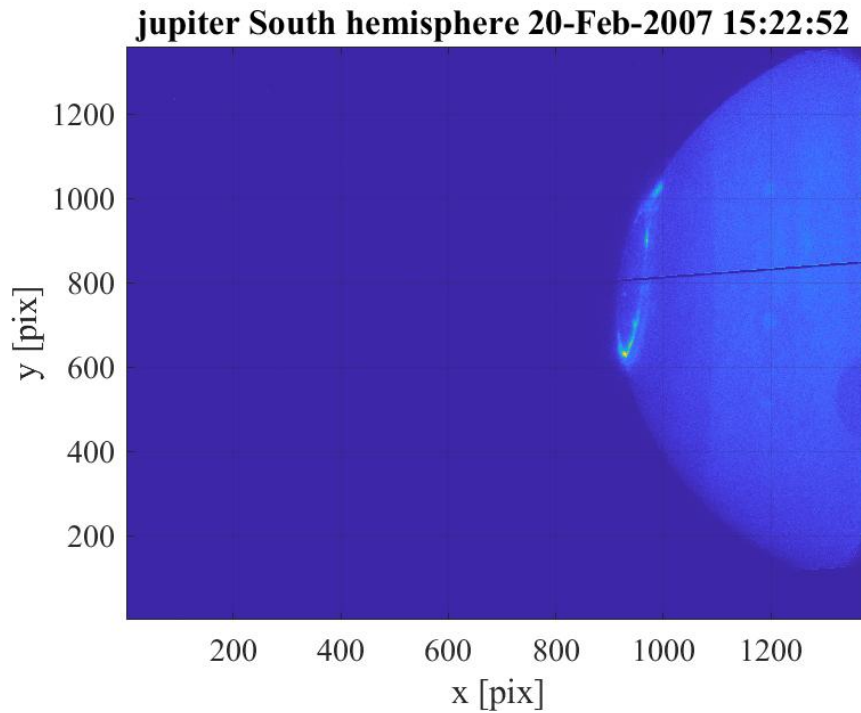


Figure 2.1: The image of Jupiter taken by HST 2007-02-20 (Dataset: j9rlb0fxq_drz). This is the raw data which has no extensions and less information than the processed one. The scale of the image is in pixel.

2.2 Determination of the center of the planet

The determination of the planetary center is a mandatory step before locating any structure in the image. The planet is modeled by the equations written in Appendix 1. The accurate center of the planet is important to get the projections of the image also to get the limb darkening model. The raw data from the APIS database has the basic information, such as the position of a particular structure. Unfortunately, HST pointing coordinates are not generally known with sufficient accuracy for this project. The precision of the star catalog along with the uncertainty in the start time of tracking motion is on the order of 1 arc sec while it is desired to have an accuracy of the order of 1 pixel, i.e. 0.02-0.03 arc sec for STIS and ACS instruments, in order to locate any structure accurately or to build polar projections of the planetary aurora[5].

The correction of the center point to the 2 dimensional image was checked with the metadata provided with the NASA Navigation and Ancillary Information Facility SPICE system [1] SPICE tool kit, internet query service (Institut de Mécanique Céleste et de Calcul des Ephémérides or IMCCE), and the manual way. In this project, I use the manually corrected center for projection and building the ariglow model.

2.3 Right ascension and declination

A standard way of specifying a location of the planet from the Earth is to use astronomical ephemeris, which refer the orbit to a *frame of reference* that is fixed relative to the stars. HST is actually orbiting about the Earth, I assumed the HST observation is almost same as from the Earth. The *frame of reference* that is commonly used can be defined in terms of x , y , and z axes. The z axis is along the Earth's rotation axis in north direction. It is at an angle of $23^{\circ}27'8''$ to the normal of the ecliptic plane. The y axis is a 'right-handed' orthogonal to the (x, z) -plane. The celestial sphere is the name given to a sphere with infinite radius centred on the Earth. The z axis meets it at a point known as the North celestial pole. The motion of the target planet P and the Sun on the celestial sphere is illustrated in Fig 2.2. The point P on the celestial sphere, the direction OP expressed in terms of the two angles α and β . The angle α between OX and the equatorial radius is called the right ascension of P . Its declination β is the angle between OP and the equatorial plane.

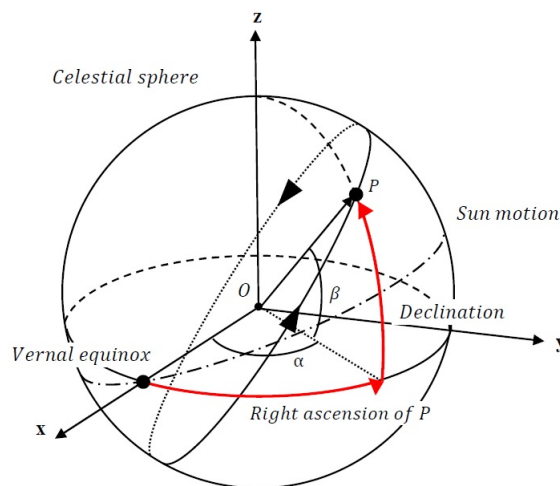


Figure 2.2: The image of celestial sphere and right ascension and declination.

2.4 Geometric transformation

Geometric transformation is an essential image processing technique that has wide applications. It could also be applied to project an image to another image plane.

2.4.1 Orthographic projection

Orthographic projection in cartography is a perspective projection. Fig 2.3 shows a sketch of orthographic projection. The orthographic perspective is the view from space, a vertical perspective from infinity with parallel rays. In which the sphere (ellipsoid) is projected onto a tangent plane of its surface point. Any point in the planetary surface can be projected into *the sky plane* and seen like as 3D image. As a map projection, the orthographic has a feature which has negligible distortion near the origin of the projection (*the sky plane*) but the areas of near the edges are distorted. Orthographic projection depicts a hemisphere, where the horizon is a great circle. The planetary model in Appendix 1 is the core algorithm of this projection method.

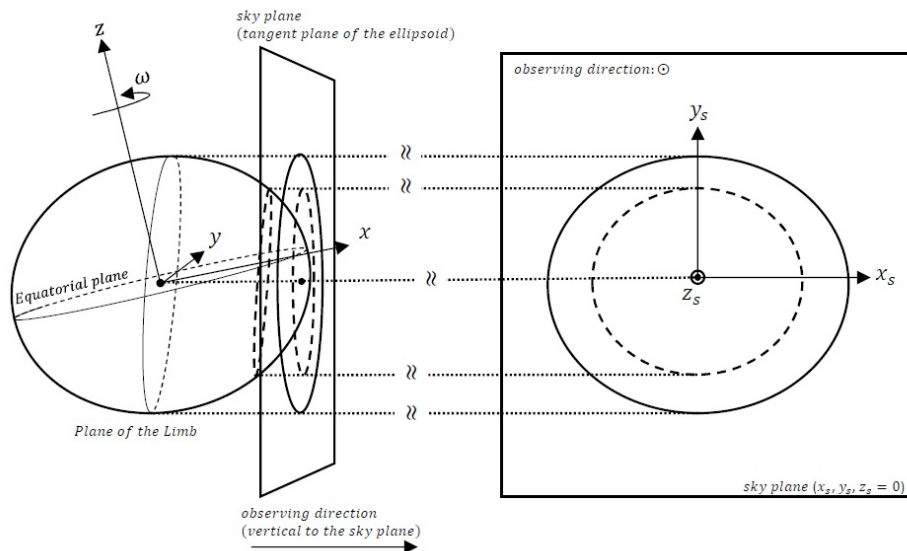


Figure 2.3: A vertical perspective from infinity with parallel rays. This is the sketch of the orthographic projection. The planetary model which has (x, y, z) -coordinate system is projected into the tangent plane (x_s, y_s) -coordinate. Orthographic image is the same image we see in the picture of the planet. The origin of the (x_s, y_s) -coordinate is always at the center of the ellipse (limb) of the planet. The center point is corresponding to the sub-Earth point of the planet to the Earth observer.

2.4.2 Rotation of the image

To apply the planetary model of Appendix 1 to the planetary image, it is required to rotate its rotation axis to align with the rotation axis of the planet in the image. Here we introduce the HST axes of reference which is called $(V1, V2, V3)$ -coordinate[9], where $V1$ is the axis pointing to the target body along the axis of optics. $V3$ represents the direction of the ordinates of the planetary image plane. ($V2$ completes the abscissa of the image plane. We can write the relation between $(V1, V2, V3)$ -coordinate and the (x, y, z) -coordinate for the planetary model built as follows

$$(x, y, z) = (-V1, -V2, V3) \quad (2.1)$$

Using the SPICE toolkit, We can get the angle α which is the position angle between celestial north (direction normal to the ecliptic) and the axis of Jupiter. Moreover, we can get the angle of α_{HST} which is the orientation angle between celestial north and the $V3$ axis from the data set. So we can know the rotation angle $(\alpha - \alpha_{HST})$ to align the planetary model rotation axis to along the y_s axis of the image plane through the system of the celestial sphere. Fig 2.4 shows the illustration of the rotation and Fig 2.5 shows the rotated image of Jupiter with planetary model grid and some markers.

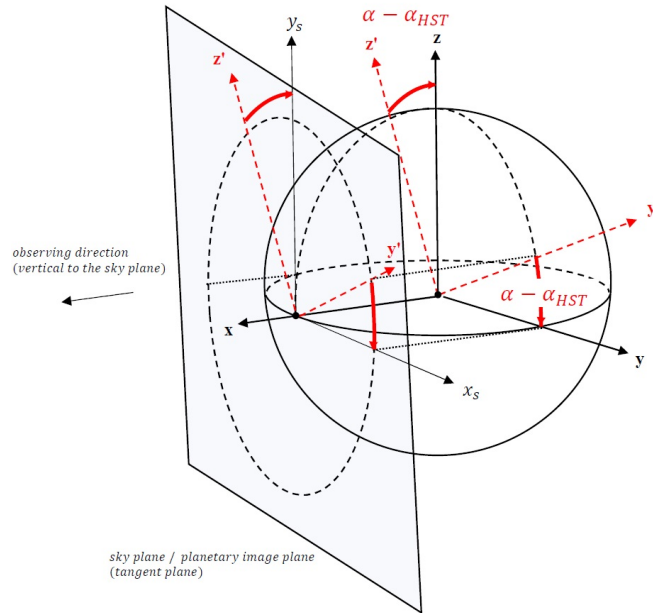


Figure 2.4: Rotation of the planetary image. The planetary model grid is rotated with the angle of $(\alpha - \alpha_{HST})$ where α is the position angle between celestial north (direction normal to the ecliptic) and the axis of the Jupiter. α_{HST} which is the orientation angle between celestial north and the $V3$ axis of HST camera.

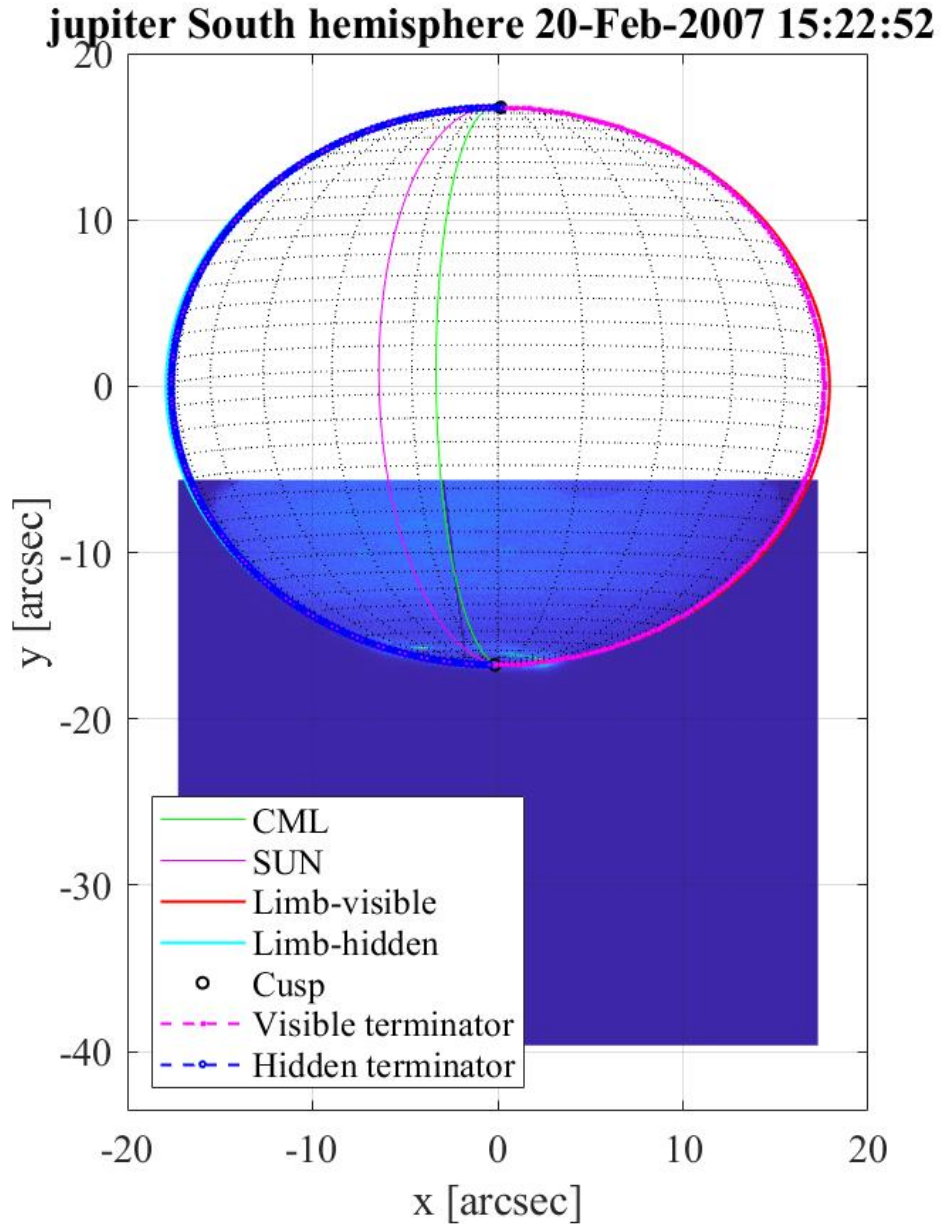


Figure 2.5: The rotated image of Jupiter with grid lines (Dataset: j9rlb0fxq_drz). This is the same image of Fig 2.1 but plotted with the planetary model grids and lines of limb and terminator generated by the algorithm of Appendix 1. The center of the planet is checked with SPICE tool kit and IMCCE query service. Then the center point is chosen manually corrected one. The scale of the image is transformed pixel to arcsec.

2.4.3 Polar projection

Polar projection is used to change the perspective of the observer to the two dimensional planetary image. This method allows us to look down the polar region of the target planet by performing the transformation (See Fig 2.6). It must be noted that we need to take into account a possible tilt of the rotation axis of the planetary model before this transformation (explained in previous section). Since the axis of the planet in the image is not always the same as y_s . To get the polar projected view from the two dimensional planetary image, firstly, we need to project back the pixel picture images ($(x_s, y_s, z_s = 0)$ -coordinate in Fig 2.6) onto the surface of the planetary model in $(-z, y, x)$ -coordinate. Then project it again into the objective direction of the sky plane.

