

Automatic detection of strong aurora

combined with detection of auroral infrasonic waves at IRF Kiruna.

Name:	Dennis van Dijk	Company:	Swedish Institute of Space Physics
Studentnumber:	11049901	Department:	Solar Terrestrial Physics (STP)
Submission date:	August 26, 2016	Location:	Kiruna, Sweden
Study:	Applied Physics	Supervisor:	M. Yamauchi
First reader:	L.H. Arntzen	Co supervisor:	J. Kero
Second reader:	R.H.M. Smit		

Abstract

A program that automatically detects auroral events in all-sky images taken by the all-sky camera of IRF has been written. Current programs that automatically detect the aurora in all-sky images using the RGB color system often miss strong aurora and sometimes interprets clouds as aurora. To improve the detection of aurora in all-sky images, filtering in the HSL colorspace has been investigated in this thesis. The aurora in the images have been detected in three different strengths. The first filter has soft requirements and is able to detect faint aurora, but also captures noise. The second filter is capturing most aurora except in cloudy days, where the last filter has a good accuracy on finding aurora with a minimum amount of noise in every situation. The program also determines the strength of the aurora by taking the strongest part and classifying the aurora as regular or strong aurora. If an event is identified as strong aurora, the UT of occurring will be saved in an event list after double checking with eye identification.

The strong aurora events are used to examine a possible correlation between infrasound and aurora, what was reported at Fairbanks in Alaska but not confirmed. This thesis investigates if there is a correlation between the strong aurora events captured by the IRF all-sky camera and the data of the infrasound array at the same location. No correlation has yet been found with the currently available infrasound data. This could mean that the strength of the aurora is not a sufficient parameter to find the type of aurora that generates infrasound at the ground. Also have the auroral infrasonic waves detected in Alaska been measured with an infrasound array that was sensitive to lower frequencies than the array used in Kiruna.

Table of contents

1	Introduction	4
2	Fundamental physics of infrasonic waves created by aurora	6
2.1	The Sun and its solar wind	6
2.2	The magnetosphere of the Earth	8
2.3	The excitation of the aurora	9
2.4	Auroral electrojets	11
2.5	Sound waves	14
3	Experimental setup	16
3.1	All-sky camera	16
3.2	Magnetometer	17
3.3	Infrasound microphones	17
3.4	Color space	18
4	Results	19
4.1	The basics of the program	19
4.2	Strong aurora	21
4.3	Magnetometer data	22
4.4	Infrasound	24
4.5	Yearly change in strong aurora events	25
4.6	Propagation direction of an auroral arc	25
5	Conclusion	27
6	Outlook	28
	Appendices	31
A	Induction equation	32
B	Captured substorm	34
C	Switch from RGB to HSL	35
D	Strong aurora eventlist 2015	37

E	Written programs	38
E.1	Automatic detection of strong aurora events	38
E.2	Moving Aurora	38

Chapter 1

Introduction

The aurora is a phenomenon where red, green and blue lights appear in the sky. The lights are visible in the night during a geomagnetic storm. The lights occur most often near the north-pole (aurora borealis) and south-pole (aurora australis). Research in the origin of the aurora was difficult before the 20th century, because the physics to describe the aurora was yet to be discovered. An important theory, to describe and understand the emission of the wavelengths by molecules and atoms, was quantum mechanics, which was discovered around the first decade of the 20th century. Before the science could provide a satisfying explanation for the origin of the aurora, different speculations existed. Scandinavian people believed that it could be a reflection of the sea and in the middle ages the aurora was called "The dance of the spirits" [12].

The researcher E. Halley mentioned in 1716 that the aurora is orientated in the upper atmosphere along the magnetic field lines of the Earth. In 1790 the height of the aurora was triangulated by H. Cavendish and in 1859 the auroral oval was plotted by the observations of E. Loomis. From the data of the geomagnetic disturbances between 1902 and 1903, researcher K. Birkeland mentioned in 1908 that field-aligned electrical currents could be associated with aurora. In 1967 a satellite showed that the suggested currents of K. Birkeland existed and are now also called Birkeland currents [10].

All the discoveries that have been mentioned in the previous paragraph, relied on own observations. To increase the efficiency of the research in aurora, all-sky cameras (a camera that makes a picture with an angle of almost 180 degree) have been placed in different locations. IRF installed an all-sky camera in the International Geophysical Year, which was in 1957 [16]. Later this camera has been replaced by a digital camera, which will be used in the thesis to automatically detect the aurora and make an event list of stronger aurora events of the last decade.

Dr. Charles R. Wilson has shown in Alaska during the 60's that large aurora forms, that propagate with supersonic speeds, radiate pressure waves. The frequency of the recorded pressure waves was between 0.025 and 0.012 Hz. However, it is not clear what type of aurora can create the infrasound pressure waves, because they found coexistence of aurora and pressure pulse only in about 15 cases over 2 years, which is a way to small number compared to total occurrences of aurora. In this sense their results are not confirmed in statistical sense. For the statistical study

we need all auroral events, which requires an automatical detection method, because of the large number of auroral image data in modern days. When IRF started recording infrasonic waves in 1971, Wilson started searching for the auroral pressure waves in Kiruna. But the equipment installed in Kiruna had a detection frequency of around 2 Hz, which was higher than the infrasound array used in Alaska and no correlation was found. The currently installed infrasound array have a lower frequency bandwidth than the old array and should be able to detect sound waves between 10 Hz and 0.1 Hz. The measurements of the new infrasound array will be used in this thesis to determine whether it is or is not possible to detect a correlation between the aurora and the infrsound waves with the new infrasound measurement equipment. The purpose of this thesis is to develop a reliable automatic detection scheme of strong aurora and examine the correlation.

In chapter 2 of the thesis the origin of the aurora and the current state of the research in infrasonic waves is explained. Also the thoughts about why the aurora could emit infrasound is mentioned. Chapter 2 covers the equipment of IRF that has been acquiring data for the thesis. The equipment has been mentioned shortly and a basic explanation of how the equipment works has been given. In chapter 4 the results of the program that has been written are given and explained what the conclusions are of the acquired results. The main results are the precision of the detected aurora and the search for the infrasound waves. As last the change in the amount of strong aurora events a year has been investigated.

Chapter 2

Fundamental physics of infrasonic waves created by aurora

The purpose of this chapter is to inform the reader about the origin of aurora. In the beginning of the chapter solar activities will be mentioned and how they interact with Earth's magnetic field in order to create the aurora. In section 2.4 the related currents will be explained. As last, in section 2.5, the sound waves and infrasonic waves will be mentioned and some properties of the already measured auroral infrasonic waves at Fairbanks will be listed.

2.1 The Sun and its solar wind

Our Sun is a star held together by its own gravity and is fusing hydrogen into helium to create a gas pressure to not collapse by the strength of the gravity. Around 98% of the radiation sent out by the Sun are photons, while the other 2 % is a constant stream of plasma from the upper atmosphere. Plasma is the fourth state of matter and is created by adding energy to a gas so that the atoms get ionized. Plasma is highly conductive and reacts strongly to electric and magnetic fields. The plasma is released in every direction and is called the solar wind.

The solar wind contains mostly charged electrons, protons and alpha particles in the energy range between 1.5 and 10 keV. The solar wind plasma has a weak magnetic field embedded, which is held together by the plasma. If the solar wind did not exist of plasma the magnetic field would be free to escape with the speed of light. When the solar wind reaches interplanetary space, the embedded magnetic field is called an IMF (Interplanetary Magnetic Field). The creation of the solar wind is a difference in gas pressure between the solar corona and interstellar space. The solar corona is an area of plasma that surrounds the Sun and extends into space for millions of kilometers. When the solar wind reaches a distance of 1 AU (Astronomical unit, average distance between the Sun and the Earth), the average strength of the IMF is around 7 nT [10,12,19].

Two different types of solar wind have been observed by satellites, a fast and slow solar wind. The fast solar wind has a speed of around 750 km/s. A typical source of fast solar wind is a coronal hole. Coronal holes are darker and cooler spots on the Sun and have lower plasma

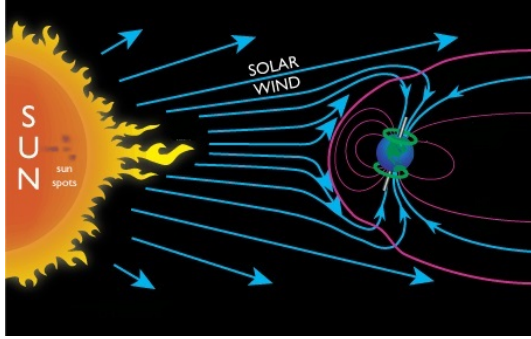


Figure 2.1: *In the figure a drawing is given of the Sun emitting the solar wind in the direction of the Earth and the paths it may follow during the interaction with Earth [3].*



Figure 2.2: *A picture of a coronal mass ejection taken in the wavelengths 17.1 and 30.4 nm by NASA/SDO [18].*

densities, because they allow more particles/plasma to escape through open magnetic field lines. A coronal hole can have a lifetime up to six months. On the other hand slow solar wind is emitted by events with closed magnetic field lines, has a higher plasma density and moves with an average speed of around 400 km/s. A drawing of possible tracks the solar wind may follow during an interaction with the Earth is given in Figure 2.1 [10, 12, 19].

On the Sun the appearance of sunspots is not unusual. How often the sunspots appear changes with a solar cycle of around 11 years. Right after a minimum, solar sunspots appear at $\pm 30^\circ$ from the equator on the Sun to move during the solar cycle to $\pm 4^\circ$ and leaps back to $\pm 30^\circ$ for a new cycle. At the end of two solar cycles the magnetic field of the Sun reverses. This means that during a cycle of about 22 years the Sun returns to its initial state. During a solar minimum the number can be as low as one sunspot in five days, where during a solar maximum three in a one day is not unusual [10, 12].

An unusual increase in solar wind from the Sun is often caused by a CME (Coronal Mass Ejection). At the occurrence of a CME the amount of plasma and magnetic field released into space by the Sun is high. A CME can be ejected by different solar events like a sunspot. A picture of a CME taken by NASA is given in Figure 2.2. Stronger CMEs can interact with Earth's magnetosphere and create aurora, more about this is given in section 2.4.

During a solar maximum CMEs are mostly the cause of aurora and during a solar minimum coronal holes tend to dominate. Because coronal holes last for up to six months and the Sun circulates in 27 days around its axis, during a solar minimum a 27 day recurrence tendency of aurora may be observed [12].

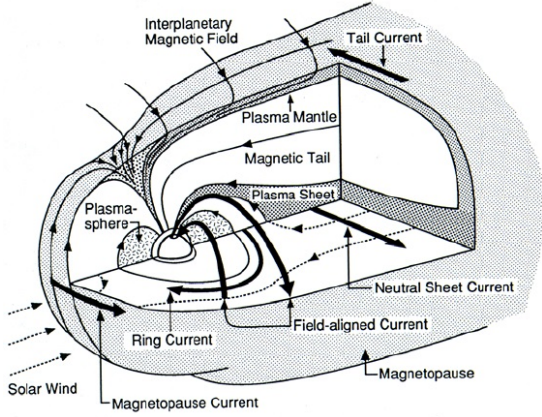


Figure 2.3: *configuration of Earth magnetosphere [9].*

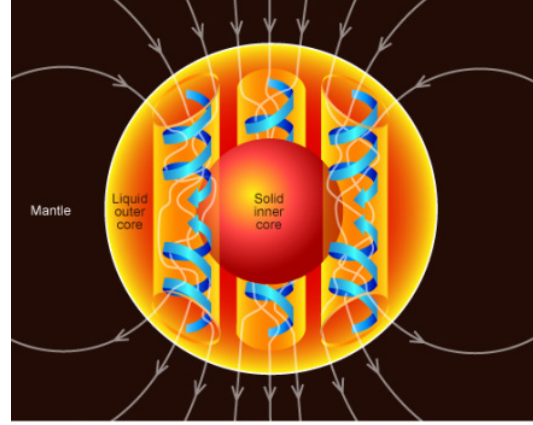


Figure 2.4: *Visualization of Earths dynamo effect [27].*

2.2 The magnetosphere of the Earth

The Earth has a magnetic field that is maintained by the motion of the melted iron around its solid core. This motion is known as a theory called the dynamo theory. The dynamo theory has a self-consistent magnetic field generation. For this effect to happen, kinetic energy is required from planetary motion. With enough kinetic energy the electrical conductive iron fluid core of the Earth creates electrical currents. The induction of the magnetic field can be described with the use of Ampere's-, Ohm's-, and Faraday's law in the following partial differential equation [12]:

$$\frac{\partial \vec{B}}{\partial t} = \frac{1}{\sigma \mu_0} \nabla^2 \vec{B} + \nabla \times (\vec{v} \times \vec{B}), \quad (2.1)$$

the derivation of this formula is given in Appendix A.

Earth's magnetic field protects the Earth from the constant bombardment of the solar wind with its embedded IMF. The planetary magnetic field looks like a typical dipole close to Earth, but further away the magnetic field lines are disturbed by the IMF. For the solar wind, Earth's magnetic field is an obstacle and flows around it. The void region of the solar wind around the Earth is named the magnetosphere. A schematic figure of Earth's magnetosphere is given in Figure 2.3. The important parts of the magnetosphere in Figure 2.3 are mentioned in more detail below [12, 25].

The bow shock: The bow shock is the boundary where the IMF reaches the magnetosphere. At the bow shock the speed of the solar wind decreases from supersonic to subsonic and is located at $\pm 9 \cdot 10^4$ km from the Earth at subsolar distance (shortest distance between the Earth and the Sun). The position of the bow shock changes by a strength variation in the solar wind [12].