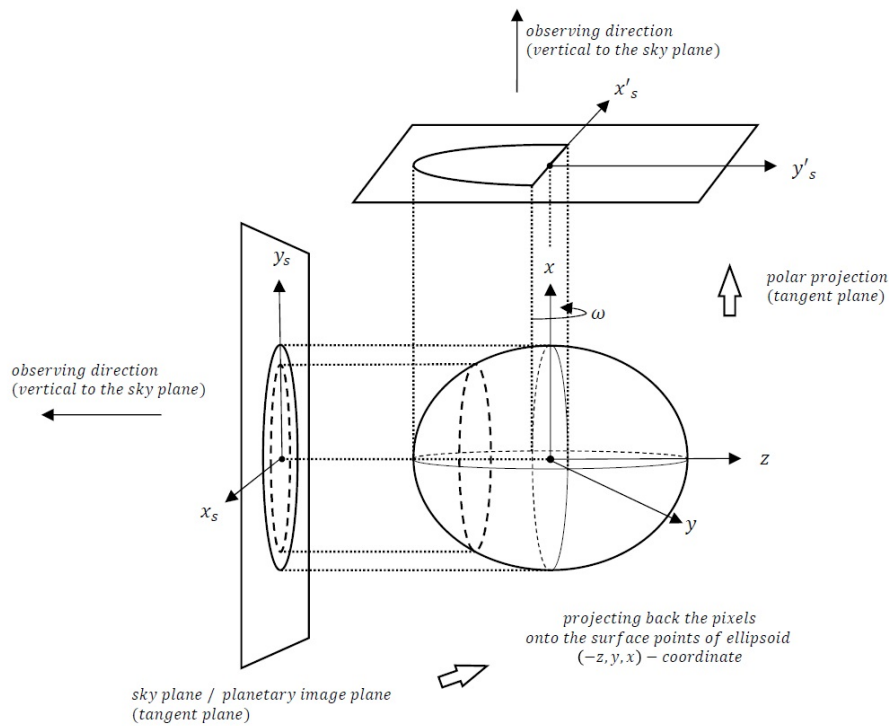


Figure 2.6: Sketch of transformation of two dimensional planetary image into polar projection.

The polar projection of the Jupiter image (Dataset: j9rlb0fxq_drz) is shown in Fig 2.7. The number of image pixels around the polar region in the original image does not have sufficient pixels to cover the polar region of the polar projected image. The dispersion of the pixels around projected polar region causes the bad effect in the image, so we need to interpolate it to reduce it.

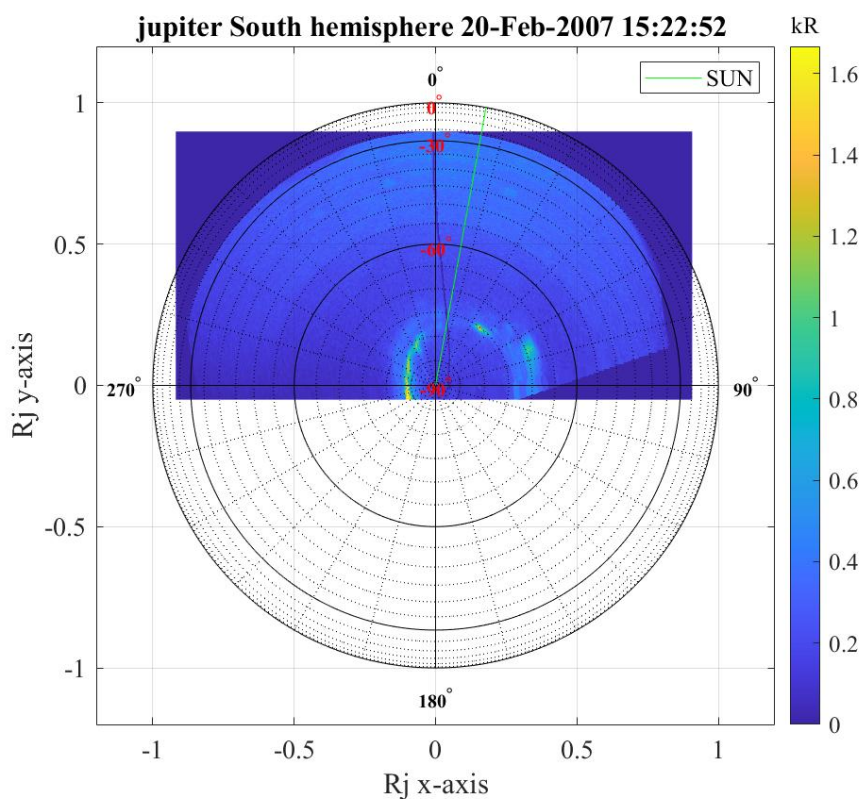


Figure 2.7: The image of Jupiter in polar projection (Dataset: j9rlb0fxq_drz) with the planetary model grid. The azimuth of 0° is the CML line. The image scale is transformed arcsec to the scale of Jovian radius R_J . The intensity of the image converted into auroral brightnesses (1 kR represents a photon source flux of $10^9 \text{ cm}^{-2} \text{ s}^{-1}$ radiating into 4π steradians) using the conversion factor $1 \text{ kR} = 1.473 \times 10^{-3} \text{ counts s}^{-1} \text{ pixel}$ [11] of total H_2 emission over the 80-170 nm spectral range.

2.5 Airglow model and subtract it from the image

To get clear auroral forms in the planetary image, subtracting the background airglow allows us to observe the auroral emission with clear location and size. Airglow is the natural “glowing” of the atmosphere of the planet. Dayglow forms when sunlight strikes the daytime atmosphere. Some of the sunlight is absorbed by the molecules in the atmosphere, which gives them excess energy. They become excited, the molecules then release this energy as light. This illumination modeling is also helpful to feed the astronomical image to VOISE algorithm to detect the auroral features.

2.5.1 Cosine illumination model

The illumination intensity from the planet surface at the 1 bar level can be modeled by the observed intensity and the cosine of the angles between the surface normal $\hat{\mathbf{n}}_s$ and the direction of the observer and the Sun. In this section, taking the observer direction as an example, we detail the cosine illumination model.

In the *planetocentric* coordinate system, the ellipsoidal model of the planet which has semi-major axis r_e , latitude β , and longitude λ , we have

$$\begin{cases} x = r_e \cos \lambda \cos \beta \\ y = r_e \sin \lambda \cos \beta \\ z = r_e \sqrt{1 - e^2} \sin \beta \end{cases} . \quad (2.2)$$

Here e is the eccentricity of the spheroid. The observing direction in the planetocentric coordinate δ_{obs} (see Fig 4.1) with the latitude angle β_{obs} , which is given by

$$\delta_{\text{obs}} = (\cos \beta_{\text{obs}}, \quad 0, \quad \sin \beta_{\text{obs}}) . \quad (2.3)$$

In addition, the direction normal to the ellipsoid at a point (x, y, z) is defined by the vector $\hat{\mathbf{n}}_s$. For an implicit function $F(x, y, z) = 0$ such as a spheroid, the vector $\hat{\mathbf{n}}_s$ is given by the gradient ∇F . In Cartesian coordinate system the normal $\hat{\mathbf{n}}_s$ at any point of the spheroid is given by

$$\hat{\mathbf{n}}_s = \frac{1}{\sqrt{x^2 + y^2 + z^2/(1 - e^2)^2}} \left(x, \quad y, \quad \frac{z}{1 - e^2} \right) . \quad (2.4)$$

Fig 2.8 shows, the intensity at the point P_s on the image-plane corresponds to a point P on the surface of the planetary model. For the observer, the intensity at P_s pixel has a factor of $\cos \theta$ to the originally radiated intensity. By using the observer

zenith angle θ which is the angle between the surface normal \hat{n}_s and the observer direction. The cosine can be written as

$$\begin{aligned} \cos \theta &= \frac{\hat{n}_s \cdot \delta_{\text{obs}}}{\|\hat{n}_s\| \|\delta_{\text{obs}}\|} \\ &= \frac{1}{\sqrt{x^2 + y^2 + z^2/(1-e^2)^2}} \left(x \cos \beta_{\text{obs}} + \frac{z}{1-e^2} \sin \beta_{\text{obs}} \right). \end{aligned} \quad (2.5)$$

This cosine illumination model is calculated for both the Observer and the Solar zenith angles. These angles are derived by using the sub-Earth and -Solar points on the modeled planetary surface. Using this feature, two background airglow models are performed in Section 2.5.2.

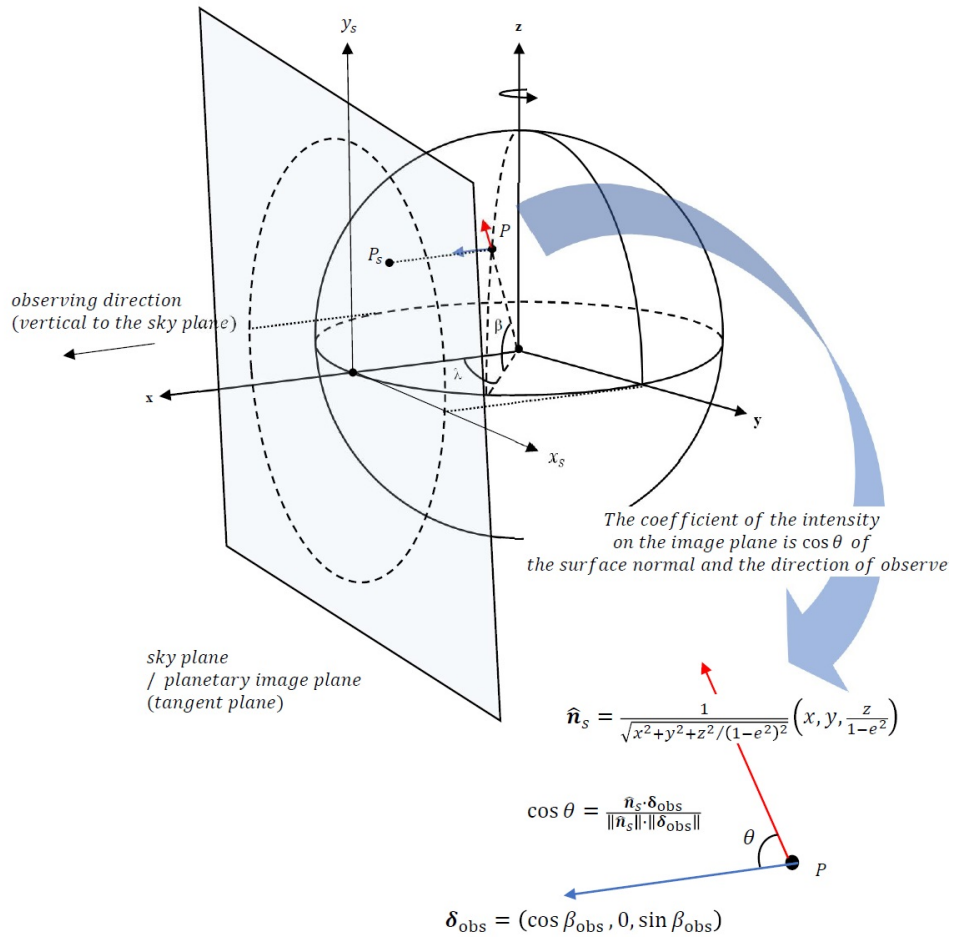


Figure 2.8: The illustration of the observer zenith angle θ and the cosine.

2.5.2 Minnaert model

The Minnaert function is a photometric function used to interpret astronomical observations and remote sensing data for the Earth. The FUV imaging instruments of the Space Telescope Spectrograph (STIS) and the Advanced Camera for Surveys (ACS) capture the planetary disks which mainly originate from reflected sunlight on the planet. The classical method to build the back ground dayglow model is the generalised Minnaert function[18, 2]

$$\ln(I\mu) = A + Bx + Cx^2 + Dx^3. \quad (2.6)$$

Where I is the observed intensity, μ and μ_0 are the cosines of the observer zenith angles and solar zenith angles respectively, $x = \ln(\mu\mu_0)$, A , B , C , and D are derived coefficients determined by fitting to the intensity at a selected latitude range. There are asymmetry in the fitting of the intensity profile between where the sunlit limb and the terminator limb occurs[18].

For a good limb darkening, two coefficients set at both the sunlit limb and terminator limb are required. My application of the Minnaert function consisted of deriving two set of coefficients which is one set each for the sunlit side and the terminator side of the planet. The original method is provided in the reference paper[18]. The algorithm is following. The coefficients for the sunlit limb were derived at a given latitude over longitudes ranging from about 85° off the Central meridian longitude (CML) toward the sunlit limb to 45° CML toward the terminator. Similarly, the coefficients for the terminator side were derived over the longitudes ranging from about 70° off the CML toward the terminator to 45° beyond the CML toward the sunlit limb. The two set of coefficients are linearly weighted with respect to longitude.

I used the one degree latitudinal bins for model fitting to the image intensity and performed it using the Matlab curve fitting toolbox. As the Fig 2.9 shows, the asymmetry is observed in the plot. My application of this algorithm in Matlab code is attached in Appendix 3.

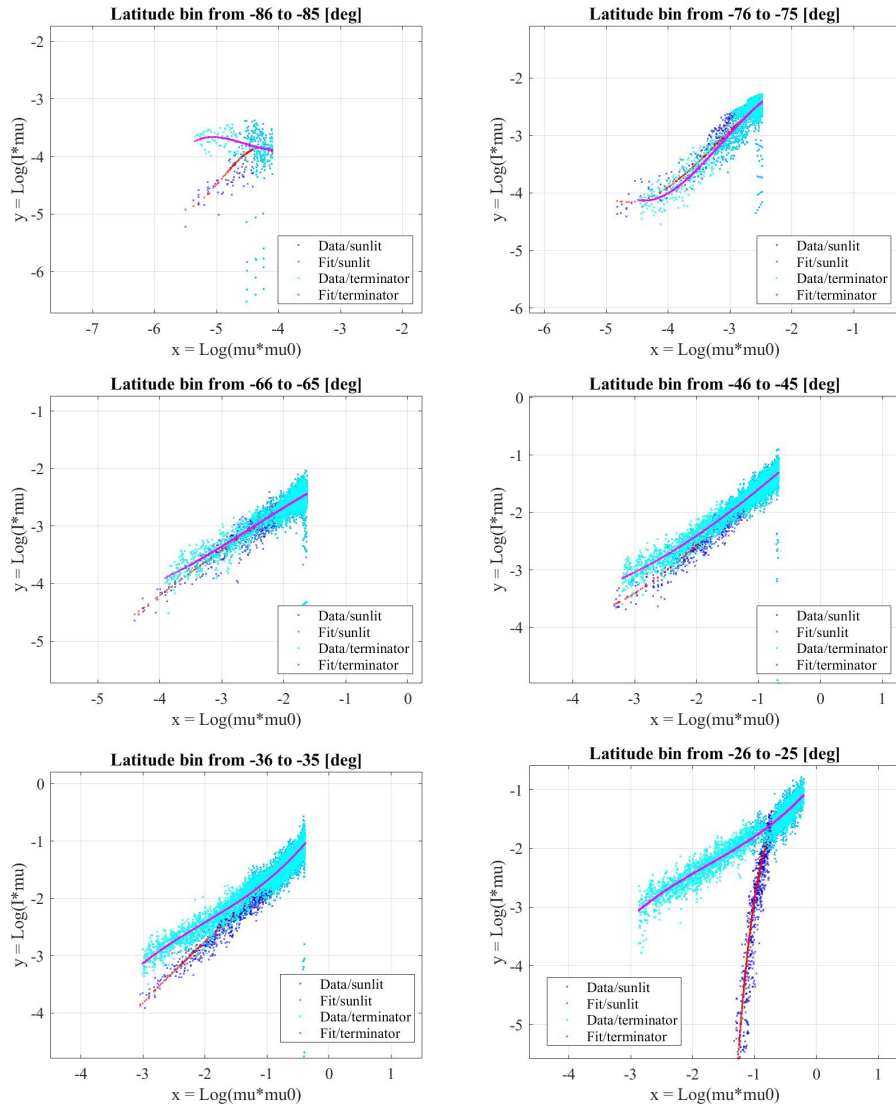


Figure 2.9: S scatter plot of $\ln(I\mu)$ as a function of $\ln(\mu\mu_0)$ with 1 degree resolution. The asymmetry of the sunlit side and the terminator side is clearly seen in the latitude bin close to the pole and the equator. The asymmetry in the polar latitude bin is considered to be caused by the auroral emission.

2.5.3 Li model

This method removes dayglow effects from UV images by using the sum of the cosines of the Observation and Solar zenith angles[10]. Detecting the auroral oval bounds in magnetic latitude (MLAT), then using the non-auroral dayglow pixels, allow to estimate the non-auroral intensity as a function of the cosine of the Solar and Observer zenith angles. The author of the paper[10] introduced a threshold technique, because of the comparable intensities of aurora and dayglow pixels. This technique consists of two parts, the first stage identifies the geomagnetic latitude range of the auroral oval, then the second stage estimates the dayglow intensity with the model function.

First stage: Geomagnetic latitude range of the auroral oval

In this stage, we determine the magnetic latitude (MLAT) range of auroral ovals. I binned the night side and day side pixels on the MLAT values with 1° resolution to compare with the Minnaert model. Then the mean intensity value as the function of MLAT is obtained. The author presented the automatic determination method using only the night side pixels to determine the auroral oval region. But this method does not cover sufficient auroral emission area in the image I used in this project. To cover the auroral oval region and to get non-auroral pixel intensities, I used the threshold condition as follows. I defined that the night side is from sub-Solar longitude dusk and the day side is from sub-Solar longitude to dawn. Firstly, the location of the peak intensity in the MLAT range is obtained in both day and night side. Then the minimum intensity values for both directions of pole-ward and equator-ward are checked for day and night side. These minimum along and peak values are used to determine the equator-ward and pole-ward bounds all over the auroral oval based on the following subjective criteria. The MLAT values for the auroral bounds pole-ward and equator-ward for day and night side, are where the MLAT value of the intensity drops from the peak value to below 80 % of the intensity difference between peak value and minimum value. If there is no applicable mean intensity to below 80 % of the intensity difference between peak value and minimum value, the minimum mean intensity value is chosen as the boundary. The histogram plot of MLAT vs mean intensity for both day and night side are shown in the Fig 2.10. The boundaries between day and night side are different. So I used the pole-ward boundary of -79° from night side and equator-ward boundary of -66° from day side to model the dayglow. The numerical model is presented in the second stage. Fig 2.11 shows the peak auroral region intensity in MLAT range of -87° to -66° , and Fig 2.12 shows the non auroral region pixel intensity vs cosine of Solar Zenith Angle (SZA) and Observer Zenith Angle (OZA).

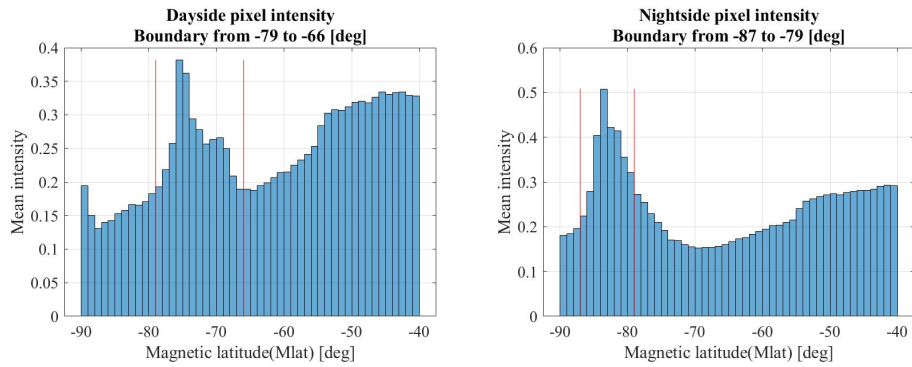


Figure 2.10: There is a difference in day and night side mean intensity function of MLAT. The range between the red vertical lines is determined as the auroral oval region following criteria. The outside of the red vertical lines is the non-auroral region.

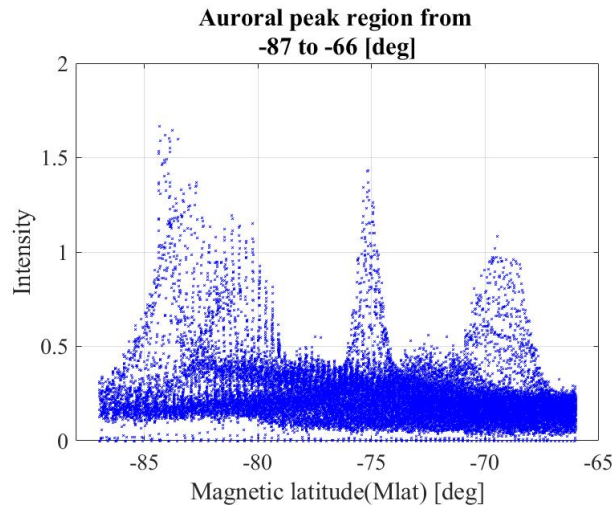


Figure 2.11: Peak auroral region intensity in MLAT range of -87° to -66° . There are seen some peaks of auroral emissions.

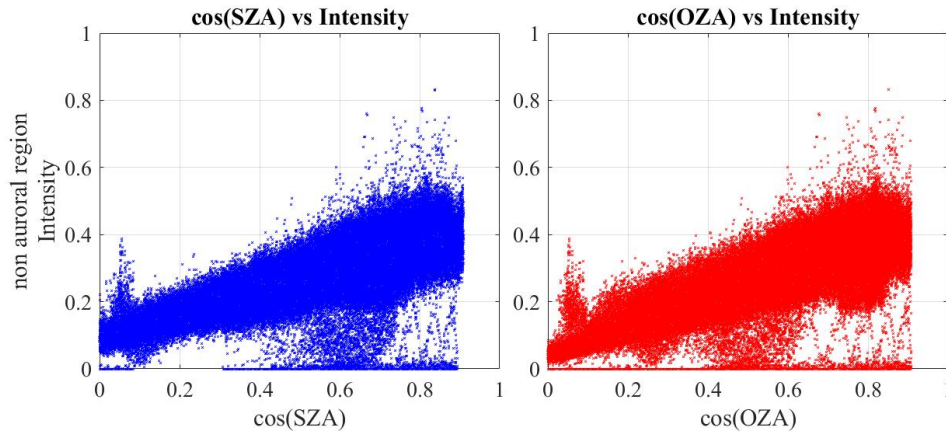


Figure 2.12: Non auroral region pixel intensity vs cosine of Solar Zenith Angle (SZA) and Observer Zenith Angle (OZA).

Second: Dayglow model

The dayglow model with the Li method is based on the fact that the intensities of dayglow pixels at FUV can be estimated as a function of the SZA and the OZA. The following equation is used for the Li model,

$$I = A + B \cos(SZA) + C \cos(OZA). \quad (2.7)$$

Where I is the estimated dayglow intensity, A , B , and C are the fitting coefficients. I used the one degree latitudinal bins for model fitting the image intensity and performed the fitting to the non-auroral region intensities, with the Matlab curve fitting toolbox. My application of this algorithm is presented in Appendix 4.

/3

Result and discussion

3.1 Result of airglow models

The output of the calculation in polar projections of subtracted images are shown in Fig 3.1. The Minnaert model is in Fig 3.2 and the Li model is in Fig 3.3.

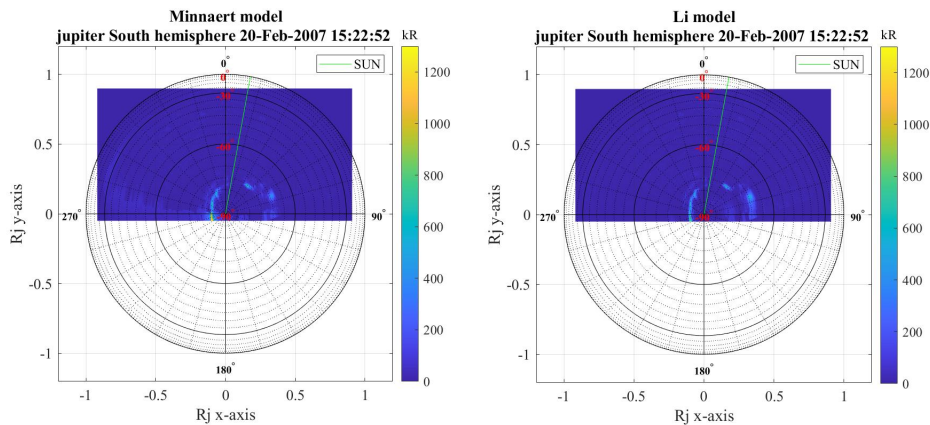


Figure 3.1: The polar projection of subtracted image. The image scale is the Jovian radius R_J . The azimuth of 0° is the CML line. The intensity of the image is converted into auroral brightnesses (1 kR represents a photon source flux of $10^9 \text{ cm}^{-2} \text{ s}^{-1}$ radiating into 4π steradians) using the conversion factor $1 \text{ kR} = 1.473 \times 10^{-3} \text{ counts s}^{-1} \text{ pixel}$ [11] for total H_2 emission over the 80-170 nm spectral range.

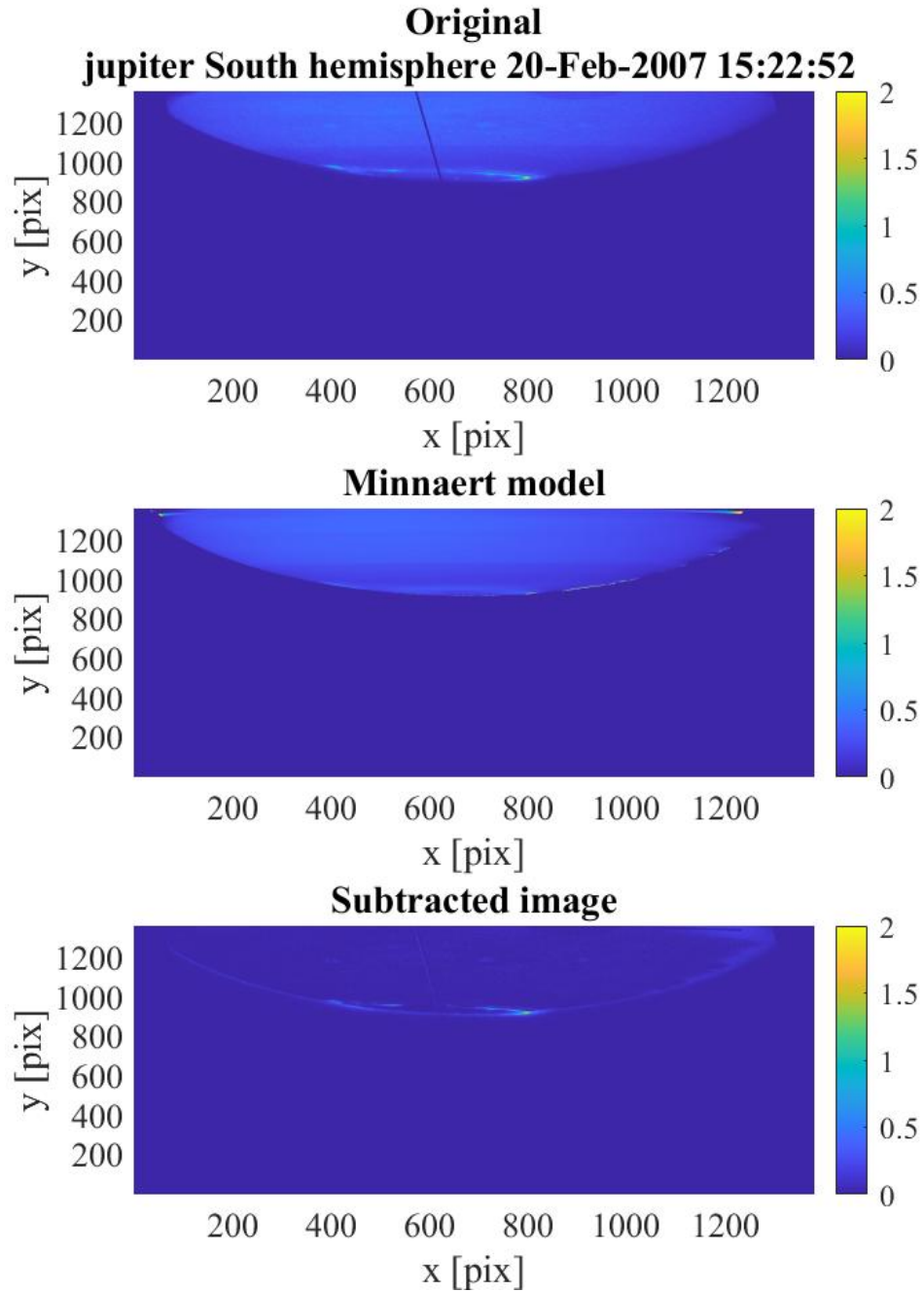


Figure 3.2: (Top) Original image (Dataset: j9rlb0fxq_drz). The intensity unit is the numbers of photons count per Pixel (Middle) calculated Minnaert model from the original image with the Minnaert model with 1 degree latitude bins, (Bottom) The subtracted image. The scale of the image is in pixel.

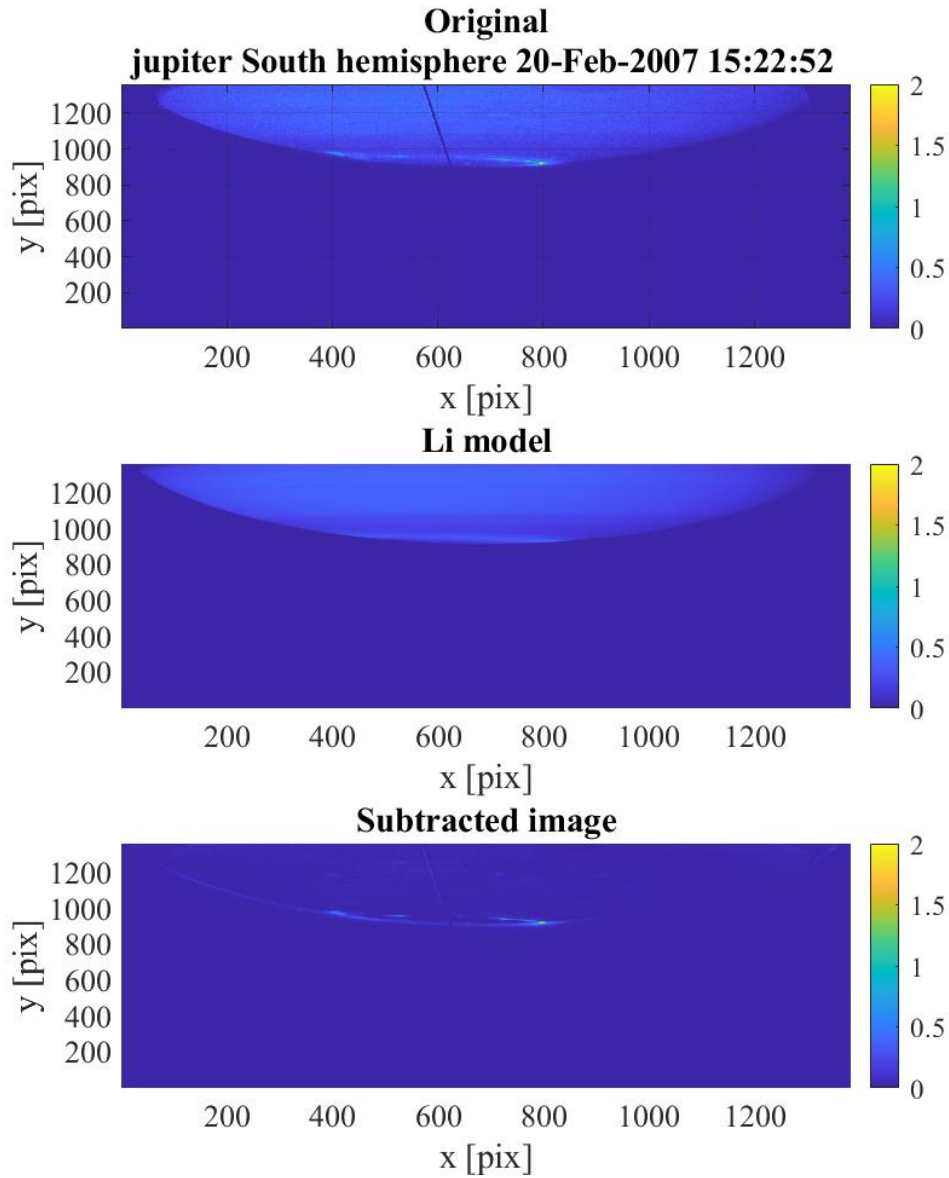


Figure 3.3: (Top) Original image (Dataset: j9rlb0fxq_drz). The intensity unit is the numbers of photons count per Pixel (Middle) calculated Li model from the original image with the Minnaert model with 1 degree latitude bins, (Bottom) The subtracted image. The scale of the image is in pixel.