Magnetosheath: At the bow shock the solar wind is pushed around the Earth into the magnetosheath. In the magnetosheath the magnetic field of the Earth is not strong enough to

completely deflect the solar wind. When the solar wind plasma flows around the Earth it compresses the magnetosphere on the sunward side and extend it on the night side [25].

Magnetopause: The magnetopause is the boundary where the plasma pressure of the solar wind and the magnetic pressure of the magnetosphere are equal. The location of the magnetopause is dynamic and can move inwards or outwards when the strength of the solar wind respectively increases or decreases. At the magnetopause a current sheet flows which separates the IMF and the magnetosphere. This thin boundary has is a weakly magnetized high density plasma of the solar wind at one side and strongly magnetized low density plasma on the Earth side [12, 25].

Magnetotail: The magnetotail consists of two tail lobes, a northern and southern one. Those lobes extend in the magnetosphere at the night side for hundreds of R_E (Earth radius, 6370 km). The tail lobes are separated by a plasma sheet that consists of plasma originating from the ionosphere and the solar wind. In the plasma sheet a current flows from dawn to dusk, which is referred to as the neutral sheet current. At the plasma sheet the direction of the magnetic field reverses from pointing away of the Earth to towards the Earth [25].

2.3 The excitation of the aurora

The aurora is the most common evidence of interaction of the solar wind with the terrestrial magnetosphere. During an auroral event, lights appear in the sky together with geomagnetic disturbances. The lights of the aurora are emitted through atomic emission in the upper atmosphere, while the geomagnetic disturbances are created by large electrical currents. These current exist of highly energetic electrons and protons that excite the atoms in the upper atmosphere. The current system in the magnetosphere is further explained in section 2.4 [4].

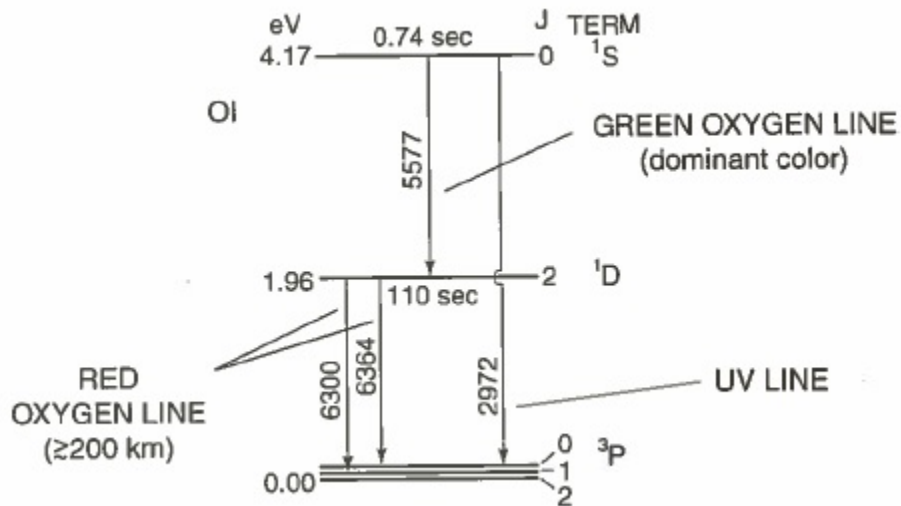


Figure 2.5: The two emission lines of oxygen that are characteristic for aurora [25].

The atoms N_2 , N_2^+ and O are responsible for the emission lines that are visible to the human eye. Also several emission lines for wavelengths in the infrared and ultraviolet emit photons. The atoms that are in an excited state through collisions emit light when they undergo electron relaxation. This happens when an electron is in a higher state and falls back to a lower state where less energy is needed. The emitted photon has a wavelength that corresponds with the same energy as the energy difference between the two states. The wavelengths that appear often during aurora are the 557.7 nm (green) and 630.0 / 636.4 nm (red) lines. Both colors are emitted by the oxygen atom. In Figure 2.5 the electron transitions of oxygen that are responsible for the emission of the green and red line are given. The red emission line occurs only at strong events where enough energetic particles enter the atmosphere, because the chance of a collision is lower in the higher altitude range more particles are needed to excite an sufficient amount of atoms. When the particle energies are higher, the particles can penetrate deeper into the atmosphere and may reach an altitude of around 90km. Around this altitude nitrogen molecules can produce red/pink aurora with a wavelength of around 670 nm [12, 15, 25].

An electron transition for the green emission line in oxygen is hard to determine because of the long lifetime of the electron state. The meta stable state of oxygen was found around 1925 by McLennan and Shrum. This while in 1866 A.J. Ångström already found the wavelength of the green line. The meta stable state is called a forbidden transition, because at sea pressure the electron in the higher state will lose its extra energy through collisions instead of sending out a photon. Because it has a long lifetime of 0.8 seconds, the transition can only occur above an altitude of 100 km where the air pressure is low enough and the atoms collide less frequently. For the red oxygen lines the lifetime is on average 120 seconds, which is even longer than the green emission transition. The red aurora is visible around an altitude of 150 km and higher [12].

The process of how oxygen sends out a photon starts with the collision of an energetic electron or proton with the atom. In the process described below an electron is used for the excitation energy [12, 15, 25]:



The high energetic electron (e) has excited the oxygen atom and continues with less energy (e'). After about 0.8 seconds the excited oxygen electron makes a transition to a lower state and sends out a photon of 557.7 nm:



If the atom does not collide in the next 2 minutes, the electron will lose its last energy in the transition to the stable state. During this transition a photon will be emitted of either 630.0 nm or 636.4 nm light.



The wavelength difference in equation 2.4 is because molecules can store energy in the form of rotation- and vibration energy and can change the wavelength of the photon by a few nanometer. This is what happens in the transition from the 1D to the 3P state of the oxygen atom and causes a wavelength difference of 6.4 nm [12].

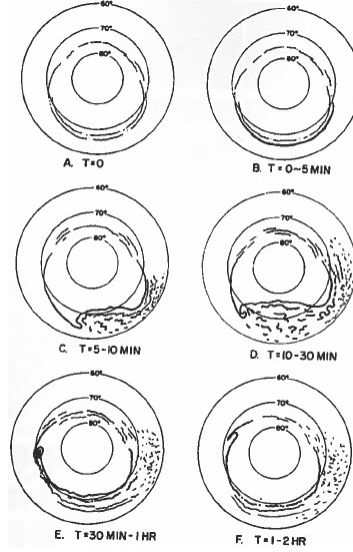


Figure 2.6: A schematic view is given of an auroral substorm divided in several timescales seen from the magnetic pole [2].

During observations of the aurora, S. Akasofu reported an often recurring pattern in the creation of the aurora. In 1964 he wrote a paper about the pattern, which he called the auroral substorm. Before this paper the scientists thought that the aurora would be visible in a fixed pattern throughout the night, but S. Akasofu showed that during an active night multiple substorms can occur, a complete substorm is given in Figure 2.6. In the first phase a substorm usual starts with a single arc in the first 5 to 10 minutes. This growth phase is visible at position A and B. After the growth phase the aurora starts to expand in the next 20 minutes and becomes more diffuse in drawings C and D. The recovery phase is shown in drawing E and F and can take up to three hours after the start of the substorm. During the recovery phase a new substorm can start. A substorm has been captured by the all-sky camera of IRF at 15 February 2016 and can be found in Appendix B [2, 12].

2.4 Auroral electrojets

During an auroral event magnetic disturbances are observed on Earth. In 1903 K. Birkeland proposed that those disturbances were created by large electrical currents, the field aligned currents. The evidence was acquired by the navigation satellite Triad in 1973, which found abrupt changes in the direction of the magnetic field. The abrupt changes proves the proposed electrical currents, because a current induces a magnetic field. This induced magnetic field can be described by Ampere's law:

$$\frac{1}{\mu_0} \nabla \times \vec{B} = \vec{J}, \quad (2.5)$$

where:

$$\mu_0 \quad \text{vacuum permeability } (4\pi \cdot 10^{-7}) \quad \text{N/A}^2$$

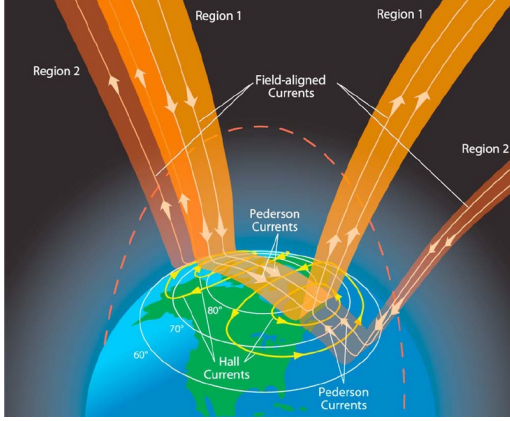


Figure 2.7: The large field aligned current of region 1 and region 2 are given. Also the Hall and Pederson currents are drawn. [13]

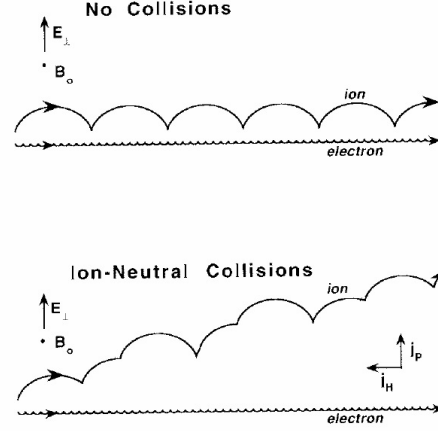


Figure 2.8: In the top figure the ions do not collide, where in the bottom figure the ions do collide and generate a Pederson and Hall current [25].

∇	nabla operator ($\frac{\partial}{\partial x}\hat{x} + \frac{\partial}{\partial y}\hat{y} + \frac{\partial}{\partial z}\hat{z}$)	
B	magnetic field	T
J	current density	A/m ² .

In the magnetosphere the field aligned currents are created by the bulk motion of the moving plasma and the IMF. The generated current can be explained by Faraday's law:

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}, \quad (2.6)$$

where:

E	electric field	N/C
t	time	s.

Faraday's law states that a spatially varying electric field is created by the change of a magnetic field [2, 6].

The field aligned currents flow from two different regions of the magnetosphere into the upper part of the atmosphere. How the field aligned currents flow in the upper atmosphere is given in Figure 2.7. The region 1 current is a poleward current system, this means that it flows close to the poles of the Earth. The poleward current closes its circuit in the current of the magnetopause. The current flowing in region 2 is an equatorward current and is connected to the ring current, which is located at 4 to 5 R_E . The location of the ring current is given in Figure 2.3 [25].

The field aligned currents can flow collisionless in the conductive plasma of the magnetosphere. But when the current reaches the ionosphere, collisions between ions and neutral atmospheric

atoms occur. The ionosphere is the boundary layer between the atmosphere and the magnetosphere. It has a sharp lower boundary at ± 90 km and extends to an upper boundary altitude of ± 300 km. The density of the neutral atoms in the ionosphere decreases by an increase of altitude. So, when the current approaches Earth the amount of collisions increase. When an ion collides with a neutral atom there is a breakdown of the $\mathbf{E} \times \mathbf{B}$ drift of the ion. Because of the collisions between ions and neutral atmospheric atoms, the ions have a lower time-average velocity and a different direction than the collision-free electrons. This results in two additional currents. The first one is the Peterson current, this current flows parallel to the electric field. The second current is the Hall current, which flows in the $\mathbf{E} \times \mathbf{B}$ direction. The flowing paths of the currents are shown in Figure 2.7. For visualization of the collisions for the Peterson- and Hall current see Figure 2.8. All the currents on there own create strong enough magnetic fields that could be measured at the ground. But the Peterson current and field aligned currents cancel each other at the ground and the magnetic disturbance of the Hall current remains. [25].

The field aligned currents carry around $1 \cdot 10^5$ ampere during quiet geomagnetic times, where during a geomagnetic storm the current can increase to more than a million ampere. The more energy the particles in the current have, the deeper they can penetrate into the atmosphere. When the average energy of the incoming particles is high enough, they will excite enough atoms in the upper atmosphere to create the lights of the aurora that can be seen from Earth. High energetic protons and electrons excite about one atom for every 36 eV of their initial energy [12].

At the end of the 20th century the incoming locations of the field aligned currents have been measured and the location have been mapped. The result was an oval shaped form, which is also called the auroral oval. The location of this oval was longer known by the mapping of the aurora occurrences and so in this oval the chances of seeing the aurora is highest throughout the year. When the strength of the field aligned currents increases the oval shifts towards the equator. This is why in lower latitude during a strong geomagnetic event the aurora can be visible. The distribution of the incoming and outgoing field aligned currents is given in Figure 2.9 [2].

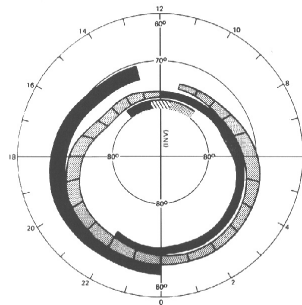


Figure 2.9: *The distribution of the field aligned currents. The black areas are incoming currents and the white areas are outgoing currents [2].*

2.5 Sound waves

Sound waves are pressure waves that need a certain medium or material to propagate through. The waves can propagate as a longitudinal wave (P-wave) or transverse wave (S-wave). P-waves are compression waves that are compressed at a certain position and stretched on the next, while S-waves are waves that moves up and down. In mediums such as air, liquids and solids the waves propagate as a P-wave, but in solids S-waves are possible as well. The two different waves are visualized in Figure 2.10. Sound waves will propagate away from it's source with the local speed of sound. This speed varies in different mediums. In air the speed of sound is around 340 m/s, while in water it is almost five times as high. Sound can move with different frequencies depending on the source [29].

Sound has a frequency spectrum with frequencies above 1 GHz are named hyper sonic and higher than $2 \cdot 10^4$ Hz is ultrasonic. The area between $2 \cdot 10^4$ Hz and 20 Hz is sonic and below 20 Hz is infrasonic. Infrasonic sound waves can travel far without being damped or absorbed by the atmosphere. The wavelength of infrasonic waves are usual larger than 17 m [29].

Different sources can create infrasound. Natural events that create infrasound are earthquakes, volcanoes and meteors. Also several human made structures like wind turbines and diesel engines produce the low frequency pressure waves. Also the aurora can create an infrasound waves [29]. The first measured AIW (auroral infrasonic wave) was measured in Alaska at Fairbanks. Several AIW signals have been identified to specific auroral arcs. The infrasound produced by the aurora is in the form of acoustic bow-waves. Acoustic bow-waves are waves created by a source that moves faster than the speed of sound. An AIW has been observed in Alaska at Fairbanks on September 11, 2005 and is given in Figure 2.11 The pressure traces of the Alaska sensors have been phase-aligned and gave a coherence of 0.98. The azimuth arrival angle was 46° and the trace velocity 0.56 km/s. The measured time of the AIW was in daylight, so no aurora pictures

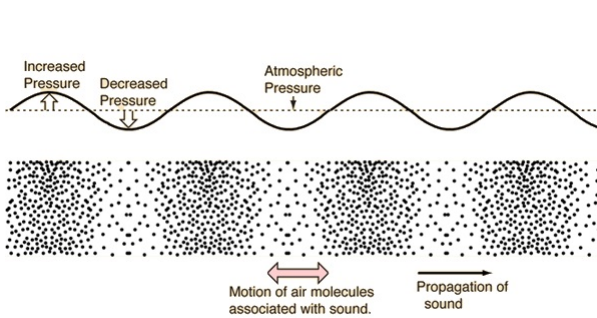


Figure 2.10: The two kind of different ways a sound wave can travel through a medium. At the top an S-wave is visualized, where at the bottom a p-wave is drawn [7]

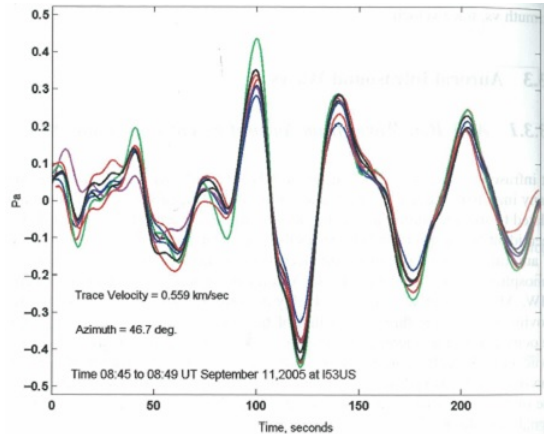


Figure 2.11: The AIW found at the start of strong geomagnetic activity at Fairbanks in Alaska. [1]

are available for the event. But the geomagnetic data of a measurement station 30 km to the north has observed strong geomagnetic fluctuations, which indicates the presence of a strong auroral electrojet current. The discussion about how aurora may be able to create AIW is not finished yet. One of the theories behind the AIW mentioned that the auroral electrojet currents may heat up the atoms and ions in the upper atmosphere and because of the increased heat they will expand and a pressure wave may be created by the expansion. [1, 29, 30].

Chapter 3

Experimental setup

3.1 All-sky camera

At IRF Kiruna an all-sky camera has been placed on the roof since the International Geophysical Year 1957. The exact location of the all-sky camera is 67° latitude and 20° longitude. This camera has been replaced by a digital camera in 2001 and has been upgrade to a Nikon D700 in 2009. The camera has a high sensitivity CMOS sensor with an maximum ISO of 6400, which is needed to detect faint auroras. Also has the camera a adjustable shutter time with the a minimum of 30 seconds. On the camera a special lens is placed that can create pictures with an angle of almost 180° . The camera makes a picture of the sky every minute of the night with a shutter time of 6 seconds [16, 20]. The pictures are saved on the IRF server, which can be accessed through the website of IRF. In Figure 3.1 an all-sky picture, captured at 18 march 2016 by IRF in Kiruna, is shown. This picture has the regular geomagnetic coordinates. So, north is up, east is right, etc.

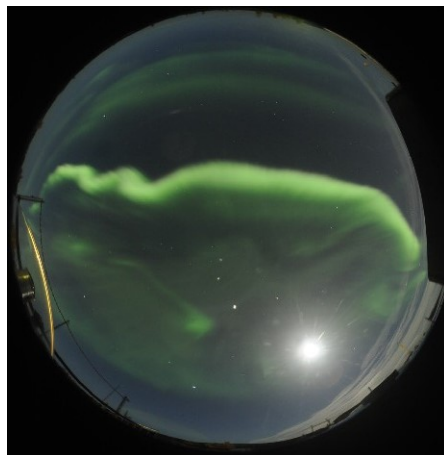


Figure 3.1: *An all-sky image taken by the Nikon D700 in Kiruna at IRF on 18 march 2016.*

3.2 Magnetometer

To measure the strength and variations in Earth's magnetic field, different magnetometers are installed at IRF. The installed magnetometers can be divided in groups for absolute measurements and short term variation measurements. For an absolute measurement of the magnetic field a PPM (proton-precession magnetometer) is used. This magnetometer has a direct current flowing through solenoids to create a strong magnetic field. This field aligns protons of a sample along this artificial field. When the current is disrupted, the protons align themselves again with Earth's magnetic field. During this alignment they precess at a frequency that is proportional to Earth's magnetic field. When the protons precess they generate a small circular magnetic field, which will be picked up by a coil. Because the strength of the magnetic field changes in the coil by the circular magnetic field, a current is generated that can be explained by Faraday's law (equation 2.6). The strength of this current is proportional to the strength of Earth's magnetic field. The results of these measurements are used every month to calibrate the offset of the short term measurement magnetometers [8, 22].

For the short term measurements a flux gate magnetometer is used and measures the small changes in magnetic field every second. An easy flux gate magnetometer makes use of two equal coils. A magnetic field is created in the first coil by an alternating current and picked up by the second coil. As can be explained by Faraday's law, the changing magnetic field in the second coil creates a current through the wire. This current can be measured and should be the exact opposite as the current through the first coil if there is no magnetic field. When there is an ambient changing magnetic field the current through the second coil will not be exactly the same but will have an offset in the direction of the change in the external field. This offset is proportional to the change in magnetic field [6, 17, 22].

Earth's magnetic field is measured at IRF in three different components, where every component has a prearranged direction. B_x , B_y and B_z are the three components and all point in a fixed direction: B_x points towards the real North, B_y to the east and B_z into Nadir, which is straight down [22].

3.3 Infrasound microphones

In Kiruna three microphones are placed in the form of a right-angled triangle. The length of the two shortest distances is 75 meter. The advantage of a three microphone system is that the angle of arrival can be determined and also the trace velocity. To determine these parameters a correlation between the three microphones has to be found. To determine the amount of correlation the cross-correlation summation is used:

$$(f \star g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[m+n]dt. \quad (3.1)$$

In the equation f and g are both a discrete signal, such as the samples of the infrasound microphones [24]. For the sound arrays the data is not used from $-\infty$ to ∞ . In the determination

of the correlation the maximum amount of time that sound needs to get at every microphone is taken into account and gives a finite number of data points for the correlation. The trace velocities of the incoming waves will be calculated, where a speed of 340 m/s means a wave that comes in horizontally and the higher the trace velocity the bigger the angle of arrival was. Because three microphones are available, the incoming direction of the pressure wave can be triangulated. The processing program is calculating these variables automatically [11].

3.4 Color space

A famous color space is RGB (Red, Green and Blue) and is used in electronic devices such as televisions, cellphones and computer monitors. RGB can be represent as a 3d plot in Cartesian coordinates as in Figure 3.2. In Cartesian coordinates, every color has its own dimension. Red along x, green along z and blue along the y-axis. By combining the three components different colors can be created [28].

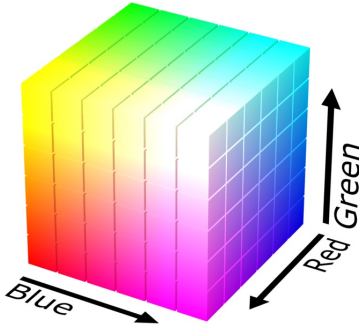


Figure 3.2: *Red, green and blue color space in Cartesian coordinates [5].*

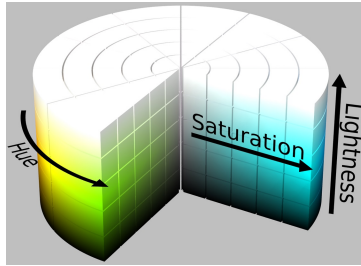


Figure 3.3: *Hue, saturation and lightness color space in cylindrical coordinates [26].*

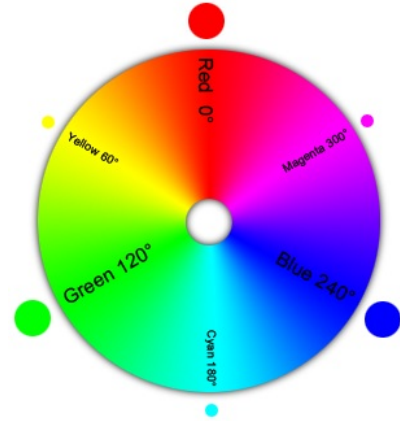


Figure 3.4: *A cross section of the hue in the HSL color space [14].*

Because it is difficult to find a certain color range through RGB, HSL (Hue, Saturation and Lightness) is used in color pickers and image editing software. This color space uses cylindrical coordinates instead of cartesian. In the software two of the HSL dimensions are shown, while the third dimension is placed on an external slider. An image of the cylindrical color space HSL is given in Figure 3.3 and a cross section of the hue is given in Figure 3.4. In HSL it is easier to pick a certain color and by changing the saturation or lightness, different variations can be made of that color. Because HSL is used in editing programs but monitors make use of RGB, there is a way to switch between HSL and RGB. How to toggle from RGB color space to HSL color space is given in appendix C [28].

Chapter 4

Results

In this chapter the basics of the program, that is written in python 3.5, will be explained, followed up by some more detailed results of the program. It will begin with explaining how the stronger aurora events are determined. Then it continues with the method how the aurora could be determined by the use of the magnetometer data. After that the infrasound data is used to search for auroral infrasonic waves. The program that has been used for the sections just mentioned has been given in appendix E.1. The chapter will finish with some long term aurora event data and a beginning has been made for determining the propagation direction of an aurora arc.

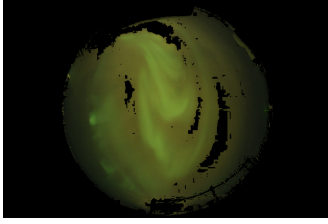
4.1 The basics of the program

In order to find aurora events in the sky without the need of scientists being outside, all-sky images are used to photograph aurora. To automatic detect aurora in the pictures a program is written in python 3.5, which searches for different preset colors. It is difficult to acquire a good accuracy on finding all the auroral events because of different weather conditions like clouds, rain, moon light, blue sky and dark night sky. In 2015 B. Ölsthorn started writing a program based on the RGB color space filtering. The RGB color space has been changed by M. Yamauchi early 2016 into the HLS color space. He mentioned that RGB was not able to separate strong aurora from regular aurora and that RGB was unable to detect aurora through clouds.

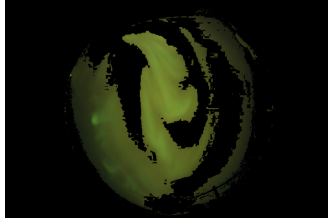
To completely automatic filter the aurora, the program is able to download the images from the server by itself and then process the picture with the preset color filters. Because the aurora

Table 4.1: *The different settings for the filters to distinguish aurora in the pictures.*

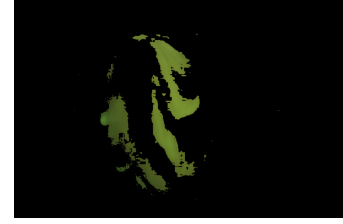
Filter	H	S	L
Maybe aurora	$0.16 < H < 0.50$	$0.10 < S < 0.80$	$0.05 < L < 0.80$
Probably aurora	$0.18 < H < 0.46$	$0.15 < S < 0.80$	$0.10 < L < 0.80$
Aurora	$0.20 < H < 0.46$	$0.25 < S < 0.80$	$0.20 < L < 0.80$
Clouds	$0.00 < H < 0.16$	$0.10 < S < 0.80$	$0.15 < L < 0.80$



(a) *Maybe aurora*



(b) *Probably aurora*



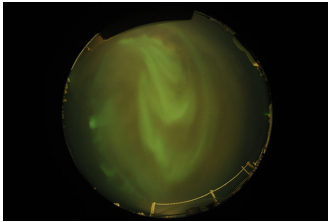
(c) *Aurora*

Figure 4.1: *The different filters used by the program. This example has aurora through clouds, which is the hardest to detect correctly. In this example it can be seen that only the detection with highest requirements is sufficient enough to detect the aurora.*

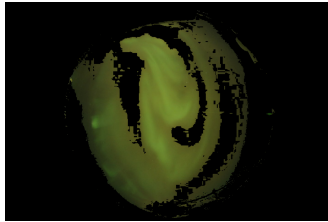
has various colors at different weather events, three different filters are used and given in table 4.1. How the different filters affect a picture of cloudy aurora, the detection capabilities of the filters is given in Figure 4.1. Also a fourth filter is made to detect the clouds, this can be used for later researches where different detection methods are used than all-sky images to detect aurora. Then it may be important to know whether there was no aurora or the clouds may have blocked the aurora.