3.2 Discussion and conclusion

The objective of this project is to develop a planetary auroral image processing tool around VOISE. Through the processes of geometric transformation and airglow modeling, This has been partly achieved. There are some points needing to be improved. First, the correction of the center point of the planetary model could be improved with high accuracy by the technique to estimate the disc parameter which is presented in the literature[5]. In this project the center point was corrected by manually. Secondary, the airglow models subtracts too much intensity from the original. Both the Minnaert model and the Li model work well to subtract the airglow from the original image. However my Minnaert model does not fit the original intensity well. The reason for this is thought that the the auroral intensity is not appropriately excluded from the fitting process of airglow.

/4

Appendix 1: Planetary model from Professor Patrick Guio's note

4.1 Planetary model

The planet is modelled by an ellipsoid, which equatorial radius (semi-major axis) r_e and polar radius (semi-minor axis) r_p , where

$$r_p^2 = r_e^2(1 - e^2), \quad (4.1)$$

here e is the eccentricity of the spheroid. The planet rotation vector is assumed to be along the z -axis with positive angular velocity $\omega > 0$ corresponds to the right-hand grip rule. In Cartesian coordinate system the ellipsoid is written in

$$\frac{x^2 + y^2}{r_e^2} + \frac{z^2}{r_e^2(1 - e^2)} = 1. \quad (4.2)$$

In spherical coordinate system where ϕ is the zenith angle or co-latitude ($0 \leq \phi \leq 2\pi$) and θ is the azimuth or longitude ($0 \leq \theta \leq 2\pi$) we have

$$\begin{cases} x = r_e \cos \theta \sin \phi \\ y = r_e \sin \theta \sin \phi \\ z = r_e \sqrt{1 - e^2} \cos \phi \end{cases}. \quad (4.3)$$

In *planetocentric* coordinate system where β is the latitude, i.e. the complementary angle to the co-latitude ϕ , $\beta + \phi = \pi/2$, $(-\pi/2 \leq \beta \leq \pi/2)$ and λ is the longitude $(-\pi \leq \lambda \leq \pi)$ we have

$$\begin{cases} x = r_e \cos \lambda \cos \beta \\ y = r_e \sin \lambda \cos \beta \\ z = r_e \sqrt{1 - e^2} \sin \beta \end{cases} . \quad (4.4)$$

The direction normal to the ellipsoid at point (x, y, z) -coordinate is defined by the vector \hat{n}_s . For an implicit function $F(x, y, z) = 0$ such as a spheroid, the vector \hat{n}_s is given by the gradient ∇F . In Cartesian coordinate system the normal \hat{n}_s at any point of the spheroid is given by

$$\hat{n}_s = \frac{1}{\sqrt{x^2 + y^2 + z^2/(1 - e^2)^2}} \left(x, \quad y, \quad \frac{z}{1 - e^2} \right) . \quad (4.5)$$

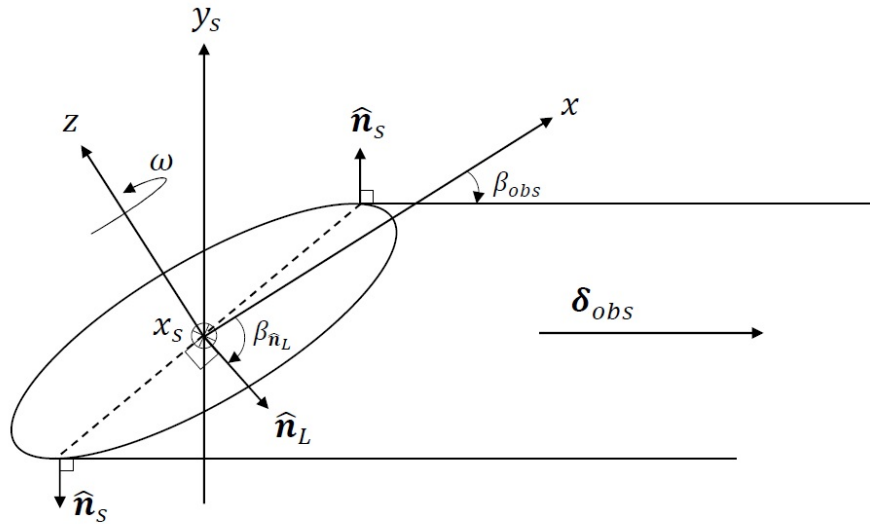


Figure 4.1: (From [7]) Sketch of the geometry of the planet and the observer. The eccentricity of the planetary ellipsoid is exaggerated for clarity. The figure is a cut through the planet that contains the planet's rotation axis z and the observer direction δ_{obs} . The x -axis is contained in the equatorial plane of the planet and in the plane of the figure. The direction \hat{n}_s is the local normal of the ellipse perpendicular to the observing direction δ_{obs} . The plane of the limb is contained in a plane perpendicular to the figure and its intersection with the plane of the figure is the dashed line segment. \hat{n}_L is the direction normal to the plane of the limb and pointing toward the observer. The angle between \hat{n}_L and the x -axis is $\beta_{\hat{n}_L}$. Finally (x_s, y_s) are the axes for the plane of the sky as seen by the observer.

4.2 Limb projection

For the limb projection calculation, we assume, without loss of generality, the observing direction in the planetocentric coordinate $\boldsymbol{\delta}_{\text{obs}}$ with latitude angle β_{obs} , which written by

$$\boldsymbol{\delta}_{\text{obs}} = (\cos \beta_{\text{obs}}, \quad 0, \quad \sin \beta_{\text{obs}}). \quad (4.6)$$

The plane of the limb of the planet consists of the points on the planet surface where the normal to the surface $\hat{\boldsymbol{n}}_s$, defined in Eq.(3.5), which is perpendicular to the observing direction $\boldsymbol{\delta}_{\text{obs}}$ (see Fig 3.1). Taking into account the relation $\hat{\boldsymbol{n}}_s \cdot \boldsymbol{\delta}_{\text{obs}} = 0$, the limb is thus defined analytically by the set of equations

$$\begin{aligned} \hat{\boldsymbol{n}}_s \cdot \boldsymbol{\delta}_{\text{obs}} &= x \cos \beta_{\text{obs}} + z \frac{\sin \beta_{\text{obs}}}{(1 - e^2)} = 0 \\ &= x(1 - e^2) \cos \beta_{\text{obs}} + z \sin \beta_{\text{obs}} = 0, \end{aligned} \quad (4.7)$$

$$\frac{x^2 + y^2}{r_e^2} + \frac{z^2}{r_e^2(1 - e^2)} = 1. \quad (4.8)$$

Eq.(3.7) expresses the fact that the limb is contained in a plane with normal vector $\hat{\boldsymbol{n}}_L$ defined by

$$\begin{aligned} \hat{\boldsymbol{n}}_L &= \frac{1}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}} \\ &\cdot ((1 - e^2) \cos \beta_{\text{obs}}, \quad 0, \quad \sin \beta_{\text{obs}}). \end{aligned} \quad (4.9)$$

We define following set of angles for the vector $\hat{\boldsymbol{n}}_L$. (see Fig 3.1)

$$\begin{cases} \cos \beta_{\hat{\boldsymbol{n}}_L} = \frac{1}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}} (1 - e^2) \cos \beta_{\text{obs}} \\ \sin \beta_{\hat{\boldsymbol{n}}_L} = \frac{1}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}} \sin \beta_{\text{obs}} \end{cases}, \quad (4.10)$$

then Eq.(3.9) becomes

$$\hat{\boldsymbol{n}}_L = (\cos \beta_{\hat{\boldsymbol{n}}_L}, \quad 0, \quad \sin \beta_{\hat{\boldsymbol{n}}_L}). \quad (4.11)$$

When the eccentricity $e = 0$, i.e. the planet is sphere (like the Moon), the direction normal to the plane containing the limb, defined by $\hat{\boldsymbol{n}}_L$, which is aligned to the observing direction $\boldsymbol{\delta}_{\text{obs}}$. In the case where $e \neq 0$ they are not aligned and we see from Eq.(3.10) that $|\beta_{\hat{\boldsymbol{n}}_L}| > |\beta_{\text{obs}}|$. Let us define the following coordinate transformation which transforms any point of (x, y, z) into a point (x', y', z')

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\sin \beta_{\hat{\boldsymbol{n}}_L} & 0 & \cos \beta_{\hat{\boldsymbol{n}}_L} \\ \cos \beta_{\hat{\boldsymbol{n}}_L} & 0 & \sin \beta_{\hat{\boldsymbol{n}}_L} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (4.12)$$

This transformation transforms the plane of the limb given by Eq.(3.7) into the (x', y') -plane, so that $z' = 0$. Since it is an *orthogonal transformation*, the inverse coordinate transformation is defined by the transpose of the matrix given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 & -\sin \beta_{\hat{n}_L} & \cos \beta_{\hat{n}_L} \\ 1 & 0 & 0 \\ 0 & \cos \beta_{\hat{n}_L} & \sin \beta_{\hat{n}_L} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}. \quad (4.13)$$

Let us transform the limb defined by Eqs.(3.7-8) using the inverse transformation. Assuming the ellipsoid is not degenerated ($e \neq 1$), being careful with $z' = 0$ the calculation Eq.(3.13) gives

$$\begin{cases} x = -y' \sin \beta_{\hat{n}_L} \\ y = x' \\ z = y' \cos \beta_{\hat{n}_L} \end{cases},$$

then do following preparation

$$\begin{aligned} x^2 &= y'^2 \sin^2 \beta_{\hat{n}_L}, \\ y^2 &= x'^2, \\ x^2 + y^2 &= x'^2 + y'^2 \sin^2 \beta_{\hat{n}_L}, \end{aligned}$$

inserting this into Eq.(3.8), we have

$$\frac{x'^2 + y'^2 \sin^2 \beta_{\hat{n}_L}}{r_e^2} + \frac{y'^2 \cos^2 \beta_{\hat{n}_L}}{r_e^2 (1 - e^2)} = 1,$$

finally,

$$\begin{aligned} z' &= 0, \\ \frac{x'^2}{r_e^2} + y'^2 \frac{\cos^2 \beta_{\hat{n}_L} + (1 - e^2) \sin^2 \beta_{\hat{n}_L}}{r_e^2 (1 - e^2)} &= 1. \end{aligned}$$

Let us take into account Eq.(3.10). Then the Eq.(3.8) becomes

$$\frac{x'^2}{r_e^2} + \frac{y'^2}{r_e^2} \frac{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + (1 - e^2) \sin^2 \beta_{\text{obs}}}{(1 - e^2) [(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}]} = 1. \quad (4.14)$$

Let us take a look at the inverse of the factor of $\frac{y'^2}{r_e^2}$ in Eq.(3.14),

$$\begin{aligned}
\frac{(1 - e^2) \left[(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}} \right]}{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + (1 - e^2) \sin^2 \beta_{\text{obs}}} &= \frac{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}{(1 - e^2) \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}} \\
&= \frac{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}{(1 - e^2) \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}} \\
&= 1 - \frac{e^2 (1 - e^2) \cos^2 \beta_{\text{obs}}}{1 - e^2 \cos^2 \beta_{\text{obs}}} \\
&= 1 - e^2 \left(\frac{\cos^2 \beta_{\text{obs}} - e^2 \cos^2 \beta_{\text{obs}}}{1 - e^2 \cos^2 \beta_{\text{obs}}} \right) \\
&= 1 - e^2 \left(\frac{1 - \sin^2 \beta_{\text{obs}} - e^2 \cos^2 \beta_{\text{obs}}}{1 - e^2 \cos^2 \beta_{\text{obs}}} \right) \\
&= 1 - e^2 \left(1 - \frac{\sin^2 \beta_{\text{obs}}}{1 - e^2 \cos^2 \beta_{\text{obs}}} \right).
\end{aligned}$$

We find the eccentricity of the ellipse corresponding to the limb in *its own plane* can be expressed as a function of the observation latitude β_{obs} and the planetary model's eccentricity e

$$e_{\beta_{\text{obs}}}^2 = e^2 \left(1 - \frac{\sin^2 \beta_{\text{obs}}}{1 - e^2 \cos^2 \beta_{\text{obs}}} \right). \quad (4.15)$$

The transformed Eq.(3.8), which is in $(x', y', z' = 0)$ -coordinate, can be expressed by the canonical form

$$\frac{x'^2}{r_e^2} + \frac{y'^2}{r_e^2 (1 - e_{\beta_{\text{obs}}}^2)} = 1. \quad (4.16)$$

Eq.(3.16) is an ellipse with centre $(0, 0)$, semi-major axis $r_e \mathbf{i}'$, and semi-minor axis $r_e \sqrt{1 - e_{\beta_{\text{obs}}}^2} \mathbf{j}'$.