In HSL the colors are placed only in the hue, which makes it easier to filter the green color. Also it is easy to use settings in lightness to remove the dark spots (night sky) and the overexposed white pixels (the moon and stars). To show the difference of what HSL can offer compared to RGB, a comparison of pictures is shown in Figure 4.2. Because the program switches from RGB to HSL and back, in principle both methods could give comparable results but this will cost more requirements in RGB and small adjustments in a filter are harder to make in RGB compared to HSL. As seen in table 4.1 the requirements of the filters shift slightly and in an easy way and have a great difference in the colors that are filtered, which makes it easier to adjust the filters for the users needs.

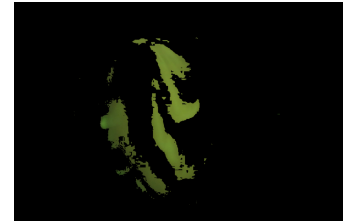
Because there is more than 10 years of pictures available, saving all the pictures and watching



(a) *Original all-sky picture*



(b) *RGB filter*



(c) *HSL filter*

Figure 4.2: *Here the difference is visible that is achieved by changing from RGB to HSL. The filter settings determined by B. Ölstorn for the RGB picture are $(G > R)$, $(G > B)$, $(G > 40)$, $(B \leq 150)$ and $(R \leq 138)$ [23]. For HSL the filter settings are $(0.20 < H < 0.46)$, $(0.25 < S < 0.80)$ and $(0.20 < L < 0.80)$.*

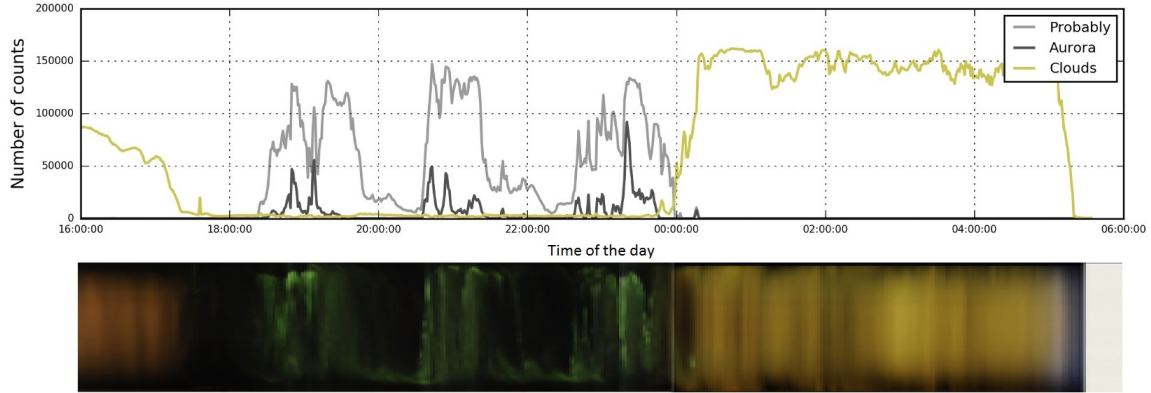


Figure 4.3: *The top plot is obtained by the filtering the photographs of the all-sky camera. The bottom part is a keogram, which is a pixel line from north to south. The result of the filtered photographs contains more information, without losing in accuracy.*

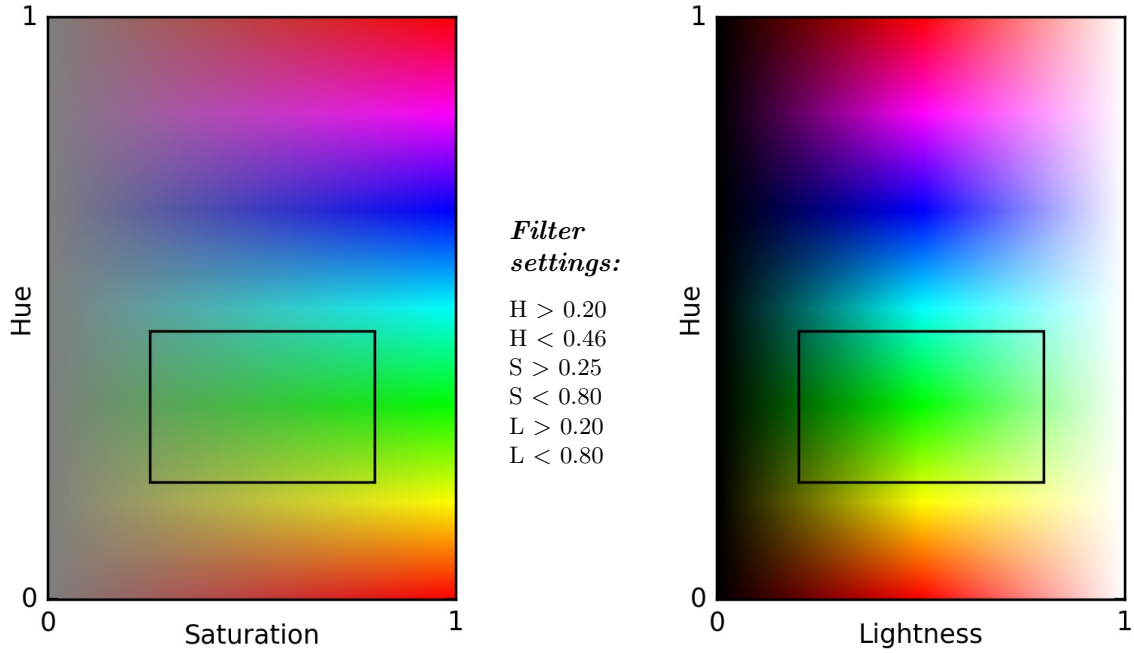
them one by one would be a long process. To give a fast overview of a complete night of data, the number of pixels that satisfies the requirements for a filter are summed for every photograph. The number of corresponding pixels can then be plotted against the time of the day. Currently on the IRF website a magnetokeogram is available, where in a magnetokeogram a pixel-line from north to south of the all-sky pictures is plotted together with the magnetometer data. To show the graph of a single day and the validity of the filters, a keogram of 8 November 2015 is taken together with the processed data of the same day by the program and plotted in Figure 4.3.

4.2 Strong aurora

To increase the chance of finding a correlation between infrasound and aurora, stronger aurora events have been examined first. Because if the aurora releases infrasound, the chance of finding the infrasonic waves is higher at stronger events. As mentioned in section 4.1 multiple filters are used in the program to distinguish the aurora. The combination of colors that is assigned in the filter labeled as aurora is given in Figure 4.4. The two given rectangles should be placed perpendicular of each other to make a 3D block full of different colors.

To determine the actual strength of an aurora event, not all the pixels that have been filtered are used in the calculation. If all of the pixels would be used, strong new aurora events within an still ongoing recovery phase of an earlier sub-storm would drastically drop the strength of the event. This is because the recovery phase has a faint diffuse aurora. To get around this obstacle the 2500 pixels with highest lightness are used in the calculation. The lightness of the 2500 pixels is summed up and then divided by 2500 to get a relative aurora strength. In the program the scale of the data has been modified so it will give a actual strength between 0% and 100%. The formula for the calculation looks like:

$$\frac{\sum_{p=0}^{2500} L_p}{2500} \cdot 100\% = L_{relative}, \quad (4.1)$$



(a) Hue and saturation color space with in the rectangle the colors used by the filter.

(b) Hue and lightness color space with in the rectangle the colors used by the filter.

Figure 4.4: The filter that is used to obtain the aurora from the all-sky images.

where:

L_p	lightness of a single pixel
$L_{relative}$	average lightness of the strongest 2500 pixels.

A strong event is when the actual strength of the aurora is above 50%. This value has been found by trial and error and has a good accuracy whether the observed aurora is strong or not. The program detects automatically if the auroral event is above 50% and will plot a sign at the strongest position. Further, it downloads the picture that corresponds with the time of the event and plots it under the graphs. A list of strong aurora events in 2015 has been posted in appendix D

4.3 Magnetometer data

As explained in section 2.4 electrojet currents are responsible for the excitation of the atoms in the atmosphere. Also in section 2.3 in Figure 2.6 it is shown that a single aurora sub storm has a regular pattern. To search for a correlation between the aurora and infrasonic waves, it may be better to search for strong aurora arcs instead of just strong aurora. It is easier to find in which direction an aurora arc is going than diffuse aurora, which may give an estimation of from what direction the auroral infrasonic wave may come. The moment the field aligned currents increase

in strength there is locally an abrupt change in Earth's magnetic field and the strength of the magnetic field returns back to the regular strength over time. To find abrupt changes in a signal the standard deviation (STD) can be used over a finite interval of the data. The used formula to determine the rapid changes is given in equation 4.2. To calculate the STD of every minute in the data an interval of 11 minutes is used. So, to calculate the STD of 11:06, the data from 11:01 until 11:11 is used within the formula.

$$s = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}, \quad (4.2)$$

where:

s	standard deviation
N	number of values used in the calculation
x_i	values used in the calculation
\bar{x}	mean of the used values.

In the calculation of the STD the B_x component is used, because as mentioned in section 2.4 the Hall current is responsible for the measurable changes in magnetic field at the ground. By using the right hand rule and the coordinate system given in section 3.2 it is found that the field changes in the B_x direction.

If the STD is high the probability of finding an aurora arc on the all-sky pictures increases. When there is a geomagnetic storm and the aurora is behind clouds, the aurora is not visible on the all-sky pictures. This is the main reason the magnetometer data can not be used on its own



Figure 4.5: The top part is the keogram and on the bottom part the x -direction of Earth's magnetic field is plotted. The abrupt change of the magnetic field in the picture has been measured by IRF on January 22, 2016 around 20:30 UTC.

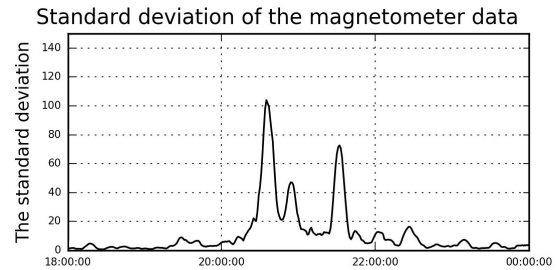


Figure 4.6: Here the standard deviation of the abrupt change is plotted. The first peak corresponds with the deep change towards 10300 nT. The last peak is the small change from the normal magnetic field strength to 10580 nT.

as an detection method for aurora. To show how the magnetometer data changes on a strong aurora event, a part of an magnetokeogram has been taken from the IRF website and placed in Figure 4.5.

4.4 Infrasound

The infrasound data of the microphones at IRF in Kiruna is processed. The processed results of an infrasonic measurement are the angle of arrival, trace velocity and the cross-correlation of the measured pressure wave. Because a correlation between aurora and infrasound is the wanted result, some data can be removed. As mentioned in section 2.5 the trace velocities of auroral infrasound waves lay between 0.5 km/s and 1.6 km/s. Also signals with a low cross correlation will be removed, because often these signals represent noise. While working with the data and removing the data that was not needed for the research of aurora infrasound waves, it seemed that the trace velocity estimation shown in the online plot differs from the one stored in files for offline use. This has turned out to be caused by the unintentional usage of two different algorithms and will be fixed in the future. In principle both algorithms should give comparable results. But because the difference in velocity seems random, not enough data points can be removed to see whether there is a correlation or not on large time intervals. The data processed on the IRF server is correct and used to search for correlations at strong aurora events.

The strong aurora events of 2015, which are given in appendix D have been investigated. The infrasound data measured during the event on January 21, 2015 is given in Figure 4.7. During this strong event, nothing noticeable happens at the different azimuth angles. The other events had no mentionable outcomes as well. Throughout the year a large amount of data is dominated by wind turbines around Kiruna. To further investigate the infrasound data, the event list is given to professor emeritus Ludwik Liszka. This professor has written his own analyze software for the Kiruna infrasound array to detect lower frequency waves. These results will be available after this thesis is finished.

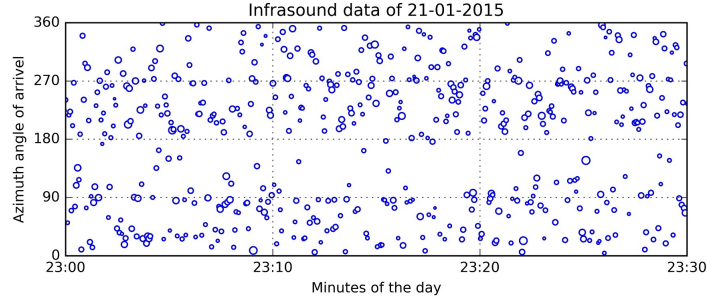


Figure 4.7: *The infrasound arrival angles during the aurora event on 21-01-2015. The bigger the circles in the plot, the higher the cross correlation of the measured signal.*

4.5 Yearly change in strong aurora events

In section 2.1 is mentioned that the Sun has a solar cycle of about 11 years. Because the all-sky images are available since 2002, statistical data of the amount of strong aurora events a year can be acquired. The data is processed by the program and compared to the number of sunspots a year. It is known that the strongest aurora events occur at the decaying phase of the solar cycle. In Figure 4.8 the amount of strong aurora events is compared with the amount of average sunspots a month throughout the year. The data on the amount of sunspots is acquired from the website of the Royal Observatory of Belgium [21].

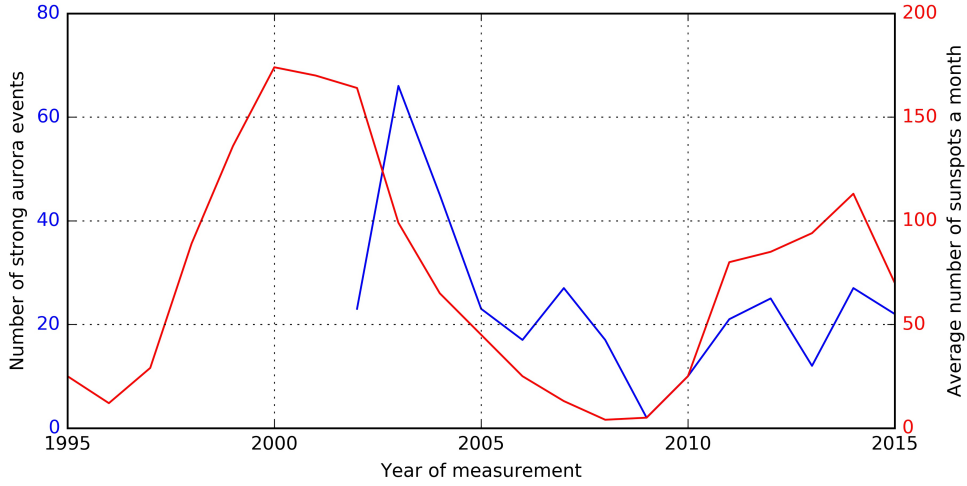


Figure 4.8: *The red line is the average amount of sunspots a month and the blue line is the amount of detected strong events by the program.*

In the figure the number of strong events is high at the decaying area of the amount of sunspots, which was expected. It has to be noted that the data before 2009 has a lower reliability, because the all-sky camera has been changed in 2009 and the filters have been written for the new camera. All the events have been crosschecked by eye and wrong events have been removed from the data. To get a better estimation of the correlation a lower requirement for strong aurora can be used to increase the amount of aurora events.

4.6 Propagation direction of an auroral arc

The direction of an auroral arc is import for further research into AIW. When the direction of an auroral arc is known, searching for infrasound events from a certain angle is more convenient. The by the program identified aurora events that contain an auroral arc have been used. The propagation of an auroral arc events is easier to determine, so it is a good place to start.

Multiple pictures are needed to find the direction of propagation in an auroral arc. The program takes the picture of a strong event and also the picture one minute before and after the event. In every picture of the three, it creates a weighted average of the location of all the aurora pixels in

a single row. During an auroral arc event this should create a line around the middle of the arc. By determining the average position of every row, all arcs that lay horizontally in the pictures are automatically ignored. These arcs do not appear often, because the arcs will usually form along the auroral oval, which is defined by the plot of the field-aligned currents in Figure 2.9. It should be noted that the orientation of the all-sky images on the website are turned by 90 degree. so, the arc flows from east to west, which is from top to bottom in the pictures.

One of the observed events, is the event on January 21, 2015. The three all-sky pictures that have been used to determine the direction of propagation are given in Figure 4.9 and the result of the program is given in Figure 4.10. The program for the detection of the propagation direction has been added in Appendix E.2.

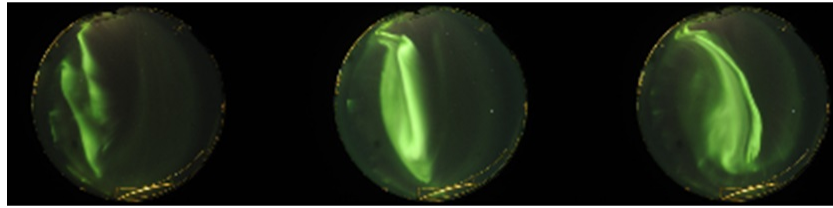


Figure 4.9: *The three pictures taken around the strong event at March 18, 2015. First picture was taken at 23:07, second 23:08 and last picture at 23:09.*

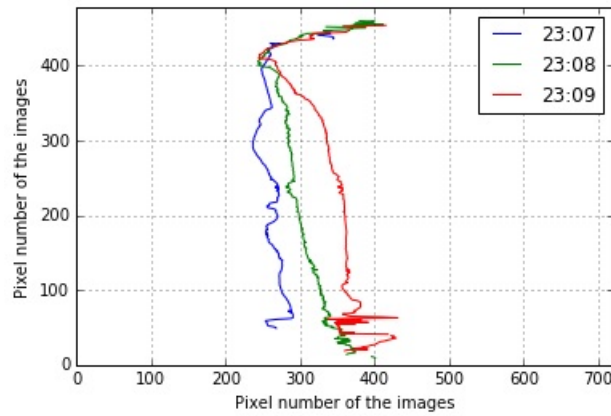


Figure 4.10: *In the picture the result of the program is given. By checking the event in Figure 4.9 it corresponds well with the expected direction. The propagation direction of this arc is towards the North because of the turned geometry of the pictures.*

Chapter 5

Conclusion

When an auroral event is detected by the written program that automatically detects aurora in all-sky images, the aurora will be placed in one of the three categories: maybe aurora, probably aurora and aurora. If aurora has been identified in the picture the strength of the aurora will be determined. When the strength of an auroral event is above 50% the UT of the event will be saved. The precision of the automatic detection program has been compared with the keogram on the IRF website, which cuts out a line from north to south for the available all-sky images. The program showed a good accuracy on the detection of aurora and clouds.

Strong events have been used to search for infrasonic waves created by the aurora. But, during the use of offline data it was discovered that the trace velocities could be measured in different ways, where the offline data had trace velocities with around a magnitude higher than the trace velocities on the website. Also the trace velocities measured during the events in Alaska have been calculated the same way as the online data of IRF. So the online trace velocities of IRF would be suitable for removing infrasound measurement data that have other origins, unfortunately the online data cannot be adapted. The search for infrasonic waves created by aurora has been continued in the online data, but so far no correlation has been found.

After detecting the strong aurora events of the last 13 years, the amount of strong events has been compared to the amount of solar activity in the same year. The expected results have been verified by the plot, which showed that the amount of strong aurora events is highest during the decaying phase of the amount of sunspots. Also the direction of propagation in some of the strong aurora event been determined, this detection is still far from perfect, but a basic start for the program has been made.

Chapter 6

Outlook

That so far no infrasound correlation has been found, does not mean that aurora is not creating infrasonic waves. Dr. emeritus Ludwik Liszka has been working with the infrasound array of IRF and wrote a program that focuses on the detection of the lower frequencies. His analysis for a correlation between infrasound and aurora will be finished after handing in the thesis, so no results are available. If in the analysis no correlation will be found, the parameter for the strength of the aurora may not be sufficient enough to detect the infrasonic waves at the ground and other requirements should be made.

For the aurora propagation direction program, aurora arcs should be detected automatically. If an events has to be found by hand like has be done in the current thesis, the pictures before and after the event can be examined as well and the possible direction may be able to be noticed by eye. An arc may be found by searching for the beginning of a rapid change in Earths magnetic field, which will result in a high standard deviation.

Bibliography

- [1] Le Pichon A. and Blanc E. *Infrasound Monitoring of Atmospheric Studies*. Springer, 2010.
- [2] Syun-Ichi Akasofu. *Exploring the Secrets of the Aurora, Second Edition*. Springer Science&Business Media, LLC, 2007.
- [3] antarcticarctic wordpress. Aurora australis.
<https://antarcticarctic.wordpress.com/2013/07/04/aurora-australis/>, 2014. accessed in August, 2016.
- [4] Diana Rojas Castillo. Conversation at August 24, 2016.
- [5] Craft hubs. The cube of color. <http://www.craft hubs.com/the-cube-of-color/35808>. Accessed in August, 2016.
- [6] David J. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, Inc, 1999.
- [7] hyperphysics. Sound waves in air.
<http://hyperphysics.phy-astr.gsu.edu/hbase/sound/tralon.html>, aug 2016. accessed in August, 2016.
- [8] IRF. Proton precession magnetometer.
<http://www.irf.se/Observatory/?file=protonmagn.html&chosen=program>, 2016. accessed in June, 2016.
- [9] Lagzi Istvan, Meszaros Robert, Gelybo Gyorgyi, and Leelossy Adam. Atmospheric chemistry.
<http://elte.prompt.hu/sites/default/files/tananyagok/AtmosphericChemistry/ch01.html>, 2016. accessed in July, 2016.
- [10] Yohsuke Kamide and Abraham Chian, editors. *Handbook of the Solar-Terrestrial Environment*. Springer Berlin Heidelberg New York, 2007.
- [11] Johan Kero. Multiple conversations in mid July.
- [12] Margaret Kivelson and Christopher Russell, editors. *Introduction to Space Physics*. Press Syndicate of the University of Cambridge, 1995.

- [13] G. Le, J. A. Slavin, and R. J. Strangeway. Space technology 5 observations of the imbalance of regions 1 and 2 field-aligned currents and its implication to the cross-polar cap pedersen currents. *Journal of Geophysical Research*, 2010.
- [14] Shea McCombs. Color systems.
<http://www.upvector.com/?section=Tutorials&subsection=Graphics%20Crash%20Course>, 1999. accessed in August, 2016.
- [15] Billy M. McCormac, editor. *Aurora and airglow*. Reinhold publishing corporation, 1967.
- [16] Arne Moström and Urban Brändström. Welcome to the kiruna all-sky cameras.
<http://www.irf.se/allsky/>, jul 2016.
- [17] Gnter Dr. Musmann, editor. *Fluxgate Magnetometers for Space Research*. BoD, 2010.
- [18] NASA. Magnificent cme. <https://svs.gsfc.nasa.gov/11095>, jul 2016.
- [19] NASA. Solar storm and space weather.
http://www.nasa.gov/mission_pages/sunearth/spaceweather/index.html, 2016. accessed in June, 2016.
- [20] Nikon. Dslr nikon. http://www.nikon.co.in/en_IN/product/discontinued/digital-slr-cameras/d700#tech_specs. accessed in August, 2016.
- [21] Royal Observatory of Belgium. Total sunspot numbers (annual).
https://www.quandl.com/data/SIDC/SUNSPOTS_A-Total-Sunspot-Numbers-Annual, aug 2016. accessed in August, 2016.
- [22] Swedish Institute of Space Physics. Magnetometers.
[http://www.irf.se/Observatory/?link\[Magnetometers\]=Description](http://www.irf.se/Observatory/?link[Magnetometers]=Description), aug 2016.
- [23] Bart Ölsthorn. Long term analysis of geomagnetic disturbances. The Hague University, 2015.
- [24] Alan Oppenheim. *Systems and Signals*. Prentice Hall, 1997.
- [25] G. Paschmann, S. Haaland, and R. Trueman, editors. *Aurora plasma physics*, volume 15 of *Space Science Series of ISSI*. Kluwer academic publishers, 2006.
- [26] Ryan. Colour! <http://ryanstutorials.net/css-tutorial/css-colour.php>. accessed in August, 2016.
- [27] Weebly. Earth’s magentic field.
<http://uclmagneticfields.weebly.com/origin-earths-inner-core.html>, aug 2016. They mentioned as reference: <http://www.abc.net.au>.
- [28] Wikipedia. Hsl and hsv. https://en.wikipedia.org/wiki/HSL_and_HSV, jul 2016.
- [29] Elmore William and Heald Mark. *The Physics of Waves*. Dover Publications Inc., 1985.
- [30] M. Yamauchi. Conversation in July.

Appendices

Appendix A

Induction equation

In order to get the induction equation, Faraday's and Ampere's law are needed together with Ohm's law [12].

$$\text{Faraday's law } \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (\text{A.1})$$

$$\text{Ampere's law } \frac{1}{\mu_0} \nabla \times \vec{B} = \vec{J} \quad (\text{A.2})$$

$$\text{Ohm's law } \vec{E} + \vec{v} \times \vec{B} = \frac{\vec{J}}{\sigma} \quad (\text{A.3})$$

where

E	electric field	N/C
B	magnetic field	T
t	time	s
μ_0	vacuum permeability	N/A ²
J	current density	A/m ²
v	velocity	m/s
σ	electrical conductivity	1/ Ωm .