Here, we introduce the basis vectors $(\mathbf{i}', \mathbf{j}')$ of the (x', y') -plane (since $z' = 0$), where the plane that the transformed limb is on. The basis vectors are given in (x, y, z) -coordinate given by

$$\mathbf{i}' = (0, 1, 0), \quad (4.17)$$

$$\mathbf{j}' = (-\sin \beta_{\hat{n}_L}, 0, \cos \beta_{\hat{n}_L}). \quad (4.18)$$

Let us put the transformed limb into the sky plane, (x_s, y_s) -coordinate, which is perpendicular to observing direction $\boldsymbol{\delta}_{\text{obs}}$. In the (x, y, z) -plane. *The sky plane* has basis vectors $(\mathbf{i}_s, \mathbf{j}_s)$ such that

$$\mathbf{i}_s = (0, 1, 0), \quad (4.19)$$

$$\mathbf{j}_s = (-\sin \beta_{\text{obs}}, 0, \cos \beta_{\text{obs}}). \quad (4.20)$$

and remind that the direction orthogonal to *the sky plane*, \mathbf{k}_s , is equal to the observing direction $\boldsymbol{\delta}_{\text{obs}}$

$$\mathbf{k}_s = (\cos \beta_{\text{obs}}, \quad 0, \quad \sin \beta_{\text{obs}}). \quad (4.21)$$

The projection of the basis vectors $(\mathbf{i}', \mathbf{j}')$ onto the basis vectors of *the sky plane* $(\mathbf{i}_s, \mathbf{j}_s)$ are defined by

$$\mathbf{i}_s = (\mathbf{i}' \cdot \mathbf{i}_s)\mathbf{i}' + (\mathbf{j}' \cdot \mathbf{i}_s)\mathbf{j}', \quad (4.22)$$

$$\mathbf{j}_s = (\mathbf{i}' \cdot \mathbf{j}_s)\mathbf{i}' + (\mathbf{j}' \cdot \mathbf{j}_s)\mathbf{j}', \quad (4.23)$$

Let us do calculations for each factors

$$\begin{aligned} \mathbf{i}' \cdot \mathbf{i}_s &= 0 + 1 + 0 \\ &= 1 \end{aligned}$$

$$\begin{aligned} \mathbf{j}' \cdot \mathbf{i}_s &= 0 + 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} \mathbf{i}' \cdot \mathbf{j}_s &= 0 + 0 + 0 \\ &= 0 \end{aligned}$$

$$\begin{aligned} \mathbf{j}' \cdot \mathbf{j}_s &= \sin \beta_{\text{obs}} \sin \beta_{\hat{\mathbf{n}}_L} + \cos \beta_{\text{obs}} \cos \beta_{\hat{\mathbf{n}}_L} \\ &= \cos \beta_{\text{obs}} \cos \beta_{\hat{\mathbf{n}}_L} + \sin \beta_{\text{obs}} \sin \beta_{\hat{\mathbf{n}}_L} \end{aligned}$$

then Eqs.(3.22-23) become

$$\mathbf{i}_s = \mathbf{i}', \quad (4.24)$$

$$\mathbf{j}_s = (\cos \beta_{\text{obs}} \cos \beta_{\hat{\mathbf{n}}_L} + \sin \beta_{\text{obs}} \sin \beta_{\hat{\mathbf{n}}_L}) \mathbf{j}'. \quad (4.25)$$

The Eqs.(3.24-25) shows that the projection of the limb into *the sky plane* has no-effect on the semi-major axis, but has an effect on the semi-minor axis with the scale factor of $(\cos \beta_{\text{obs}} \cos \beta_{\hat{\mathbf{n}}_L} + \sin \beta_{\text{obs}} \sin \beta_{\hat{\mathbf{n}}_L})$. This factor can be simplified into $\cos(\beta_{\text{obs}} - \beta_{\hat{\mathbf{n}}_L})$ and is given by $\boldsymbol{\delta}_{\text{obs}} \cdot \hat{\mathbf{n}}_L$ using Eq.(3.6) and Eq.(3.9),

$$\begin{aligned} \boldsymbol{\delta}_{\text{obs}} \cdot \hat{\mathbf{n}}_L &= \cos(\beta_{\text{obs}} - \beta_{\hat{\mathbf{n}}_L}) \\ &= \frac{(1 - e^2) \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}} = \frac{1 - e^2 \cos^2 \beta_{\text{obs}}}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{obs}} + \sin^2 \beta_{\text{obs}}}}. \end{aligned} \quad (4.26)$$

Finally the projection of the limb into *the sky plane* appearing as the ellipse is written in *the sky plane*, (x_s, y_s) -coordinate as

$$\frac{x_s^2}{r_e^2} + \frac{y_s^2}{r_e^2(1 - e^2 \cos^2 \beta_{\text{obs}})} = 1. \quad (4.27)$$

The Eq.(3.27) shows that the ellipse of the limb, seen from the observer, always appearing with the same semi-major axis as the equatorial radius of the planet. But

the semi-minor axis changes depending on the observing latitude β_{obs} . When the observer is watching down from the polar region of the planet, the semi-minor axis becomes equal to the equatorial radius. The eccentricity of the limb projected into *the sky plane*, is written by

$$e_L = e \cos \beta_{\text{obs}}, \quad (4.28)$$

as the same way, the ellipse in the *the sky plane*, (x_s, y_s) -coordinate, can be expressed in the canonical form

$$\frac{x_s^2}{r_e^2} + \frac{y_s^2}{r_e^2(1 - e_L^2)} = 1. \quad (4.29)$$

4.3 Terminator projection

A terminator or twilight zone is a moving line that divides the daylight side and the dark night side on the planetary body. The terminator lines can also be interpreted as the limb for an *observing direction* corresponding to the illumination source and further projected into *the sky plane* along the observing direction. In this section, we present the method to project the terminator into *the sky plane*, (x_s, y_s) -coordinate. We can define, without loss of generality, the coordinate system (x_1, y_1, z_1) , where z_1 is the planet rotation axis, which is identical to z . The x_1 is defined so that the (x_1, z_1) -plane contains the illumination source, as we saw in the limb calculation section, and the latitude of the illumination source, β_{sun} , moreover, introduce the *phase angle* between the x -axis and x_1 -axis, which is written by $\Delta\lambda = \lambda_{\text{sun}} - \lambda_{\text{obs}}$. The coordinate transformation from (x, y, z) into (x_1, y_1, z_1) is equivalent to a rotation along the z -axis with the negative quantity of *the phase angle*, $-\Delta\lambda$.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos \Delta\lambda & \sin \Delta\lambda & 0 \\ -\sin \Delta\lambda & \cos \Delta\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (4.30)$$

and inversely, the transformation from (x_1, y_1, z_1) into (x, y, z) is obtained by taking the transpose of the matrix which is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos \Delta\lambda & -\sin \Delta\lambda & 0 \\ \sin \Delta\lambda & \cos \Delta\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}. \quad (4.31)$$

In the (x_1, y_1, z_1) -coordinate system, the direction of the illumination source, the sun, δ_{sun} , is similarly to the limb calculation case, given by

$$\delta_{\text{sun}} = (\cos \beta_{\text{sun}}, \quad 0, \quad \sin \beta_{\text{sun}}), \quad (4.32)$$

while in the (x, y, z) -coordinate system, δ_{sun} is transformed by Eq.(3.31), becomes

$$\delta_{\text{sun}} = (\cos \beta_{\text{sun}} \cos \Delta\lambda, \quad \cos \beta_{\text{sun}} \sin \Delta\lambda, \quad \sin \beta_{\text{sun}}). \quad (4.33)$$

In the (x_1, y_1, z_1) -coordinate system, the terminator is defined with similar equations like as Eqs.(3.7-8). We have $\hat{\mathbf{n}}_s \cdot \boldsymbol{\delta}_{\text{sun}} = 0$ and the equation of ellipse given by

$$\begin{aligned} \hat{\mathbf{n}}_s \cdot \boldsymbol{\delta}_{\text{sun}} &= x_1 \cos \beta_{\text{sun}} + z_1 \frac{\sin \beta_{\text{sun}}}{1 - e^2} = 0 \\ &= x_1(1 - e^2) \cos \beta_{\text{sun}} + z_1 \sin \beta_{\text{sun}} = 0, \end{aligned} \quad (4.34)$$

$$\frac{x_1^2 + y_1^2}{r_e^2} + \frac{z_1^2}{r_e^2(1 - e^2)} = 1. \quad (4.35)$$

The ellipse formed by the terminator is contained in a plane with normal vector $\hat{\mathbf{n}}_T$ defined by in a similar manner as for the limb normal calculation in the Eqs.(3.9-11). Eq.(3.34) expresses the fact that the terminator is contained in a plane with normal vector $\hat{\mathbf{n}}_T$ defined by

$$\hat{\mathbf{n}}_T = \frac{1}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{sun}} + \sin^2 \beta_{\text{sun}}}} \cdot ((1 - e^2) \cos \beta_{\text{sun}}, \quad 0, \quad \sin \beta_{\text{sun}}), \quad (4.36)$$

We define following set of angles for the vector $\hat{\mathbf{n}}_T$

$$\begin{cases} \cos \beta_{\hat{\mathbf{n}}_T} = \frac{1}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{sun}} + \sin^2 \beta_{\text{sun}}}} (1 - e^2) \cos \beta_{\text{sun}} \\ \sin \beta_{\hat{\mathbf{n}}_T} = \frac{1}{\sqrt{(1 - e^2)^2 \cos^2 \beta_{\text{sun}} + \sin^2 \beta_{\text{sun}}}} \sin \beta_{\text{sun}} \end{cases}, \quad (4.37)$$

then Eq.(3.36), which is the terminator normal $\hat{\mathbf{n}}_T$ in the (x_1, y_1, z_1) -coordinate system, becomes

$$\hat{\mathbf{n}}_T = (\cos \beta_{\hat{\mathbf{n}}_T}, \quad 0, \quad \sin \beta_{\hat{\mathbf{n}}_T}). \quad (4.38)$$

As we did in the limb calculation, we can define the following coordinate transformation which transforms any point of (x_1, y_1, z_1) -coordinate into a point (x'_1, y'_1, z'_1) -coordinate such that

$$\begin{bmatrix} x'_1 \\ y'_1 \\ z'_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\sin \beta_{\hat{\mathbf{n}}_T} & 0 & \cos \beta_{\hat{\mathbf{n}}_T} \\ \cos \beta_{\hat{\mathbf{n}}_T} & 0 & \sin \beta_{\hat{\mathbf{n}}_T} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}. \quad (4.39)$$

Same to the limb calculation, it transforms the plane of the terminator Eq.(3.38), which is in the (x_1, y_1, z_1) -coordinate, into the (x'_1, y'_1) -plane, since we transformed it to get $z'_1 = 0$. The inverse coordinate transformation is also defined by the transpose of the matrix

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} 0 & -\sin \beta_{\hat{\mathbf{n}}_T} & \cos \beta_{\hat{\mathbf{n}}_T} \\ 1 & 0 & 0 \\ 0 & \cos \beta_{\hat{\mathbf{n}}_T} & \sin \beta_{\hat{\mathbf{n}}_T} \end{bmatrix} \begin{bmatrix} x'_1 \\ y'_1 \\ z'_1 \end{bmatrix}. \quad (4.40)$$

Let us transform the terminator defined by Eqs.(3.34-3.35) using the inverse transformation Eq.(3.40). Assuming the ellipsoid is not degenerated ($e \neq 1$), in the same manner we did in the limb calculation section, Eqs.(3.34-3.35) become to the equation of an ellipse in the $(x'_1, y'_1, z'_1 = 0)$ -plane with

$$\frac{x_1'^2}{r_e^2} + \frac{y_1'^2}{r_e^2(1 - e_{\beta_{\text{sun}}}^2)} = 1. \quad (4.41)$$

The projected ellipse has its centre at $(0, 0)$, semi-major axis $r_e \mathbf{i}'_1$, and semi-minor axis $r_e \sqrt{1 - e_{\beta_{\text{sun}}}^2} \mathbf{j}'_1$, where $e_{\beta_{\text{sun}}}$ is given by Eq.(3.15) replacing β_{obs} with β_{sun} ,

$$e_{\beta_{\text{sun}}}^2 = e^2 \left(1 - \frac{\sin^2 \beta_{\text{sun}}}{1 - e^2 \cos^2 \beta_{\text{sun}}} \right). \quad (4.42)$$

In the same manner of the limb calculation, here we want to introduce the basis vectors $(\mathbf{i}'_1, \mathbf{j}'_1)$ of the (x'_1, y'_1) -plane for transform the terminator into (x, y, z) -coordinate system are expressed by

$$\begin{aligned} \mathbf{i}'_1 &= (-\sin \Delta\lambda, \quad \cos \Delta\lambda, \quad 0), \\ \mathbf{j}'_1 &= (-\sin \beta_{\hat{n}_T} \cos \Delta\lambda, \quad -\sin \beta_{\hat{n}_T} \sin \Delta\lambda, \quad \cos \beta_{\hat{n}_T}). \end{aligned}$$

In the (x, y, z) -plane, *the sky plane* has vector basis given by Eqs.(3.19-20) and therefore projecting the basis $(\mathbf{i}'_1, \mathbf{j}'_1)$ onto the sky basis, as previously we did in Eqs.(3.22-23) taking into account Eq.(3.37)

$$\begin{aligned} \mathbf{i}'_1 \cdot \mathbf{i}_s &= \cos \Delta\lambda \\ \mathbf{j}'_1 \cdot \mathbf{i}_s &= -\sin \beta_{\hat{n}_T} \sin \Delta\lambda \\ \mathbf{i}'_1 \cdot \mathbf{j}_s &= \sin \beta_{\text{obs}} \sin \Delta\lambda \\ \mathbf{j}'_1 \cdot \mathbf{j}_s &= \sin \beta_{\hat{n}_T} \sin \beta_{\text{obs}} \cos \Delta\lambda + \cos \beta_{\hat{n}_T} \cos \beta_{\text{obs}} \end{aligned}$$

$$\mathbf{i}_s = \cos \Delta\lambda \mathbf{i}'_1 - \sin \beta_{\hat{n}_T} \sin \Delta\lambda \mathbf{j}'_1, \quad (4.43)$$

$$\mathbf{j}_s = \sin \beta_{\text{obs}} \sin \Delta\lambda \mathbf{i}'_1 + (\sin \beta_{\hat{n}_T} \sin \beta_{\text{obs}} \cos \Delta\lambda + \cos \beta_{\text{obs}} \cos \beta_{\hat{n}_T}) \mathbf{j}'_1. \quad (4.44)$$

We can write the projector from (x'_1, y'_1) onto (x_s, y_s) as

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix} = A \begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix}, \quad (4.45)$$

where

$$a = \cos \Delta\lambda, \quad (4.46)$$

$$b = -\sin \beta_{\hat{n}_T} \sin \Delta\lambda, \quad (4.47)$$

$$c = \sin \beta_{\text{obs}} \sin \Delta\lambda, \quad (4.48)$$

$$d = \sin \beta_{\text{obs}} \sin \beta_{\hat{n}_T} \cos \Delta\lambda + \cos \beta_{\text{obs}} \cos \beta_{\hat{n}_T}, \quad (4.49)$$

$$A = \begin{bmatrix} \cos \Delta\lambda & -\sin \beta_{\hat{n}_T} \sin \Delta\lambda \\ \sin \beta_{\text{obs}} \sin \Delta\lambda & \sin \beta_{\text{obs}} \sin \beta_{\hat{n}_T} \cos \Delta\lambda + \cos \beta_{\text{obs}} \cos \beta_{\hat{n}_T} \end{bmatrix}, \quad (4.50)$$

and inversely

$$\begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} x_s \\ y_s \end{bmatrix}, \quad (4.51)$$

and

$$\begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix} = \frac{1}{ad - bc} \begin{bmatrix} \sin \beta_{\text{obs}} \sin \beta_{\hat{n}_T} \cos \Delta\lambda + \cos \beta_{\text{obs}} \cos \beta_{\hat{n}_T} & \sin \beta_{\hat{n}_T} \sin \Delta\lambda \\ -\sin \beta_{\text{obs}} \sin \Delta\lambda & \cos \Delta\lambda \end{bmatrix} \begin{bmatrix} x_s \\ y_s \end{bmatrix}, \quad (4.52)$$

with

$$\begin{aligned} ad - bc &= \sin \beta_{\text{obs}} \sin \beta_{\hat{n}_T} \cos^2 \Delta\lambda + \cos \beta_{\hat{n}_T} \cos \beta_{\text{obs}} \cos \Delta\lambda + \sin \beta_{\text{obs}} \sin \beta_{\hat{n}_T} \sin^2 \Delta\lambda \\ &= \sin \beta_{\text{obs}} \sin \beta_{\hat{n}_T} + \cos \beta_{\text{obs}} \cos \beta_{\hat{n}_T} \cos \Delta\lambda. \end{aligned} \quad (4.53)$$