By substituting A.2 in A.3 the following formula is acquired:

$$\vec{E} + \vec{v} \times \vec{B} = \frac{1}{\sigma \mu_0} \nabla \times \vec{B} \quad (\text{A.4})$$

Taking the curl of both sides:

$$\nabla \times \vec{E} + \nabla \times \vec{v} \times \vec{B} = \frac{1}{\sigma \mu_0} \nabla \times \nabla \times \vec{B} \quad (\text{A.5})$$

By applying the following two rules:

$$\nabla \times \nabla \times \vec{B} = \nabla(\nabla \cdot \vec{B}) - \nabla^2 \vec{B} \quad (\text{A.6})$$

And:

$$\nabla \cdot \vec{B} = 0 \tag{A.7}$$

The following formula will be the result:

$$\nabla \times \vec{E} + \nabla \times (\nu \times \vec{B}) = -\frac{1}{\sigma\mu_0} \nabla^2 \vec{B} \tag{A.8}$$

substituting equation A.1 in equation A.8 the wanted formula is achieved:

$$\frac{\partial \vec{B}}{\partial t} = \frac{1}{\sigma\mu_0} \nabla^2 \vec{B} + \nabla \times (\vec{\nu} \times \vec{B}) \tag{A.9}$$

Appendix B

Captured substorm

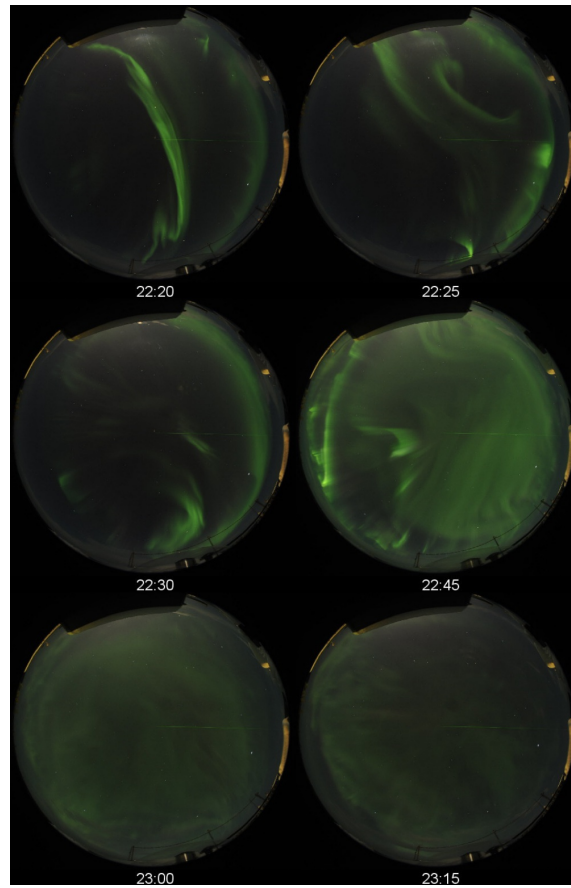


Figure B.1: *A aurora substorm captured by the all-sky camera of IRF at 15 Februari, 2016.*

Appendix C

Switch from RGB to HSL

It is possible to switch between RGB and HSL through calculations [28]. To get from RGB to HSL first the maximum, minimum and difference between R,G and B, which is called the chroma, has to be calculated:

$$M = \max(R, G, B)$$

$$m = \min(R, G, B)$$

$$C = M - m$$

With these variables the hue, saturation and lightness can be determined, for the hue one of the following formulas is needed:

$$H' = 0 \text{ if } (C=0)$$

$$H' = \frac{G - B}{C} \cdot \text{mod}(6) \text{ if } (M=R)$$

$$H' = \frac{B - R}{C} + 2 \text{ if } (M=G)$$

$$H' = \frac{R - G}{C} + 4 \text{ if } (M=B)$$

Where $\text{mod}(6)$ means the modulus of 6, which is the remaining part after $\frac{G-B}{C}$ has been divided by, in this case, 6. The hue itself corresponds than with:

$$H = 60^\circ \cdot H'$$

The lightness is an average of the largest and smallest color of the three components. When the lightness is equal to zero, the pixel is black. Where a white pixel corresponds with a lightness equal to one. The lightness can be determined with:

$$L = \frac{1}{2}(M + m)$$

The last component is the saturation. The saturation is a fit of the chroma so that it would always fit between 0 and 1 and is calculated with:

$$S = 0 \text{ if } (C=0)$$
$$S = \frac{C}{1 - |2L - 1|} \text{ otherwise.}$$

Appendix D

Strong aurora eventlist 2015

Date of event	Time of event	Strength of event	Number of aurora pixels
2015-01-05	21:32:00	91.35	8718
2015-01-21	23:08:00	78.67	37978
2015-01-30	19:42:00	86.03	3183
2015-02-18	03:08:00	66.42	24358
2015-02-22	23:42:00	50.12	20843
2015-03-04	22:37:00	70.05	6478
2015-03-17	23:22:00	74.03	54131
2015-03-18	19:11:00	56.63	17644
2015-09-07	23:19:00	53.27	52772
2015-09-09	22:05:00	68.17	60087
2015-09-10	19:34:00	78.39	9873
2015-09-11	21:28:00	61.93	27468
2015-09-15	23:08:00	63.83	19855
2015-10-06	20:26:00	83.47	82608
2015-10-13	20:46:00	74.97	66198
2015-10-15	19:38:00	52.17	11475
2015-10-29	23:09:00	60.32	12314
2015-11-05	22:47:00	80.64	31289
2015-11-08	23:19:00	87.20	72396
2015-11-27	00:28:00	72.89	4760
2015-11-28	20:49:00	60.61	10282
2015-12-26	21:19:00	93.70	22482

Appendix E

Written programs

E.1 Automatic detection of strong aurora events

Behind the thesis the pages 39 to 44 contain the program that detects strong auroral events, gathers magnetodata and when available, takes the infrasound data.

E.2 Moving Aurora

The second program that detects the direction of the arc has been placed in page 45 to 47.

```

# -*- coding: utf-8 -*-
"""
Created on Mon Jun  6 13:09:16 2016

@author: Dennis
"""

import numpy as np
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import pandas as pd
import urllib
import math
import time
import os
#import datetime

dict = {'Year': 2015, 'Month': 1, 'EndMonth': 12, 'Day': 1, 'EndDay': 31}
dict = {**{'Accumulate': 2500, 'DecayingFactor': 0.997},**dict}
SavePictures = -1 # -1 is no, 0 is only aurora, 1 is yes
ReSheet= 0 # 1 retakes the data, 0 may take the excisting numpy.npy

dict = {**{'SavePictures': SavePictures, 'ReSheet': ReSheet, 'YearString': str(dict['Year'])},**dict}

def Main():
    if not os.path.exists('ProgramData/Needed_Data'):
        os.makedirs('ProgramData/Needed_Data')
    yearurl = ""
    data = ""
    dict['YearList'] = []

    for line in data:
        line = line.decode('utf-8')
        dict['YearList'].append(line)

    while (dict['Month'] <= dict['EndMonth']):
        month()
        dict['Month'] += 1

def month():
    dict['MonthReset'] = 0
    dict['MonthString'] = UpUp(dict['Month'])
    dict['NpSavePath'] = "".join(('ProgramData/Numpy/',dict['YearString'],'/'))
    GatherMagnetodata()

    while(dict['Day'] <= dict['EndDay'] and dict['MonthReset'] == 0):
        dict['TIMEIT'] = time.time()
        Day()
        dict['Day'] += 1

    dict['Day'] = 0

def Day():
    dict['Hour'], dict['LoopHour'] = 12, 0
    dict['Factor'], dict['count'] = 1, 1
    dict['DayString'] = UpUp(dict['Day'])
    dict['NumpyData'] = np.array([[date and time],[L>0.05],[L>0.1],[L>0.2],[relative L],[FinalURL],[Clouds]])
    dict['NumpySaveFilename'] = "".join((dict['NpSavePath'], dict['YearString'],'-', dict['MonthString'],'-',dict['DayString'],'.npy'))
    dict['DateString'] = "".join((dict['YearString'], dict['MonthString'], dict['DayString']))
    dict['ProgramDate'] = "".join((dict['DayString'], '-', dict['MonthString'], '-', dict['YearString']))
    dict['PictureSavePath'] = "".join(('ProgramData/Allsky_images/', dict['YearString'],'/',dict['DateString']))
    SearchDate = [y for y in dict['YearList'] if dict['DateString'] in y]
    if (len(SearchDate) > 0):
        GatherInfrasoundData()
    try:
        if (len(SearchDate) > 0 and os.path.exists(dict['NumpySaveFilename']) is False or len(SearchDate) > 0 and dict['ReSheet'] == 1):
            #If there are pictures found for finaldate, than:
            dict['DayList'] = []
            dict['UrlDate'] = "".join(('http://www.irf.se/allsky/', dict['YearString'], '/', dict['DateString'], '/jpgs/'))
            data = urllib.request.urlopen(dict['UrlDate'])

            for line in data:
                line = line.decode('utf-8')
                dict['DayList'].append(line)

            while (dict['LoopHour'] <= 23):
                Hour()
                dict['Hour'] += 1
                dict['LoopHour'] += 1
            #Here Numpy saves the data in an .npy file

            if (len(dict['NumpyData']) > 1):
                if not os.path.exists(dict['NpSavePath']):
                    os.makedirs(dict['NpSavePath'])
                np.save(dict['NumpySaveFilename'],dict['NumpyData'])

                if (os.path.exists(dict['NumpySaveFilename']) is True):
                    LoadingDataForPlot()
            except:
                print('jpgs folder on irf/allsky/year/date/jpgs not found...')

def Hour():
    if (dict['Hour']>= 24):
        Midnight()

    dict['NumpySaveFileDate'] = "".join((dict['YearString'],'-',dict['MonthString'],'-',dict['DayString']))
    dict['UrFileName'] = "".join(('KRN', dict['DateString'], 'T'))
    dict['Minute'] = 0
    dict['HourString'] = UpUp(dict['Hour'])

    while (dict['Minute']<60):
        Minute()
        dict['Minute'] += 1

    print("----- %3.2f seconds for" % (time.time() - dict['TIMEIT']), dict['NumpySaveFileDate'], ',', dict['HourString'], 'Hour')
    #
    #
def Minute():
    dict['MinuteString'] = UpUp(dict['Minute'])

```



```

dict['SearchFinalFileName'] = """.join(' ', dict['UrlFileName'], dict['HourString'], dict['MinuteString'], '00E'))
FoundFileName = [y for y in dict['DayList'] if dict['SearchFinalFileName'] in y]

if (len(FoundFileName)>0):
    Cut = FoundFileName[0].find(dict['SearchFinalFileName'])
    neededline = FoundFileName[0][Cut+1:]
    Cut = neededline.find('.JPG')
    dict['PictureName'] = neededline[Cut+4:]
    dict['FinalURL'] = """.join((dict['UrlDate'], dict['PictureName'] ))
    dict['NumpySaveFileTime'] = """.join((dict['HourString'], ':', dict['MinuteString']))

    try:
        AllSkyPicture = mpimg.imread(urllib.request.urlretrieve(dict['FinalURL'])[0])/255
        # --- python3 program ---
        # Automatic Aurora identification program for PNG images
        # - v0 by B. Olsthoorn 2015-03 (RGB judgement)
        # - v1 by M. Yamauchi 2016-01 (HLS judgement)
        # - v2 by D. van Dijk 2016-05 (HLS array transition)
        #
        # Convert RGB to HLS for each pixel, and examine H, S, L values.
        # note: HLS means Hue, Lightness, Saturation.
        # The hue component is approximate spectrum color.
        # The lightness component is an easy indicator for the strength of an aurora.
        # note: L = 1.0 means a white pixel, and maximum limit exists
        # Color of aurora in RGB pictures changes with moon and twilight.
        # Therefore, fine tuning is needed for H, S, L ranges.

        RGB_everything_RGB(AllSkyPicture)
        NumpyPreparation()
    except:
        dict['NumpyDateTime'] = """.join((dict['NumpySaveFileDate'], ' ', dict['NumpySaveFileTime']))
        dict['NumpyTimeStamp'] = pd.to_datetime(dict['NumpyDateTime'])
        dict['NumpyData'] = np.hstack((dict['NumpyData'], np.array([[dict['NumpyTimeStamp'],
                                                                [0], [0], [0], [0.0], [dict['FinalURL']], [0]]])))

        print('Could not download image')

    if (dict['SavePictures'] == 1 or dict['CountStrong'] > 2500 and dict['SavePictures'] == 0):
        SavePicturesToDisk()

def UpUp(UP):
    UPstr = str(UP)
    UPlen = len(UPstr)

    if (UPlen == 1):
        UPfinal = """.join(('0', UPstr))
    else:
        UPfinal = UPstr

    return(UPfinal)

def GatherMagnetometerData():
    dict['MagnetometerTIMEIT'] = time.time()
    url = """.join(('http://www.irf.se/maggraphs/iaga/kir', dict['YearString'], dict['MonthString'], 'dmin.min'))
    MagnetometerData = []
    webpage = urllib.request.urlopen(url).readlines()[13:]

    for line in webpage:
        line = line.decode('utf-8').split()
        MagnetometerData.append(line)

    MagnetometerData = np.asarray(MagnetometerData)
    dict['TimestampMagnetometerData'] = np.empty(len(MagnetometerData), dtype="<U25")
    for x in range(len(MagnetometerData)):
        dict['TimestampMagnetometerData'][x] = """.join((MagnetometerData[x,0], ' ', MagnetometerData[x,1]))

    MagnetometerData = MagnetometerData[:,3].astype(np.float)
    # dict['TimestampMagnetometerData'] = pd.to_datetime(dict['TimestampMagnetometerData'])

    res = 11
    MagnetometerSTD = np.zeros((res, MagnetometerData.size//res))

    for i in range(11):
        MagnetometerData2 = MagnetometerData[i:res*((MagnetometerData.size-i)//res)+i]
        STD = np.std(MagnetometerData2.reshape((-1, res)), axis=1)
        MagnetometerSTD[i,:STD.size] = STD

    STDData = MagnetometerSTD.T.flatten()
    dict['STDMagnetometerData'] = np.zeros(dict['TimestampMagnetometerData'].shape)
    dict['STDMagnetometerData'][dict['STDMagnetometerData'].size-STDData.size:] = STDData
    print('Downloaded the magnetometer data from the IRF website in %3.2f seconds' % (time.time() - dict['MagnetometerTIMEIT']))

def GatherInfrasoundData():
    dict['InfrasoundTIMEIT'] = time.time()
    InfrasoundFilenamePart = """.join((dict['YearString'], dict['MonthString'], '/', dict['YearString'][3], dict['MonthString'],
                                         dict['DayString'], '00.caz'))
    InfrasoundFilename = """.join(('ProgramData/Needed_Data/', InfrasoundFilenamePart))

    NextDayData = UpUp(dict['Day']+1)
    InfrasoundFilenamePartNextDay = """.join((dict['YearString'], dict['MonthString'], '/', dict['YearString'][3], dict['MonthString'],
                                         NextDayData, '00.caz'))
    InfrasoundFilenameNextDay = """.join(('ProgramData/Needed_Data/', InfrasoundFilenamePartNextDay))
    dict['InfrasoundChecker'] = 0

    if os.path.exists(InfrasoundFilename) is True:
        Infrasound_Data1 = np.genfromtxt(InfrasoundFilename, dtype=float)
        Infrasound_Data1, InfrasoundTimestampData1 = InfrasoundTimestampGenerator(Infrasound_Data1, dict['DayString'])
        dict['InfrasoundChecker'] = 1

        if os.path.exists(InfrasoundFilenameNextDay) is True:
            Infrasound_Data2 = np.genfromtxt(InfrasoundFilenameNextDay, dtype=float)
            Infrasound_Data2, InfrasoundTimestampData2 = InfrasoundTimestampGenerator(Infrasound_Data2, NextDayData)
            dict['Infrasound_Data'] = np.vstack((Infrasound_Data1, Infrasound_Data2))
            dict['InfrasoundTimestamp'] = np.hstack((InfrasoundTimestampData1, InfrasoundTimestampData2))
        else:
            dict['Infrasound_Data'] = Infrasound_Data1
            dict['InfrasoundTimestamp'] = InfrasoundTimestampData1

```

```

#         dict['InfrasoundTimestamp'] = pd.to_datetime(dict['InfrasoundTimestamp'])

        print('Found and opened the infrasound data of IRF Kiruna in %3.2f seconds.' % (time.time() - dict['InfrasoundTIMEIT']))

    else:
        print('No infrasound data has been found. Place it in the correct folder.')

def InfrasoundTimestampGenerator(Infrasound_Data,Day):
    Requirement_Time      = Infrasound_Data[:,0]
    Requirement_Speed      = Infrasound_Data[:,2]
    Requirement_Corr       = Infrasound_Data[:,4]
    mask                   = (Requirement_Time < 1440) & (Requirement_Speed > 0) & (Requirement_Speed < 999999) & (Requirement_Corr > 0.05)
    Infrasound_Data        = Infrasound_Data[mask, :]

#     idxs = np.any(Infrasound_Data != -1, axis=1)
#     Infrasound_Data        = Infrasound_Data[idxs, :]

    InfrasoundTimestampHours      = np.round(Infrasound_Data[:,0]/60-0.5).astype(int)
    InfrasoundTimestampHours[InfrasoundTimestampHours>24]=24
    InfrasoundTimestampMinutes    = np.round(Infrasound_Data[:,0]%60-0.5).astype(int)
    InfrasoundTimestampMinutes[InfrasoundTimestampMinutes>59]=59
    InfrasoundTimestampSeconds    = np.round((Infrasound_Data[:,0]%60-InfrasoundTimestampMinutes)*60-0.5).astype(int)
    InfrasoundTimestampSeconds[InfrasoundTimestampSeconds>59]=59
    TimestampElement              = 0
    InfrasoundTimestamp           = np.empty(len(Infrasound_Data), dtype="<U15")

    for x in Infrasound_Data:
        InfrasoundTimestamp[TimestampElement] = "".join((dict['YearString'],dict['MonthString'],Day,' ',
                                                         UpUp(InfrasoundTimestampHours[TimestampElement])),
                                                         UpUp(InfrasoundTimestampMinutes[TimestampElement])),
                                                         UpUp(InfrasoundTimestampSeconds[TimestampElement]))

        TimestampElement      += 1
    return(Infrasound_Data, InfrasoundTimestamp)

def Midnight():
    dict['Hour']              = 0
    dict['DayString']         = UpUp(dict['Day']+1)
    dict['DateString']        = "".join((dict['YearString'], dict['MonthString'], dict['DayString']))
    SearchDayInList          = [y for y in dict['DayList'] if dict['DateString'] in y]

    if (len(SearchDayInList) == 0 and dict['Month'] < 12 and dict['Day'] >= 28):
        dict['DayString']     = '01'
        dict['MonthString']   = UpUp(dict['Month']+1)
        dict['MonthReset']    = 1

    elif (len(SearchDayInList) == 0 and dict['Month'] == 12 and dict['Day'] == 31):
        dict['DayString']     = '01'
        dict['MonthString']   = '01'
        dict['YearString']    = UpUp(dict['Year']+1)
        dict['MonthReset']    = 1

    dict['DateString']        = "".join((dict['YearString'], dict['MonthString'], dict['DayString']))

def RGB_everything_RGB(picca):
    """
    @rgb to hls author: Paul, edited by: Dennis and SailAvid
    @hls to rgb author: SailAvid, edited by: Dennis
    """

    def Stitch_Plot(image1,image2,image3,image4):
        totalImg          = np.hstack((image1,image2,image3,image4))
        return(totalImg)

    def HLS_to_RGBfig(H2,L2,S2):
        C                  = (1-np.absolute(2*L2-1))*S2
        H2                 *= 6
        X                  = C*(1-np.absolute(np.mod(H2,2)-1))
        R                  = np.zeros(H2.shape, float)
        G                  = np.zeros(H2.shape, float)
        B                  = np.zeros(H2.shape, float)
        mask = (H2>=0)     == (H2 < 1)
        R[mask],G[mask]    = C[mask], X[mask]
        mask = (H2>=1)     == (H2 < 2)
        R[mask],G[mask]    = X[mask], C[mask]
        mask = (H2>=2)     == (H2 < 3)
        G[mask],B[mask]    = C[mask], X[mask]
        mask = (H2>=3)     == (H2 < 4)
        G[mask],B[mask]    = X[mask], C[mask]
        mask = (H2>=4)     == (H2 < 5)
        R[mask],B[mask]    = X[mask], C[mask]
        mask = (H2>=5)     == (H2 < 6)
        R[mask],B[mask]    = C[mask], X[mask]
        m                  = L2 - 0.5 * C
        R, G, B            = R+m, G+m, B+m
        rgb                = np.zeros((R.shape[0],R.shape[1],3), 'float')
        rgb[..., 0], rgb[..., 1], rgb[..., 2] = R, G, B
        return(rgb)

    R2, G2, B2            = picca[:, :, 0], picca[:, :, 1], picca[:, :, 2]
    M                      = np.amax(picca, axis=2)
    m                      = np.amin(picca, axis=2)
    C                      = M-m #chroma
    Cmsk                   = C!= 0
    #Hue:
    H                      = np.zeros(R2.shape, float)
    mask                   = (M==R2)&Cmsk
    H[mask]                = np.mod((G2[mask]-B2[mask])/C[mask], 6)
    mask                   = (M==G2)&Cmsk
    H[mask]                = ((B2[mask]-R2[mask])/C[mask] + 2) * 41
    mask                   = (M==B2)&Cmsk
    H[mask]                = ((R2[mask]-G2[mask])/C[mask] + 4)
    H                      /= 6
    #Lightness:
    L                      = 0.5*(M+m)
    #Saturation:
    S                      = np.zeros(R2.shape, float)

```

```

S[Cmsk] = ((C[Cmsk])/(1-np.absolute(2*L[Cmsk]-1)))
S[S>1.0] = 1
#HLS filter
mask = (H>0.16) & (H<0.50) & (S>0.10) & (S<0.8) & (L>0.05) & (L<0.8)
Hmaybe, Smaybe, Lmaybe = np.zeros(H.shape,float), np.zeros(S.shape,float), np.zeros(L.shape,float)
Hmaybe = H * mask
Smaybe = S * mask
Lmaybe = L * mask
# LumMaybe = np.sum(Lmaybe)
dict['CountMaybe'] = np.count_nonzero(Hmaybe)
mask = (H>0.18) & (H<0.46) & (S>0.15) & (S<0.8) & (L>0.10) & (L<0.8)
Hyes, Syes, Lyes = np.zeros(H.shape,float), np.zeros(S.shape,float), np.zeros(L.shape,float)
Hyes = H * mask
Syes = S * mask
Lyes = L * mask
# LumYes = np.sum(Lyes)
dict['CountYes'] = np.count_nonzero(Hyes)
mask = (H>0.20) & (H<0.46) & (S>0.25) & (S<0.8) & (L>0.20) & (L<0.8)
Hstrong, Sstrong, Lstrong = np.zeros(H.shape,float), np.zeros(S.shape,float), np.zeros(L.shape,float)
Hstrong = H * mask
Sstrong = S * mask
Lstrong = L * mask
dict['LumStrongAccumulate'] = np.sum(np.sort(Lstrong, None)[-dict['Accumulate']:])*dict['Factor']

# LumStrong = np.sum(Lstrong)
dict['CountStrong'] = np.count_nonzero(Hstrong)

mask = (H>0.00) & (H<0.16) & (S>0.10) & (S<0.8) & (L>0.15) & (L<0.80)
Hcloud = np.zeros(H.shape,float)
Hcloud = H * mask
dict['CountCloud'] = np.count_nonzero(Hcloud)

if (dict['CountStrong'] > dict['Accumulate']):
    dict['count'] += 1
    if (dict['count'] >= 15):
        dict['Factor'] *= dict['DecayingFactor']
else:
    dict['Factor'] = 1
    dict['count'] = 1

if (dict['SavePictures'] == 1 or dict['CountStrong'] > 2500 and dict['SavePictures'] == 0):
    rgbmaybe = HLS_to_RGBfig(Hmaybe,Lmaybe,Smaybe)
    rgbyes = HLS_to_RGBfig(Hyes,Lyes,Syes)
    rgbstrong = HLS_to_RGBfig(Hstrong,Lstrong,Sstrong)
    dict['FinalImg'] = Stitch_Plot(picca,rgbmaybe,rgbyes,rgbstrong)

return()

def NumpyPreparation():
    dict['AverageLumS'] = dict['LumStrongAccumulate'] / dict['Accumulate'] * 100

    if (math.isnan(dict['AverageLumS'])) is True or (dict['CountStrong'] < dict['Accumulate']):
        dict['AverageLumS'] = 0.0

    dict['NumpyDateTime'] = ''.join((dict['NumpySaveFileDate'], ' ', dict['NumpySaveFileTime']))
    dict['NumpyTimeStamp'] = pd.to_datetime(dict['NumpyDateTime'])
    dict['NumpyData'] = np.hstack((dict['NumpyData'],np.array([[dict['NumpyTimeStamp']],dict['CountMaybe']],
                                                            [dict['CountYes']],dict['CountStrong']],dict['AverageLums'],
                                                            [dict['FinalURL']],dict['CountCloud']]])))

def SavePicturesToDisk():
    if not os.path.exists(dict['PictureSavePath']):
        os.makedirs(dict['PictureSavePath'])

    savefile = ''.join((dict['PictureSavePath'],'/',dict['PictureName'] ))
    fig = plt.imshow(dict['FinalImg'])
    fig.axes.get_xaxis().set_visible(False)
    fig.axes.get_yaxis().set_visible(False)
    plt.axis('off')
    plt.savefig(savefile, dpi=500, bbox_inches='tight', pad_inches=0.0)
    plt.cla()

def LoadingDataForPlot():
    dict['NumpyData'] = np.load(dict['NumpySaveFilename'])
    PlotImage(dict['TimestampMagnetodata'],dict['STDmagnetodata'],pd.to_datetime(dict['NumpyData'][0,1:]),
              dict['NumpyData'][2,1:].astype(np.float64),dict['NumpyData'][3,1:].astype(np.float64)
              ,dict['NumpyData'][4,1:].astype(np.float64),dict['NumpyData'][5,1:],dict['NumpyData'][6,1:])

def PlotImage(Datax,Datay,Data2x,Data2y,Data3y,Data4y,ProcessedURL,Data5y):
    if (dict['InfrasoundChecker'] == 1):
        Infrasound_Data = dict['Infrasound_Data'][:,1]
        InfrasoundPlotTimestamp = pd.to_datetime(dict['InfrasoundTimestamp'])
        Datax = pd.to_datetime(dict['TimestampMagnetodata'])
        Data4y = (Data4y-20)*(100/60)
        # dstart = datetime.datetime(2016,1,1,00,00)
        # dend = datetime.datetime(2016,1,31,23,59)
        # start = serie_1.index.searchsorted(Data2x[0])
        # end = serie_1.index.searchsorted(Data2x[-1])+1
        # rng = pd.date_range(dstart, dend, freq = 'Min')
        # serie_1 = serie_1.reindex(rng)
        #
        #
        # plot = serie_1.ix[start:end]
        # plot = plot.reset_index().values

        color = '#ffffff'
        color1 = '#9a9a9a'
        color2 = '#515151'
        color3 = '#ccc650'
        fontsize = 6

    fig = plt.figure(facecolor=color,figsize = (10,15))
    Label1 = 'Standard deviation'
    Label2 = 'Probably'
    Label3 = 'Aurora'
    Label4 = str(dict['Accumulate'])+' strongest'
    Label5 = 'Clouds'