4.4 Cusp points and terminator determinant

Cusp points are the points where the plane of the limb and the plane of the terminator are identical, which means $\delta_{\text{obs}} \times \delta_{\text{sun}} = 0$. Otherwise, since *the sky plane* seen from the observer and *the sky plane* seen from the sun are each perpendicular to observing direction δ_{obs} and illumination source direction δ_{sun} . So the vector $\delta_{\text{obs}} \times \delta_{\text{sun}}$ can be interpreted as the direction of the line defined as the intersection of *the sky plane* seen from the observer and *the sky plane* seen from the sun. Therefore the line with direction $\delta_{\text{obs}} \times \delta_{\text{sun}}$ intersects the limb projection and the terminator projection at two points, these points are called the *cusp points*. In the (x, y, z) -coordinate system, $\delta_{\text{obs}} \times \delta_{\text{sun}}$ is given by calculating Eq.(3.6) and Eq.(3.33), we have

$$\begin{aligned} \delta_{\text{obs}} \times \delta_{\text{sun}} &= \begin{bmatrix} \cos \beta_{\text{obs}} \\ 0 \\ \sin \beta_{\text{obs}} \end{bmatrix} \times \begin{bmatrix} \cos \beta_{\text{sun}} \cos \Delta\lambda \\ \cos \beta_{\text{sun}} \sin \Delta\lambda \\ \sin \beta_{\text{sun}} \end{bmatrix} \\ &= \begin{bmatrix} -\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \sin \Delta\lambda \\ \sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda - \cos \beta_{\text{obs}} \sin \beta_{\text{sun}} \\ \cos \beta_{\text{obs}} \cos \beta_{\text{sun}} \sin \Delta\lambda \end{bmatrix} \end{aligned} \quad (4.54)$$

We define the rotation that transforms any point of (x, y, z) -coordinate into (x_s, y_s, z_s) -coordinate system

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -\sin \beta_{\text{obs}} & 0 & \cos \beta_{\text{obs}} \\ \cos \beta_{\text{obs}} & 0 & \sin \beta_{\text{obs}} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (4.55)$$

where (x_s, y_s) (since $z_s = 0$) are points in *the sky plane*. Let us transform the vector $\delta_{\text{obs}} \times \delta_{\text{sun}}$ into (x_s, y_s, z_s) -coordinate by using Eqs.(3.54-55).

$$x_s = \sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda - \cos \beta_{\text{obs}} \sin \beta_{\text{sun}},$$

$$\begin{aligned} y_s &= \sin^2 \beta_{\text{obs}} \cos \beta_{\text{sun}} \sin \Delta\lambda + \cos^2 \beta_{\text{obs}} \cos \beta_{\text{sun}} \sin \Delta\lambda \\ &= \cos \beta_{\text{sun}} \sin \Delta\lambda, \end{aligned}$$

$$z_s = -\cos \beta_{\text{obs}} \sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \sin \Delta\lambda + \cos \beta_{\text{obs}} \sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \sin \Delta\lambda = 0,$$

Finally, we get $\boldsymbol{\delta}_{\text{obs}} \times \boldsymbol{\delta}_{\text{sun}}$ in the (x_s, y_s, z_s) -coordinate as follows

$$\boldsymbol{\delta}_{\text{obs}} \times \boldsymbol{\delta}_{\text{sun}} = (\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda - \cos \beta_{\text{obs}} \sin \beta_{\text{sun}}, \quad \cos \beta_{\text{sun}} \sin \Delta\lambda, \quad 0), \quad (4.56)$$

which as expected is lying on *the sky plane*, i.e. $z_s = 0$.

The inverse rotation is written

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 & -\sin \beta_{\text{obs}} & \cos \beta_{\text{obs}} \\ 1 & 0 & 0 \\ 0 & \cos \beta_{\text{obs}} & \sin \beta_{\text{obs}} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}. \quad (4.57)$$

The feature of the intersection of the two *sky planes*, also give the major axis of the ellipse formed by the terminator. Let us define the vector of *the sky plane* $\mathbf{u} = (u, v)$ which is co-linear to $\boldsymbol{\delta}_{\text{obs}} \times \boldsymbol{\delta}_{\text{sun}}$ in (x_s, y_s, z_s) -coordinate,

$$\begin{cases} u &= \sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda - \sin \beta_{\text{sun}} \cos \beta_{\text{obs}} \\ v &= \cos \beta_{\text{sun}} \sin \Delta\lambda \end{cases}, \quad (4.58)$$

and the parametric form of the line containing the major axis of the ellipse is given by

$$\begin{cases} x_s &= ut \\ y_s &= vt \end{cases}. \quad (4.59)$$

The *cusps* are the two points of the end of this line that intersect the projection of the limb in *the sky plane* given by Eq.(3.29), cusp points of the line with parameter $t = t_1$. Let us do calculation, insert $(x_s = ut_1, y_s = vt_1)$ into Eq.(3.29)

$$\begin{aligned} \frac{u^2 t_1^2}{r_e^2} + \frac{v^2 t_1^2}{r_e^2 (1 - e_L^2)} &= 1 \\ t_1^2 (u^2 (1 - e_L^2) + v^2) &= r_e^2 (1 - e_L^2) \end{aligned}$$

then with Eq.(3.28), this becomes

$$t_1^2 = \frac{r_e^2 (1 - e^2 \cos^2 \beta_{\text{obs}})}{u^2 (1 - e^2 \cos^2 \beta_{\text{obs}}) + v^2}. \quad (4.60)$$

Meanwhile the ellipse formed by the terminator is a tilted ellipse with tilt angle θ_T with respect to the x_s -axis which expressed as

$$\theta_T = \tan^{-1} \left(\frac{v}{u} \right) = \left(\frac{\cos \beta_{\text{sun}} \sin \Delta\lambda}{\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda - \sin \beta_{\text{sun}} \cos \beta_{\text{obs}}} \right). \quad (4.61)$$

Similarly, naturally the minor axis is lying on the straight line with direction \mathbf{u}_\perp orthogonal to \mathbf{u} , the vector is given $\mathbf{u}_\perp = (v, -u)$, in *the sky plane*, (x_s, y_s) -coordinate. Projecting back the line $\mathbf{u}_\perp = (v, -u)$, onto the (x'_1, y'_1) -plane with

Eq.(3.53). Let us do calculation using Eq.(3.53) and $(x_s = vt, y_s = -ut)$, we have

$$\begin{aligned} \begin{bmatrix} x'_1 \\ y'_1 \end{bmatrix} &= \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} vt \\ -ut \end{bmatrix} \\ &= \frac{1}{ad - bc} \begin{bmatrix} dvt + but \\ -(cvt + aut) \end{bmatrix} \\ &= \frac{t}{ad - bc} \begin{bmatrix} dv + bu \\ -(cv + au) \end{bmatrix} \end{aligned}$$

so we can write (x'_1, y'_1)

$$x'_1 = t \frac{dv + bu}{ad - bc}, \quad (4.62)$$

$$y'_1 = -t \frac{cv + au}{ad - bc}. \quad (4.63)$$

Substituting (x'_1, y'_1) into the equation of the terminator Eq.(3.41) with $t = t_2$

$$\begin{aligned} \frac{x'^2_1}{r_e^2} + \frac{y'^2_1}{r_e^2(1 - e^2_{\beta_{\text{sun}}})} &= 1 \\ \frac{t_2^2}{(ad - bc)^2 r_e^2} \left((dv + bu)^2 + \frac{(cv + au)^2}{(1 - e^2_{\beta_{\text{sun}}})} \right) &= 1 \\ t_2^2 \left((dv + bu)^2 (1 - e^2_{\beta_{\text{sun}}}) + (cv + au)^2 \right) &= (ad - bc)^2 r_e^2 (1 - e^2_{\beta_{\text{sun}}}) \\ t_2^2 &= \frac{r_e^2 (1 - e^2_{\beta_{\text{sun}}}) (ad - bc)^2}{(dv + bu)^2 (1 - e^2_{\beta_{\text{sun}}}) + (cv + au)^2} \end{aligned} \quad (4.64)$$

Which is the intersection between the ellipse of the terminator and the line with direction \mathbf{u}_\perp . Then the semi-major and semi-minor axes of the projection of the terminator onto *the sky plane* are expressed as follows

$$a_T^2 = (u^2 + v^2)t_1^2, \quad (4.65)$$

$$b_T^2 = (u^2 + v^2)t_2^2. \quad (4.66)$$

With the tilt angle θ_T , and the semi-major and semi-minor axes Eqs.(3.65-66), the projection of the terminator can be written in parametric form in *the sky plane* as

$$\begin{cases} x_T &= a_T \cos \theta_T \cos \theta + b_T \sin \theta_T \sin \theta \\ y_T &= -a_T \sin \theta_T \cos \theta + b_T \cos \theta_T \sin \theta \end{cases}, \quad (4.67)$$

where θ is an oriented angle measured from the x_s -axis.

4.5 Find visible and hidden terminator

In order to find the visible (bright) and hidden (dark) terminator, we can project any point of the terminator and limb, along the direction perpendicular to the line joining the two cusps, and can check the orientation of the projection of the sun. The direction of the sun in *the sky plane*, $(x_s, y_s, z_s = 0)$ -coordinate, δ_{sun_s} , is calculated by Eq.(3.55) and Eq.(3.33) with $z_s = 0$,

$$\begin{aligned} \delta_{\text{sun}_s} &= \begin{bmatrix} 0 & 1 & 0 \\ -\sin \beta_{\text{obs}} & 0 & \cos \beta_{\text{obs}} \\ \cos \beta_{\text{obs}} & 0 & \sin \beta_{\text{obs}} \end{bmatrix} \begin{bmatrix} \cos \beta_{\text{sun}} \cos \Delta\lambda \\ \cos \beta_{\text{sun}} \sin \Delta\lambda \\ \sin \beta_{\text{sun}} \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta_{\text{sun}} \sin \Delta\lambda \\ -\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda + \cos \beta_{\text{obs}} \sin \beta_{\text{sun}} \\ 0 \end{bmatrix}, \end{aligned}$$

so we have δ_{sun_s} in reduced dimension (x_s, y_s) written as

$$\delta_{\text{sun}_s} = (\cos \beta_{\text{sun}} \sin \Delta\lambda, \quad -\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda + \cos \beta_{\text{obs}} \sin \beta_{\text{sun}}). \quad (4.68)$$

Let us consider any point $P(x_s, y_s)$ of the projection of the terminator in *the sky plane* and the direction $\mathbf{u}_\perp = (v, -u)$ orthogonal to \mathbf{u} . We can construct the algebraic measure

$$D = (\delta_{\text{sun}_s} \cdot \mathbf{u}_\perp) (OP \cdot \mathbf{u}_\perp), \quad (4.69)$$

or written with details of the components as follows

$$\begin{aligned} D &= \left((\cos \beta_{\text{sun}} \sin \Delta\lambda)^2 + (\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda - \cos \beta_{\text{obs}} \sin \beta_{\text{sun}})^2 \right) \\ &\quad (\cos \beta_{\text{sun}} \sin \Delta\lambda x_s + (-\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda + \cos \beta_{\text{obs}} \sin \beta_{\text{sun}}) y_s), \end{aligned} \quad (4.70)$$

then check for its sign. $D > 0$ means visible terminator, which is the bright limb, $D < 0$ means hidden terminator, which is the dark limb, $D = 0$ means cusp point. In addition, that the sign of D is the same as the same of the quantity of D' .

$$D' = \cos \beta_{\text{sun}} \sin \Delta\lambda x_s + (-\sin \beta_{\text{obs}} \cos \beta_{\text{sun}} \cos \Delta\lambda + \cos \beta_{\text{obs}} \sin \beta_{\text{sun}}) y_s. \quad (4.71)$$

Any point $P(x_s, y_s)$ (since $z_s = 0$) of the projection of the terminator in *the sky plane* has the following coordinates in the (x, y, z) system. Let us do calculation using Eq.(3.57)

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= \begin{bmatrix} 0 & -\sin \beta_{\text{obs}} & \cos \beta_{\text{obs}} \\ 1 & 0 & 0 \\ 0 & \cos \beta_{\text{obs}} & \sin \beta_{\text{obs}} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s = 0 \end{bmatrix} \\ &= \begin{bmatrix} -y_s \sin \beta_{\text{obs}} \\ x_s \\ y_s \cos \beta_{\text{obs}} \end{bmatrix} \end{aligned} \quad (4.72)$$

then the normal to the terminator \hat{n}_T can be also transformed into (x, y, z) system which is calculated by using Eq.(3.31) and Eq.(3.38)

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= \begin{bmatrix} \cos \Delta\lambda & -\sin \Delta\lambda & 0 \\ \sin \Delta\lambda & \cos \Delta\lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \beta_{\hat{n}_T} \\ 0 \\ \sin \beta_{\hat{n}_T} \end{bmatrix} \\ &= \begin{bmatrix} \cos \beta_{\hat{n}_T} \cos \Delta\lambda \\ \cos \beta_{\hat{n}_T} \sin \Delta\lambda \\ \sin \beta_{\hat{n}_T} \end{bmatrix} \end{aligned} \quad (4.73)$$

The signed distance between P and the plane containing the terminator is given by calculating $D = \mathbf{OP} \cdot \hat{n}_T$ with Eqs.(3.72-73)

$$\begin{aligned} \mathbf{OP} \cdot \hat{n}_T &= \begin{bmatrix} -y_s \sin \beta_{\text{obs}} \\ x_s \\ y_s \cos \beta_{\text{obs}} \end{bmatrix} \begin{bmatrix} \cos \beta_{\hat{n}_T} \cos \Delta\lambda \\ \cos \beta_{\hat{n}_T} \sin \Delta\lambda \\ \sin \beta_{\hat{n}_T} \end{bmatrix} \\ &= -y_s \cos \beta_{\hat{n}_T} \sin \beta_{\text{obs}} \cos \Delta\lambda + x_s \cos \beta_{\hat{n}_T} \sin \Delta\lambda + y_s \sin \beta_{\hat{n}_T} \cos \beta_{\text{obs}} \\ &= x_s \cos \beta_{\hat{n}_T} \sin \Delta\lambda + y_s (-\cos \beta_{\hat{n}_T} \sin \beta_{\text{obs}} \cos \Delta\lambda + \sin \beta_{\hat{n}_T} \cos \beta_{\text{obs}}). \end{aligned} \quad (4.74)$$

and points such that $D > 0$ belong to the visible terminator while points with $D < 0$ are hidden, and $D = 0$ are cusp points.

/5

Appendix 2: Voronoi diagram and VOISE algorithm

5.1 Voronoi diagram

The shape of Voronoi diagram (VD) evokes fractals and fibonacci spirals, which turns up frequently in the natural world. The first VD concept was established in 17th century by Peter Gustav Lejeune Dirichlet and Georgy Feodosevich Voronoy. VD is a collection of boundary points in the N dimensions space which divided into a set of disjoint convex polygons called Voronoi Region (VR). Each VR is generated by a corresponding point which called seeds, sometimes germs, or generators. Let us start with considering a set of seeds $S = \{s_1, s_2, \dots, s_n\}$, where s_i are distinct points in the plane \mathbb{R}^2 . The VR generated by s_i , written in $\mathcal{R}(s_i)$, is the set of points \mathbf{p} in the plane \mathbb{R}^2 closer to s_i than other seeds of s_j . As regards Euclidean distance d the formal form of $\mathcal{R}(s_i)$ is given by

$$\mathcal{R}(s_i) = \{\mathbf{p} \in \mathbb{R}^2 \mid d(\mathbf{p}, s_i) < d(\mathbf{p}, s_j), \quad s_j \in S \setminus \{s_i\}\}. \quad (5.1)$$

We can also express its *closure*, written in $\overline{\mathcal{R}}(s_i)$, of the $\mathcal{R}(s_i)$ defined by

$$\overline{\mathcal{R}}(s_i) = \{\mathbf{p} \in \mathbb{R}^2 \mid d(\mathbf{p}, s_i) \leq d(\mathbf{p}, s_j), \quad s_j \in S \setminus \{s_i\}\}. \quad (5.2)$$

The VD formed by a set of seeds S , written in $\mathcal{V}(S)$, which consists of the boundary points of a set of VRs. So VD shapes graph or 'skeleton' form by means of the

boundaries of VRs, which is defined by

$$\mathcal{V}(S) = \bigcap_{s_i \in S} \overline{\mathcal{R}(s_i)}. \quad (5.3)$$

The geometrical representation of VD is seen its set of boundary lines associating VRs. Fig 4.1 illustrates a VD for a set of 6 seeds in \mathbb{R}^2 . In mathematical termi-

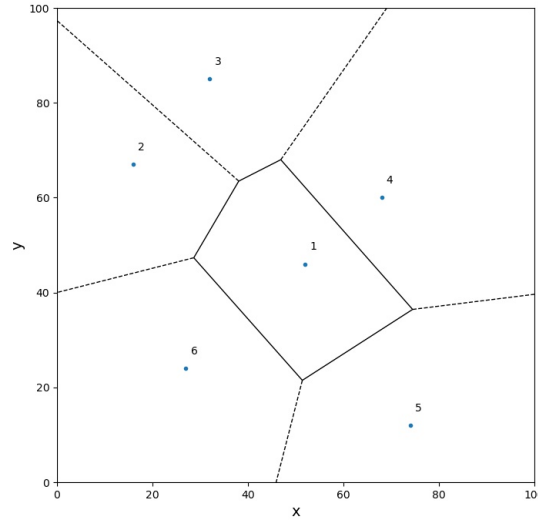


Figure 5.1: An example of Voronoi diagram. Dots are seeds, dashed lines and solid lines are bisector between associate seeds. Each region with a point separated by the lines is VR. VD appears with a form of dashed and solid lines.

nology, a VR $\mathcal{R}(s_i)$ generated by a seed s_i is called the *influence zone* or the *dominance region* of the seed s_i . The whole picture of the VRs is seen like as a *tessellation* of polygons of $\mathcal{R}(s_i)$ and VD $\mathcal{V}(S)$ forms the *SKelton by Influence Zones* which abbreviated into SKIZ.

5.2 Delaunay triangulation

In addition, we see another representation of the VD in graph theory. The Voronoi graph, $G(S) = (V, E)$, is a planar graph in two-dimensional space, which represents the topological structure of the neighbouring VRs. Fig 4.2 shows the Voronoi graph into Fig 4.1. In Fig 4.2, the vertices or nodes, V , are seen as blue dots and the line connecting two neighbour vertices is called edge written in E of the Voronoi graph G . The neighbour seeds of a seed s_1 are $\mathcal{N}(s_1) = \{s_2, s_3, s_4, s_5, s_6\}$ in the given set of seeds S . The links of Voronoi graph structure associated by a seed s_1 are

$\{s_1, s_2\}$, $\{s_1, s_3\}$, $\{s_1, s_4\}$, $\{s_1, s_5\}$, and $\{s_1, s_6\}$. As a dual of Voronoi graph, there is Delaunay triangulation (DT) which appears by connecting all the pairs of neighbour seeds. DT is the unique triangulation, which elements of one DT share one particular circumcircle. The circumcircle does not contain other vertices in/on it. Therefore, the DT optimizes the geometric 'compactness' of the triangulation which means DT maximises the minimum angles of the triangle.

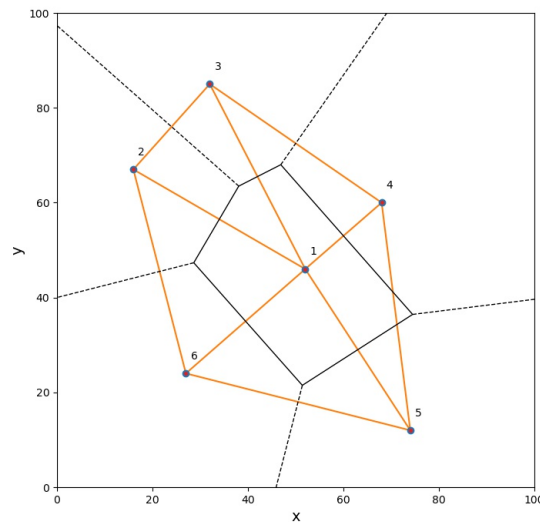


Figure 5.2: An example of the appearance of Voronoi graph and the Delaunay triangulations.

5.3 VOISE algorithm

VD has been studied in many image-processing applications, including interpolation, compression, pattern recognition, texture analysis [17], and object representation, require partitioning the support of the images [14]. VOISE algorithm [6] is a dynamic and self-organising algorithm to create a partition of an image pixel into VD according to prescribed homogeneity criterion. The created VD is updated in iterative way by adding and removing the seeds in the image plane. However, updating VD for the whole plane would be burden on computation. To be more efficient algorithm, an incremental algorithm was chosen. In addition, there is a requirement VD to be discrete because of their degeneracy. As long as the three elements of DT belong to one circumcircle, the DT is a unique triangulation which satisfies the discreteness of VD. However, if there are seeds more than 4 sharing a circumcircle, the DT would not be unique, where VD is in degenerate condition. For a successful

algorithm, VOISE was developed using *discrete version of SKIZ algorithm*[14] which can deal with the degeneracy due to the cocircular seeds.

The characterization of the image with VD depends on the location of the seeds which are one per polygon. The areas of the image with greater gradient in pixel intensities are characterised by smaller polygons and uniform image intensities have larger polygons. The emergence of the seeds on the image plane, the emphasized features of the target image, help us to identify the lines and curves of the planetary auroral images with less noise. Here after, introduce four phases in the VOISE algorithm, which are initialisation phase, dividing phase, merging phase, and regularisation phase.

5.4 Initialization phase

In this phase the seeds for the VDs are initially distributed in the finite image plane Ω in a random way according to the uniform probability distribution which is known as a binomial point process. The information of the seed s_i and its VRs are an unique value depending on the region $\mathcal{R}(s_i)$ and the intensity of the image pixels within that region. This information defines the initial tessellation in the image plane and be used in the subsequent phases.

5.5 Dividing phase

The dividing phase is an iterative process and is the most important phase. After the initialization phase, each VRs extracts the homogeneous of the pixel intensity in it. The predefined homogeneity criteria is applied to the region. Untile the VR fills the criteria the iteration continues and the tessellation in the image plane develops. To get the non-homogeneous region satisfy the homogeneity criteria, the homogeneity merit function $\chi(\mathcal{I}, s)$ is introduced. The homogeneity merit function of Voronoi region associated seed s_i is defined:

$$\chi(\mathcal{I}, s_i) = \frac{\max_{\mathbf{p} \in \mathcal{R}(s_i)} [\mathcal{I}(\mathbf{p})] - \min_{\mathbf{p} \in \mathcal{R}(s_i)} [\mathcal{I}(\mathbf{p})]}{\|\chi(\mathcal{I}, S)\|} \quad (5.4)$$

$$\|\chi(\mathcal{I}, S)\| = \max_{s_i \in S} \left\{ \max_{\mathbf{p} \in \mathcal{R}(s_i)} [\mathcal{I}(\mathbf{p})] - \min_{\mathbf{p} \in \mathcal{R}(s_i)} [\mathcal{I}(\mathbf{p})] \right\} \quad (5.5)$$

Where \mathcal{I} is the image pixels within the $\mathcal{R}(s_i)$ and $\mathcal{I}(\mathbf{p})$ is the image intensity at point \mathbf{p} . The denominator, Eq.(4.5), is the largest value of the numerator over all

VRs, so that $\chi(\mathcal{I}, \mathbf{s})$ takes its value in the bound of $[0, 1] \forall \mathbf{s} \in S$. The similarity is seen its low value and dissimilarity is seen its closeness to 1. We check the homogeneity to over all VRs with merit function and the predefined criteria χ_m , if it is below the χ_m such that $\chi(\mathcal{I}, \mathbf{s}_i) \geq \chi_m$. If $\chi(\mathcal{I}, \mathbf{s}_i)$ is not below the threshold, by adding seeds to the $\mathcal{R}(\mathbf{s}_i)$ and recreate the Voronoi diagram. The number of seeds added corresponds to the number of vertices in the VD. Each added seed is on the extension line passing the corresponding single vertex and the corresponding seed locating in the inhomogeneous VR. The weight parameters, w_s corresponding to the seeds and w_v corresponding to the vertices of the diagram, greatly affect the location of added seeds and the shape of VD. Those weight parameters also correspond to a fixed relative distance from the seed. The most probable number of vertices/edges of a typical cell of a Poisson-Voronoi diagram is six[12] VOISE author found a good compromise value for $w_v = \frac{3}{8}$ and for $w_s = \frac{5}{8}$ between quick convergence and comparable size of new region. Fig 4.3 shows how the effect of the weight parameters.[6] In each iteration, homogeneity threshold χ_m is calculated

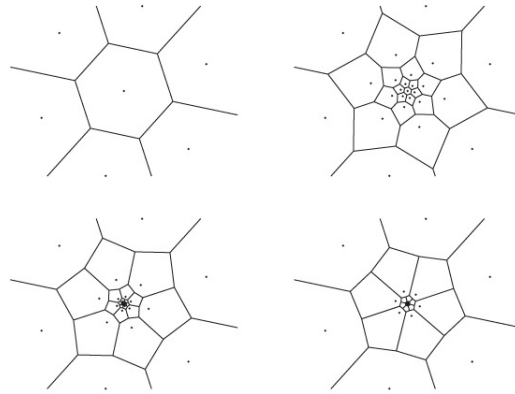


Figure 5.3: (from [6]) The image of how the weights effect on seed placement. Seeds are added to the central hexagonal region (upper-left panel) recursively for three iterations, adding 18 seeds. The weights are $w_s = \frac{1}{4}$, $w_v = \frac{3}{4}$ (upper-right), $w_s = \frac{1}{2}$, $w_v = \frac{1}{2}$ (lower-left), $w_s = \frac{3}{4}$, $w_v = \frac{1}{4}$ (lower-right),

using statistics on the sample χ_i of the measure of homogeneity,

$$\chi_i = \{\chi(\mathcal{I}, \mathbf{s}_i), \mathbf{s}_i \in S\}. \quad (5.6)$$

χ_m is defined as the value of the homogeneity measure for which a fraction $\frac{pD}{100}$ of the polygons are homogeneous so that $\chi(\mathcal{I}, \mathbf{s}_i) < \chi_m$. Defining the probability Prob we can write the following:

$$\text{Prob} [\chi(\mathcal{I}, \mathbf{s}_i) < \chi_m] = \frac{pD}{100} \quad (5.7)$$

During the dividing phase, the region with higher measure of homogeneity $\chi(\mathcal{I}, \mathbf{s}_i)$ is split into regions which have smaller measure of homogeneity than $\chi(\mathcal{I}, \mathbf{s}_i)$. As the dividing iteration goes on, the Eq.(4.5) decreases which results in the increase of the strictness of the criteria χ_m . The dividing iteration continues until all VRs are classified as homogeneous, or the size of the polygon does not allow further addition of new seeds. The information of the minimum distance d_m between two seeds is also checked by VOISE. For efficient operation of the dividing phase to the most contrasted region and not to low-contrasted region, the author of the VOIS presented a large percentile value of $p_D = (70 - 80) \%$ with a median merit function $\chi(\mathcal{I}, \mathbf{s}_i)$. Otherwise, for the smaller region detection or more noisy features, the percentile of $p_D = 60 - 70 \%$ is also suggested but not for the most contrasted features. The minimum distance d_m is the parameter which limits the length scale of the features we want to detect. So d_m needs to be carefully chosen by taking into account the size of the smallest resolvable features.

5.6 Merging phase

When VRs becomes homogeneous with respect to the threshold χ_m or its shape does not allow adding new seed in, the merging phase starts. This phase is iterative process consists of removing unnecessary ‘fine’ regions where adjacent VRs have very similar characteristics. The unnecessary VRs have undesirable effect of fragmenting regions. In order to reduce the bad effect, redundant seeds are removed with the following conditions. The averaged intensity μ_i of the image for the region $\mathcal{R}(\mathbf{s}_i)$ associated with a seed \mathbf{s}_i . A seed \mathbf{s}_i is defined as redundant and ought to be removed when the following conditions are fulfilled.

- (a) The region $\mathcal{R}(\mathbf{s}_i)$ is homogeneous with respect to the threshold χ_m .

$$\chi(\mathcal{I}, \mathbf{s}_i) < \chi_m \quad (5.8)$$

where

$$\text{Prob}[\chi(\mathcal{I}, \mathbf{s}_i) < \chi_m] = \frac{P_M}{100}. \quad (5.9)$$

P_M is a parameter to be set in the VOISE configuration.

- (b) Among its neighbouring seeds $\mathbf{s}_j \in \mathcal{N}(\mathbf{s}_i)$ which are around in the homogeneous region, the relative difference between the averaged values μ_i and μ_j is less than the predefined threshold $\Delta\mu$

$$|\mu_i - \mu_j| < \Delta\mu |\mu_i|. \quad (5.10)$$

- (c) The total length \mathcal{L} of the edges shared with non-homogeneous neighbours normalized to the perimeter \mathcal{P} of the region does not exceed a prescribed threshold $\Delta\mathcal{H}$

$$\frac{\mathcal{L}}{\mathcal{P}} < \Delta\mathcal{H} \quad (5.11)$$

At each iteration all seeds are checked. If all three conditions are satisfied, the seed is removed and the VD is updated. The neighbour cells expand to include the pixels previously allocated to that seed. Because $\chi(\mathcal{I}, \mathbf{s}_i)$ can only increase with the removal of seeds, condition (a) becomes more strict in the iteration and so the merging phase ends when no more regions meet all three conditions. A large percentile of p_M leads to less merging so the more seeds and VRs remain. A smaller value of p_M leads to more merging so the number of seeds and VRs become small.

5.7 Regularization phase

Usually, the VRs after the second and third phases are highly irregular seen by many acute angles and elongated polygons. So the VRs are regularized in this phase in iterative process. This process is to calculate a new VD where each seed \mathbf{s}_i is replaced by the center of the mass $\xi_i(\mathbf{s}_i)$ of the VR;

$$\xi = \frac{\sum_{\mathbf{p} \in \mathcal{R}(\mathbf{s}_i)} \mathbf{p} \rho(\mathbf{p})}{\sum_{\mathbf{p} \in \mathcal{R}(\mathbf{s}_i)} \rho(\mathbf{p})} \quad (5.12)$$

Where the ρ is a pixel intensity density function at pixel \mathbf{p} , which is close to uniform within each VR after the second and third phases. This iterative process stops when the difference between seed position \mathbf{s}_i and $\xi_i(\mathbf{s}_i)$ is less than 1 for all $\mathbf{s}_i \in S$. Because the center of mass and seeds are only defined in the image plane Ω , the 1 pixel limit is enough. And also when the iteration reached to the predefined maximum iteration, the process can be terminated. After regularisation, the process is complete.