```

```

Label6                                = 'Trace Velocity'

ax1                                   = plt.subplot2grid((26,4), (0,0), rowspan=4, colspan=4, axisbg=color)
ax1.plot(Datax,Datay,'k-',label=Label1, linewidth=1)
plt.title('Standard deviation of the magnetometer data at %s' %dict['ProgramDate'])
plt.ylabel('The standard deviation')
for tl in ax1.get_yticklabels():
    tl.set_color('k')
ax1.legend(loc='upper right', fontsize= 'small')
plt.grid()
plt.ylim([0,150])
plt.xticks(rotation=0, fontsize=fontsize)
plt.yticks(rotation=0, fontsize=fontsize)

ax2                                   = plt.subplot2grid((26,4), (4,0), sharex=ax1, rowspan=4, colspan=4, axisbg=color)
ax2.plot(Data2x,Data2y,color1,label=Label2, linewidth=1.5)
ax2.plot(Data2x,Data3y,color2,label=Label3, linewidth=1.5)
ax2.plot(Data2x,Data5y,color3,label=Label5, linewidth=1.5)
plt.ylabel('Number of counts')
ax2.locator_params(axis='y',nbins=5)

for tl in ax2.get_yticklabels():
    tl.set_color('k')
ax2.legend(loc='upper right', fontsize= 'small')
plt.grid()
plt.xticks(rotation=0, fontsize=fontsize)
plt.yticks(rotation=0, fontsize=fontsize)

ax3                                   = plt.subplot2grid((26,4), (8,0), sharex=ax1, rowspan=4, colspan=4, axisbg=color)
ax3.plot(Data2x,Data4y,color2,label=Label4, linewidth=1.5)
ax3.set_ylim([0, 100])

plt.grid()
plt.xticks(rotation=0, fontsize=fontsize)
plt.yticks(rotation=0, fontsize=fontsize)
plt.ylabel('Relative strength of the aurora')
ax3.legend(loc='upper right', fontsize= 'small')
Begin = pd.Timestamp(''.join((dict['NumpySaveFilename'][-14:-4], ' 16:00')))
End = Begin + pd.Timedelta('14 hours')

try:
    MaximumStrong = np.argmax(Data4y)
    ColumLoop=0
    RowLoop=13

    #CREATING LIST OF EVENTS
    if os.path.exists('ProgramData/Eventlist.txt'):
        List = np.genfromtxt("ProgramData/Eventlist.txt", dtype=str, delimiter='\n')
        FirstDataPoint = 0
    else:
        List = 'Date\t\tTime\t\tPixels\tStrength'
        FirstDataPoint = 1
    CHECKEVENT = 0
    #-----

    while (Data4y[MaximumStrong]>50 and ColumLoop <= 2):
        X_Time = Data2x[MaximumStrong]
        X_Axis = (X_Time - Begin) / pd.Timedelta('14 hours')
        Y_Axis = (Data4y[MaximumStrong])/100
        ax3.annotate('Strong',xy = (X_Axis,Y_Axis), xycoords='axes fraction',
                    xytext=(15,0), textcoords='offset points',
                    bbox={'boxstyle':'round4', 'fc':'w'},
                    arrowprops={'facecolor':'k', 'arrowstyle':'->', 'relpos':(0.1, 0.8)},
                    fontsize=6, color = 'k',
                    horizontalalignment='left', verticalalignment='bottom')

        AllSkyPictureMax = mpimg.imread(urllib.request.urlretrieve(ProcessedURL[MaximumStrong])[0])[0,:90:630,:])
        plt.subplot2grid((26,4), (12,ColumLoop), rowspan=6, colspan=1)
        plt.imshow(AllSkyPictureMax)
        ax = plt.gca()
        # ax4.imshow(AllSkyPictureMax)
        ax.axes.get_xaxis().set_visible(False)
        ax.axes.get_yaxis().set_visible(False)

        plt.subplot2grid((26,4), (12,ColumLoop+1), rowspan=6, colspan=1)
        ax = plt.gca()
        ax.axis([0, 10, 0, 10])
        ax.axes.get_xaxis().set_visible(False)
        ax.axes.get_yaxis().set_visible(False)
        ax.axis('off')
        ax.text(0, 9, 'Peak:', fontsize=6, style='italic', weight='bold')
        ax.text(0, 7.5, pd.to_datetime(dict['NumpyData'][0,MaximumStrong+1]), fontsize=6, style='italic')
        ax.text(0, 6, '2500 pixels: %0f %%' %Data4y[MaximumStrong], fontsize=6, style='italic')

        ColumLoop += 2
        RowLoop = 19

    #CREATING EVENT FOR THE LIST OF EVENTS
    if (ColumLoop ==2 ):
        NewEvent = ''.join((str(dict['NumpyData'][0,MaximumStrong+1])[10:],'t',
                            str(dict['NumpyData'][0,MaximumStrong+1])[11:],'t',
                            str(dict['NumpyData'][4,MaximumStrong+1]-20)*(100/60))[5:],'t',
                            str(dict['NumpyData'][3,MaximumStrong+1])))
        print(NewEvent)
        if (FirstDataPoint == 1):
            CHECKEVENT = 1
            List = np.hstack((List,NewEvent))
        elif (max(np.core.defchararray.rfind(NewEvent,List)) < 0):
            CHECKEVENT = 1
            List = np.hstack((List,NewEvent))
        #-----

        Data4y[MaximumStrong-10:MaximumStrong+10] = 0
        MaximumStrong = np.argmax(Data4y)

#SAVING THE EVENTS IN THE LIST

if (CHECKEVENT == 1):

```

```

        List[1:] = np.sort(List[1:])
        File = open("ProgramData/EventList.txt", 'w')
        for Sentence in List:
            Sentence = "".join((Sentence, '\n'))
            File.write(Sentence)
        File.close
    #-----
except:
    print('Could not save an event or find maximum in graph.')

for tl in ax3.get_yticklabels():
    tl.set_color('k')

if (dict['InfrasoundChecker'] == 1):
    ax4 = plt.subplot2grid((26,4), (RowLoop,0), sharex=ax1, rowspan=7, colspan=4, axisbg=color)
    ax4.plot(InfrasoundPlotTimestamp,Infrasound_Data,'xk',color=color2,label=Label6, linewidth=1.5)
    ax4.set_ylim([0, 360])
    ax4.set_title('Infrasound trace velocity.', fontsize = fontsize+2)
    ax4.legend(loc='upper right', fontsize= 'xx-small')
    plt.xticks(rotation=0, fontsize=fontsize)
    plt.yticks(np.arange(0, 361, 90.0),rotation=0, fontsize=fontsize)

ax3.set_xlim(Begin, End)

plt.subplots_adjust(left=.10, bottom=.05, right=.96, top=.98, wspace=.2, hspace=8)

filename = "".join(('ProgramData/Graphs/', dict['NumpySaveFilename'][-19:-4], '.jpg'))
if not os.path.exists(filename[0:24]):
    os.makedirs(filename[0:24])
plt.savefig(filename,dpi=250,facecolor=fig.get_facecolor())
plt.show()
print('This plot is also saved on the hard disc')
fig.clf

```

Main()

```

# -*- coding: utf-8 -*-
"""
Created on Fri Aug 12 11:05:41 2016

@author: Dennis
"""
import numpy as np
import matplotlib.image as mpimg
import matplotlib.pyplot as plt
import urllib
plt.close('all')

def RGB_everything_RGB(picca):
    """
    @rgb to hls author: Paul, edited by: Dennis and SailAvid
    @hls to rgb author: SailAvid, edited by: Dennis
    """
    def HLS_to_RGBfig(H2,L2,S2):
        C = (1-np.absolute(2*L2-1))*S2
        H2 = H2 * 6
        X = C*(1-np.absolute(np.mod(H2,2)-1))
        R = np.zeros(H2.shape, float)
        G = np.zeros(H2.shape, float)
        B = np.zeros(H2.shape, float)
        mask = (H2>=0)
        R[mask],G[mask] = C[mask], X[mask]
        mask = (H2>=1)
        R[mask],G[mask] = X[mask], C[mask]
        mask = (H2>=2)
        G[mask],B[mask] = C[mask], X[mask]
        mask = (H2>=3)
        G[mask],B[mask] = X[mask], C[mask]
        mask = (H2>=4)
        R[mask],B[mask] = X[mask], C[mask]
        mask = (H2>=5)
        R[mask],B[mask] = C[mask], X[mask]
        m = L2 - 0.5 * C
        R, G, B = R+m, G+m, B+m
        rgb = np.zeros((R.shape[0],R.shape[1],3), 'float')

```

```

    rgb[..., 0], rgb[..., 1], rgb[..., 2] = R, G, B

R2, G2, B2          = picca[:, :, 0], picca[:, :, 1], picca[:, :, 2]
M                    = np.amax(picca, axis=2)
m                    = np.amin(picca, axis=2)
C                    = M-m #chroma
Cmsk                 = C!= 0
#Hue:
H                    = np.zeros(R2.shape, float)
mask                 = (M==R2)&Cmsk
H[mask]              = np.mod((G2[mask]-B2[mask])/C[mask], 6)
mask                 = (M==G2)&Cmsk
H[mask]              = ((B2[mask]-R2[mask])/C[mask] + 2)
mask                 = (M==B2)&Cmsk
H[mask]              = ((R2[mask]-G2[mask])/C[mask] + 4)
H                    /= 6
#Lightness:
L                    = 0.5*(M+m)
#Saturation:
S                    = np.zeros(R2.shape, float)
S[Cmsk]              = ((C[Cmsk])/(1-np.absolute(2*L[Cmsk]-1)))
S[S>1.0]             = 1

mask                 = (H>0.20) & (H<0.46) & (S>0.25) & (S<0.8) & (L>0.20) & (L<0.8)
Hstrong, Sstrong, Lstrong = np.zeros(H.shape, float), np.zeros(S.shape, float), np.zeros(L.shape, float)
Hstrong              = H * mask
Lstrong              = L * mask
Sstrong              = S * mask
HLS_to_RGBfig(Hstrong, Lstrong, Sstrong)
return(Hstrong, Lstrong, Sstrong)

NumpyData            = np.load('ProgramData/Numpy/2015/2015-01-21.npy')
NumpyD               = NumpyData[:, 1:]
Maximum              = np.argmax(NumpyD[4, :]) - 1
NumpyD               = NumpyD[:, Maximum:Maximum+3]
mask                 = (NumpyD[4, :] != 0.0)
NumpyD               = NumpyD[:, mask]

```

```

"""
Fast numpy calculations for weighted average on each row
"""
P = ['23:07', '23:08', '23:09']
fig = plt.figure()
plt.ylabel('Pixel number of the images')
plt.xlabel('Pixel number of the images')
plt.grid()
for count in range(len(NumpyD[5,:])):
    FinalURL = NumpyD[5,count]
    AllSkyPicture = mpimg.imread(urllib.request.urlretrieve(FinalURL)[0])/255
    H,L,S = RGB_everything_RGB(AllSkyPicture)
    if (count == 0):
        Ltest = np.zeros(L.shape)
    Largsort = np.argsort(L, axis = 1)
    Lsort = np.sort(L, axis = 1)
    Laverage = np.ma.average(Largsort, axis=1, weights=Lsort)
    LFLIPY = Laverage[::-1]

    plt.plot(LFLIPY, range(479), label = '%s' %P[count])
    plt.legend()

    for x in range(len(Laverage)):
        if Laverage[x] != 0:
            z = int(Laverage[x])
            Ltest[x,z] = 1/(len(NumpyD[4,:])+1)*(count+1)
    plt.ylim([0,479])
    plt.xlim([0,720])
    Yaxis = np.sum(Ltest,axis=1)
    if count == 0:
        Sticked = AllSkyPicture
    else:
        Sticked = np.hstack((Sticked,AllSkyPicture))

fig = plt.figure(figsize=(8,6))
plt.imshow(Sticked)

```


The Hague University of applied sciences

Faculty for Technology, Society & Innovation Delft

ASSESSMENT FORM

Name of the student: Dennis van Dijk

Year: 2016 Period: 4

Company/Institute: Swedish Institute of Space Physics

Department: STP

Supervisor: Yamauchi Masatoshi

Please indicate below your evaluation of the student:

	E	G	A	U	
1. Grasp of assignment	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
2. Accuracy of work	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
3. Diligence	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
4. Initiative	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
5a. Cooperation with mentor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
5b. Cooperation with others	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
6. Independence	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
7. Planning	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
8a. Written reporting					
- technical content	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
- usefulness	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
8b. Oral reporting					
- presentation	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
- ability to express	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
9. Attitude as employee	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Overall assessment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____

E= excellent, G=good, A=acceptable, U=unacceptable

Comments on discussion with student: He is very good at questioning of what he did not understand or grasp. We welcome him to back to us.

Date: 25/8/2016 Signature mentor [Signature] Signature student DvDijk

The Hague University of applied sciences

Faculty for Technology, Society & Innovation Delft

ASSESSMENT FORM

Name of the student: Dennis van Dijk Year: 2016 Period April-August

Company/Institute: Swedish Institute of Space Physics Department: _STP

Supervisor: Masatoshi Yamauchi (co-supervisor: Johan Kero)

Please indicate below your evaluation of the student:

	E	G	A	U	
1. Grasp of assignment	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
2. Accuracy of work	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
3. Diligence	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
4. Initiative	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
5a. Cooperation with mentor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
5b. Cooperation with others	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
6. Independence	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
7. Planning	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
8a. Written reporting					
- technical content	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
- usefulness	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
8b. Oral reporting					
- presentation	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
- ability to express	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
9. Attitude as employee	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____
Overall assessment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	_____

E= excellent, G=good, A=acceptable, U=unacceptable

Comments on discussion with student:

He is very good as questioning of what he could not grasp. We welcome him to come back for master thesis if he wants (we are thus happy with the work).