/6

Appendix 3: Minnaert model Matlab code

```
1 function imgsub = minnvince(params,img,se,lat,lon,oza,sza)
2
3 % INPUT
4 % img : image data
5 % se : sub-Earth point on the target planet for the ...
   calculation of linear
6 %     weight with longitude to the 2 models calculation.
7 % lat : Latitude matrix corresponding to the image pixel
8 % lon : Longitude matrix corresponding to the image pixel
9 % oza : Observer zenith angle corresponding to the image pixel
10 % sza : Solar zenith angle corresponding to the image pixel
11 %
12 % OUTPUT
13 % imgsub : subtracted image
14 %
15 % (Vincent)  $\log(\mu \cdot I) = C_0 + C_1 \cdot x + C_2 \cdot x.^2 + C_3 \cdot x.^3;$ 
16 % (Left side // estimated image intensity)
17 % I :observed image intensity
18 % mu :cosine of the observation zenith angle
19 % mu0 :cosine of the solar zenith angle
20 % x :log(mu*mu0)
21 %
22 % Vincent: doi:10.1006/icar.1999.6232
23 % coefficients are derived by a least-squares fit to the ...
   intensity at a
```

```

24 % selected latitude range.
25 %
26
27 img = params.W;
28
29 % conversion factor
30 d2r = cspice_rpd; % degree to darian
31 r2d = cspice_dpr; % radian to degree
32
33 sza = sza*d2r; oza = oza*d2r;
34
35 % mu and mu0
36 mu = cos(oza); mu0 = cos(sza);
37
38 % exclude aurora emission
39 imgcalc = img; a_region=find(lat<-60 & lat≠-100);
40 a_intensity = isoutlier(imgcalc(a_region),'median');
41 imgcalc(a_region(a_intensity))=NaN; ...
    imgcalc=fillmissing(imgcalc,'linear');
42
43 ydata = log(imgcalc.*mu); ydata(isinf(ydata))=0; % -inf
44 x1=log(mu.*mu0);
45
46 model = zeros(size(img));
47
48 % latitude bin
49 latbin=linspace(-90,0,91);
50
51 for i = 1:length(latbin)
52     latid = find(lat>latbin(i) & lat≤latbin(i)+1);
53     poscos = latid(find(mu(latid)>0 & mu0(latid)>0));
54     % index sunlit-side longitude range with latitude bin
55     idsunlit = find(lat>latbin(i) & lat≤latbin(i)+1 & ...
        lon≥-85 & lon≤45);
56     % index terminator-side longitude range with latitude bin
57     idtermi = find(lat>latbin(i) & lat≤latbin(i)+1 & ...
        lon≥-45 & lon≤70);
58
59     % index negative cosines
60     negative_s = find(mu(idsunlit)<0);
61     negative_t = find(mu0(idtermi)<0);
62
63     if numel(idsunlit)≠0 && numel(idtermi)≠0
64         % pick up x1,ydata in the bin for generating ...
            fitting functions
65         X1=x1(poscos);
66
67         X1s = x1(idsunlit); Ydatas = ydata(idsunlit);
68         outs = ...
            excludedata(X1s(:),Ydatas(:),'indices',negative_s);
69         [X1s,Is] = sort(X1s); Ydatas=Ydatas(Is);
70         outs = outs | excludedata(X1s(:),Ydatas(:),'range', ...
            [-8 -0.01]);
71

```

```

72     X1t=x1(idtermi); Ydatat=ydata(idtermi);
73     outt = ...
       excludedata(X1t(:),Ydatat(:),'indices',negative_t);
74     [X1t,It] = sort(X1t); Ydatat=Ydatat(It);
75     outt = outt | excludedata(X1t(:),Ydatat(:),'range', ...
       [-8 -0.01]);
76
77     if length(Ydatas(~outs))>4 && ...
       length(Ydatat(~outt))>4 % check if there is the ...
       enough sample data
78         % generate fit function
79         warning off
80         [fs,~,~]=fit(X1s(:),Ydatas(:),'poly3','Exclude',outs,'Robust','on');
81         [ft,~,~]=fit(X1t(:),Ydatat(:),'poly3','Exclude',outt,'Robust','on');
82         warning on
83
84         if 1 % fit data plot
85             if latbin(i)== -86 || latbin(i)== -76 || ...
               latbin(i)== -66 || latbin(i)== -56 ...
86                 || latbin(i)== -46 || latbin(i)== -36 ...
                   || latbin(i)== -26
87                 X1s = X1s(~outs); Ydatas=Ydatas(~outs);
88                 X1t = X1t(~outt); Ydatat=Ydatat(~outt);
89                 figure;
90                 % plot(fs,X1s(:),Ydatas(:))
91                 plot(X1s(:),Ydatas(:),'bx','DisplayName','Data/sunlit')
92                 hold on
93                 % plot(ft,X1t(:),Ydatat(:))
94                 plot(X1s(:),feval(fs,X1s(:)),'rx','DisplayName','Fit/sunlit')
95                 plot(-X1t(:),Ydatat(:),'bo','DisplayName','Data/terminator')
96                 plot(-X1t(:),feval(ft,X1t(:)),'ro','DisplayName','Fit/terminator')
97                 hold off
98                 title(['Latitude bin from ' ...
                   num2str(latbin(i)) ' to ' ...
                   num2str(latbin(i)+1) ' [deg]'])
99                 legend('Location','southeast');
100                xlim([-6 6]);
101                ylim([-8 0]);
102                xlabel('x = Log(mu*mu0)')
103                ylabel('y = Log(I*mu)')
104            end
105        end
106
107        for j=1:length(poscos)
108            % linear unit weight wrt longitude to the ...
            set of coefficients
109            model(poscos(j)) = ...
                feval(ft,X1(j))*((90-se.lon+lon(poscos(j)))/180)+ ...
                ...
                feval(fs,X1(j))*(1-(90-se.lon+lon(poscos(j)))/180);
110
111            if model(poscos(j))>0 || model(poscos(j))==0
112                model(poscos(j))=0;
113            else
114                model(poscos(j))=exp(model(poscos(j)))/mu(poscos(j));

```

```
115         end
116     end
117 end
118 end
119 end
120
121
122 imgsub = img - model;
123 imgsub(imgsub<0) = 0;
124
125 end
```




Appendix 4: Li model Matlab code

```
1 function imgsub = li_dayglow(params,img,lat,lon,oza,sza,ss,se)
2
3 % INPUT
4 % img : image data
5 % se : sub-Earth point on the target planet
6 % ss : sub-Solar point on the target planet
7 % lat : Latitude matrix corresponding to the image pixel
8 % lon : Longitude matrix corresponding to the image pixel
9 % oza : Observer zenith angle corresponding to the image pixel
10 % sza : Solar zenith angle corresponding to the image pixel
11 %
12 % OUTPUT
13 % imgsub : subtracted image
14 %
15 % >(Li)  $I = A + B*\cos(SZA) + C*\cos(VZA)$ ; (1)
16 %
17 % I :estimated image intensity
18 % SZA :solar zenith angle
19 % VZA :observation zenith angle
20 % A,B,and C :fitting coefficients
21 %
22 % Li: DOI:10.1109/IGARSS.2004.1369944
23 % fit based on the equation (1) is used to estimate the ...
    dayglow intensity
24 % as function of  $\cos(SZA)$  and  $\cos(VZA)$ 
```

```

25 %
26
27 img = params.W;
28
29 % conversion factor
30 d2r = cspice_rpd; r2d = cspice_dpr;
31
32 sza = sza*d2r; oza = oza*d2r;
33
34 % cosines of zenith angles
35 x1 = cos(sza); % cosines of solar zenith angle
36 x2 = cos(oza); % cosines of observer zenith angle
37
38 lon = lon - (ss.lon-se.lon); % shift longitude mesh to get ...
    the noon at the center
39
40 % Magnetic latitude 'Mlat' and the Mean intensity values 'Mlatv'
41 Mlat=linspace(-90,-40,51);
42 Mlatnv=zeros(size(Mlat));Mlatdv=zeros(size(Mlat));
43 for i = 1:length(Mlat)
44     % data in latitude bin night side
45     Mlatidn=find(lat>=Mlat(i) & lat<=Mlat(i)+1 & lon>0);% ...
        nightside
46     Mlatidd=find(lat>=Mlat(i) & lat<=Mlat(i)+1 & lon<0);% dayside
47     Mlatnv(i)=mean(img(Mlatidn));% nightside
48     Mlatdv(i)=mean(img(Mlatidd));% dayside
49 end
50
51 % nightside peak
52 [peakvn, peakidn]=max(Mlatnv,[],'all','linear')
53 % nightside peak in Mlat
54 peakMlatn=Mlat(peakidn)
55
56 % nightside pole-ward minimum
57 [poleminn,poleminidn]=min(Mlatnv(Mlat<peakMlatn),[],'all','linear')
58 % nightside pole-ward
59 poleboundn=max(Mlat(find(Mlatnv(Mlat<peakMlatn)<0.8*(peakvn-poleminn))))
60
61 % nightside equator ward
62 [equminn,equidn]=min(Mlatnv(Mlat>peakMlatn),[],'all','linear')
63 equidn = equidn+peakidn-1;
64 % nightside find equator-ward boundary by the subjective ...
    criteria
65 Mlatequn = Mlat(Mlat>peakMlatn); Mlatvequn = ...
    Mlatnv(Mlat>peakMlatn);
66 equboundn=min(Mlatequn(Mlatvequn < 0.8*(peakvn-equminn)))
67
68
69 % dayside peak
70 [peakvd, peakidd]=max(Mlatdv,[],'all','linear')
71 % dayside peak location in Mlat
72 peakMlatd=Mlat(peakidd)
73
74 % dayside pole-ward minimum

```

```

75 [polemind,poleminidd]=min(Mlatdv(Mlat<peakMlatd),[],'all','linear')
76
77 % dayside pole-ward
78 poleboundd=max(Mlat(Mlatdv(Mlat<peakMlatd)<0.8*(peakvd-polemind)))
79
80 % dayside equator-ward
81 [equmind,equidd]=min(Mlatdv(Mlat>peakMlatd),[],'all','linear')
82 equidd = equidd+peakidd-1;
83 % dayside find equator-ward boundary by the subjective criteria
84 Mlatequd = Mlat(Mlat>peakMlatd); Mlatvequd = ...
    Mlatdv(Mlat>peakMlatd);
85 equboundd=min(Mlatequd(Mlatvequd<0.8*(peakvd-equmind)));
86 if isempty(equboundd)
87     equboundd = Mlat(equidd)
88 end
89
90 if poleminn > peakvn/2 | equminn > peakvn/2
91     poleboundn = Mlat(poleminidd);
92     equboundn = Mlat(equidd);
93 end
94
95
96 % night side index info for non auroral and auroral region ...
    in the image
97 equiregionn=find(lat>equboundn & lat≠-100);
98 poleregionn=find(lat<poleboundn & lat≠-100);
99 % dayside index info for non auroral and auroral region in ...
    the image
100 equiregiond=find(lat>equboundd & lat≠-100);
101 poleregiond=find(lat<poleboundd & lat≠-100);
102
103 non_auroran=vertcat(equiregionn,poleregionn); % non auroral ...
    region indexes
104 non_auroramix=vertcat(equiregiond,poleregionn); % non ...
    auroral region mix side indexes
105 auroran=find(lat>poleboundn & lat<equboundn); % auroral ...
    region indexes
106 aurorad=find(lat>poleboundd & lat<equboundd); % auroral ...
    region indexes
107 auroramix=find(lat>poleboundn & lat<equboundd); % auroral ...
    region mix side indexes
108
109
110 model = zeros(size(img)); latbin=linspace(-90,0,91);
111 for i=1:length(latbin)
112     if latbin(i)>equboundd || latbin(i)<poleboundn
113         % for non auroral region
114         latid1=find(lat>latbin(i) & lat<latbin(i)+1);
115         X1=[x1(latid1) x2(latid1)]; I=img(latid1);
116         flag=1;
117     else % for auroral region
118         latid1=find(lat>latbin(i) & lat<latbin(i)+1);
119         latid2=latid1(~isoutlier(img(latid1),'median')); % ...
            filter auroral intensity

```

```
120     X1=[x1(latid1) x2(latid1)];
121     X2=[x1(latid2) x2(latid2)]; I=img(latid2);
122     flag=0;
123 end
124 % pick up cos(sza) cos(oza)
125 if length(I(I≠0))>3 % check if there is the enough ...
    sample data
126     if flag
127         warning off
128         [fli,~,~]=fit(X1,I,ft);
129         warning on
130         for j=1:length(latid1)
131             % using cos(sza) and cos(oza) get estimated ...
                intensity by
132             % the generated fit function 'fli'
133             model(latid1(j)) = feval(fli,X1(j,:));
134         end
135     else
136         warning off
137         [fli,~,~]=fit(X2,I,ft);
138         warning on
139         for j=1:length(latid1)
140             % using cos(sza) and cos(oza) get estimated ...
                intensity by
141             % the generated fit function 'fli'
142             model(latid1(j)) = feval(fli,X1(j,:));
143         end
144     end
145 end
146 end
147 subimg = img-model; subimg(subimg<0) = 0;
148 end
```

Bibliography

- [1] Charles H. Acton. “Ancillary data services of NASA’s Navigation and Ancillary Information Facility.” In: *Planetary and Space Science*. Planetary data system 44.1 (Jan. 1, 1996), pp. 65–70. ISSN: 0032-0633. DOI: 10.1016/0032-0633(95)00107-7.
- [2] Bertrand Bonfond. “Morphology and dynamics of the Io UV footprint.” In: (Oct. 2009).
- [3] P. Guio, N. Achilleos, and R. Prange. “Auroral Feature Detection by Combined Image Segmentation and Clustering Analysis.” In: (2010).
- [4] P. Guio, N. Achilleos, and R. Prange. “Determining Disc Parameters in Planetary Images.” In: (2010).
- [5] P. Guio and N. Achilleos. “A new method for determining geometry of planetary images.” In: (Oct. 2010).
- [6] P. Guio and N. Achilleos. “The VOISE algorithm: a versatile tool for automatic segmentation of astronomical images.” In: *Monthly Notices of the Royal Astronomical Society* 398.3 (2009), pp. 1254–1262. ISSN: 0162-8828. DOI: 10.1111/j.1365-2966.2009.15218.x.
- [7] Patrick Guio. “Note about Planetary Limb/Terminator calculation.” In: (2016).
- [8] L. Lamy et al. “The Auroral Planetary Imaging and Spectroscopy (APIS) service.” In: *Astronomy and Computing*. The Virtual Observatory: II 11 (June 1, 2015), pp. 138–145. ISSN: 2213-1337. DOI: 10.1016/j.ascom.2015.01.005. URL: <https://www.sciencedirect.com/science/article/pii/S2213133715000062>.
- [9] Laurent Lamy. “Etude des émissions radio aurorales de Saturne, modélisation et aurores UV.” In: (Sept. 9, 2008). URL: <https://tel.archives-ouvertes.fr/tel-00328763>.
- [10] X. Li et al. “Dayglow removal from FUV auroral images.” In: *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*. IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium. Vol. 6. Sept. 2004, 3774–3777 vol.6. DOI: 10.1109/IGARSS.2004.1369944.
- [11] J. Nichols et al. “Oscillation of Saturn’s southern auroral oval.” In: *Journal of Geophysical Research (Space Physics)* 113 (Nov. 1, 2008), p. 11205. DOI: 10.1029/2008JA013444.
- [12] Atsuyuki Okabe. *Spatial tessellations*. John Wiley & Sons, 2000.

- [13] W. D. Pence et al. “Definition of the Flexible Image Transport System (FITS), version 3.0.” In: *Astronomy & Astrophysics* 524 (Dec. 1, 2010). ISSN: 0004-6361, 1432-0746. DOI: 10.1051/0004-6361/201015362.
- [14] R. E. Sequeira and F. J. Preteux. “Discrete Voronoi diagrams and the SKIZ operator: a dynamic algorithm.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.10 (1997), pp. 1165–1170. ISSN: 0162-8828. DOI: 10.1109/34.625128.
- [15] Tilman Spohn, Doris Breuer, and Torrence Johnson. *Encyclopedia of the Solar System*. Elsevier, May 30, 2014. ISBN: 978-0-12-416034-7.
- [16] *STIS Instrument Handbook - HST User Documentation*. URL: <https://hst-docs.stsci.edu/stisihb>.
- [17] M. Tuceryan and A.K. Jain. “Texture Segmentation Using Voronoi Polygons.” In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 12 (1999), pp. 211–216. DOI: 10.1109/34.44407.
- [18] Mark B. Vincent et al. “Mapping Jupiter’s Latitudinal Bands and Great Red Spot Using HST/WFPC2 Far-Ultraviolet Imaging.” In: *Icarus* 143.2 (Feb. 1, 2000), pp. 189–204. ISSN: 0019-1035. DOI: 10.1006/icar.1999.6232. URL: <https://www.sciencedirect.com/science/article/pii/S0019103599962322>.

