Indian Academy of Sciences, Bangalore
Indian National Academy of Sciences, New Delhi
The National Academy of Sciences India, Allahabad

SUMMER RESEARCH FELLOWSHIPS — 2013

Format for the final Report*

Name of candidate : Rohit Venkat Gandhi Mendadhala

Appl. registration no. : ENGS 6427

Date of joining : 10th May 2013

Date of completion : 3rd July 2013

Name of Guide : Raaj Ramsankaran

Guide's institution : IIT Bombay

Project title : Development of computer programs for
1. Estimation of LST from Landsat TM data
2. Estimation of Ground Surface Reflectance from Landsat TM data.

Address with pin code to which the certificate could be sent:
Flat no: C-4, Aakriti Apts, A.U Road, Balaji Nagar.
Siripuram, Visakhapatnam-530003. Andhra Pradesh.

E-mail ID: rvg296@gmail.Com

Phone No: +91 9030979298

Rohit
Signature of candidate
Date: 4/7/2013

Signature of guide
Date: 4/7/2013

*The final report could be anywhere between 20 and 25 pages including tables, figures etc.
This format should be the first page of the report and should be stapled with the main report.

(For office use only; do not fill/tear)

Candidate's name:	Fellowship amount:
Student: Teacher:	Deduction:
Guide's name:	TA fare:
KVPY Fellow: INSPIRE Fellow:	Amount to be paid:
Others	A/c holder's name:

SUMMER RESEARCH FELLOWSHIP PROGRAMME 2013

FINAL REPORT

ON

DEVELOPMENT OF COMPUTER PROGRAMS FOR

1. ESTIMATION OF LAND SURFACE TEMPERATURE FROM LANDSAT THEMATIC MAPPER TIR DATA

2. ESTIMATION OF GROUND SURFACE REFLECTANCE FROM LANDSAT THEMATIC MAPPER OPTICAL DATA

Submitted By

M.Rohit Venkat Gandhi
(ENGS6427)

Under the Guidance of
Dr. RAAJ Ramsankaran

Department of Civil Engineering,
Remote Sensing & Geodesy Division,
Indian Institute of Technology, Bombay
Powai, Mumbai- 400076

MAY-JUNE 2013

DECLARATION SHEET

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

M.Rohit Venkat Gandhi

(ENGS6427)

ACKNOWLEDGEMENTS

I express whole-heartedly my deep sense of gratitude and regards to my project guide, Dr.RAAJ Ramsankaran, Department of Civil Engineering, for his valuable guidance, persistent encouragement, constant motivation, thoughtful discussion and supervision throughout the course of this work.

I would like to thank all my Post-Doctoral Research Scholars, MTech students of Remote sensing Division, Department of Civil Engineering, IIT Bombay for their guidance and encouragement throughout this work

I would like to thank Department of Civil Engineering, Central library and Computer Centre at IIT Bombay for their kind help and cooperation throughout this work.

I would like to specially thank Indian Academy of sciences, Bangalore; Indian National Academy of Sciences, New Delhi; The National Academy of Sciences India, Allahabad for giving me this wonderful opportunity to work under the guidance of Dr.RAAJ Ramsankaran.

I would like to thank Prithvi Ambati who has given valuable guidance for the computer programming part of this project without whom it would have been difficult to achieve the results.

I would also like to thank my colleagues for healthy discussion on different issues relating to this report and last but not least I would like to express my heartfelt thanks to my beloved parents for their blessings, my friends/classmates for their help and wishes for the successful completion of this project.

ABSTRACT

Land surface Temperature (LST) and Ground surface reflectance (GSR) are some of the important parameters in the field of remote sensing whose estimation is necessary. The former is used to estimate the temperature of a particular chosen surface which the satellite sensor sees through and the latter helps in finding out the characteristics of the objects or features or targets under consideration. The manual measurement of both the parameters through the acquisition of field data is tedious and time consuming where as many satellites provides a data which covers huge areas and are free of cost, but these are available in the form of Digital Number (DN). Hence there is a need to convert these DN values to LST and GSR using suitable band data for further applications.

In the present work a methodology has been adopted which integrates the concept of computer programming and Digital Number conversion to estimate both the parameters. In the case of ground surface reflectance estimation, the digital numbers are corrected for atmospheric effects and provided as input. This methodology was applied for a particular study area by acquiring its satellite imagery bands (thermal and optical bands) and the results have been achieved and studied. The results found were adequate to check out the land surface temperature and ground reflectance values of each and every pixel of the Digital Satellite imagery bands. Hence this methodology has a vast application in finding out the temperature and reflectance in a systematic and simple way, where the manual measurement through field data acquisition is difficult.

Contents

DECLARATION SHEET ii
ACKNOWLEDGEMENT iii
ABSTRACT iv
LIST OF TABLES vii
LIST OF FIGURES vii
1. INTRODUCTION 1
1.1 Background 1
1.2 Brief overview of LANDSAT programme 1
1.3 The Thematic Mapper 2
1.3.1 TM Bands 2
1.3.2 TM Technical Specifications 2
1.5 LANDSAT Applications 3
1.4 Objectives 3
2. METHODOLOGY ADOPTED 4
2.1 Need for Converting Digital Numbers to Radiance. 4
2.2 Digital Number (DN) to Temperature Conversion 4
2.3 Digital Number (DN) to Ground surface reflectance conversion. 5
2.4 Radiometric Pre-processing: 6
2.5 Dark Object Subtraction Technique: 6
2.5.1 Advantages of DOS: 8
2.5.2 Disadvantages of DOS: 8
2.6 Input Parameters and their dependence: 8
3. STUDY AREA AND DATA USED 9
3.1 Study Area: 9
3.2 Data Used: 9
4. ALGORITHM IMPLEMENTATION 11
4.1 Landsat Scene Details: 11
4.2 Step by step procedure for implementing algorithm: 11
4.3 ASCII Input: 12
4.3.1 About Commons IO 13
4.4 Softwares Used: 13

4.4.1 NetBeans: 13
4.4.2 GRASS GIS: 13
4.4.3Arc GIS: 14
5. PROGRAMMING DETAILS 15
5.1 Process involved in Digital Number to Temperature Conversion: 15
5.1.1 Constants used for Digital Number to Temperature Conversion: 15
5.1.2 Java code for Digital Number to Temperature Conversion: 15
5.1.3 Result: 18
5.1.4 Output file on desktop (Initial part) 19
5.1.5 Output File on Desktop (other part) 19
5.2 Process involved in Digital Number to Reflectance Conversion 20
5.2.1 Constants used for Digital Number to Ground Reflectance conversion 20
5.2.2 Java Code for Digital Number to Ground surface reflectance conversion 20
5.2.3 Result: 24
5.2.4 Output file on desktop (initial part): 25
5.2.5 Output File on Desktop (other part): 25
6. RESULTS AND DISCUSSIONS 26
7. CONCLUSIONS 27
8. REFERENCES 28

LIST OF TABLES

Table 1: TM Band Wavelengths and Resolution...2

LIST OF FIGURES

Figure 1: Histogram minimum method for correction of atmospheric effects...7

Figure 2 : Band 6 (Thermal Band of Data used)..10

Figure 3 : Band 2 (Green) (Optical band of the data used) 10

Figure 4: Raster to ASCII conversion of Thermal Band (Band 6) 12

Figure 5: Raster to ASCII conversion of Optical Band Green (Band 2) 12

Figure 6 : Image showing the different products of Arc GIS...14

Figure 7: Properties file showing the constant values used for DN to temperature conversion ..15

Figure 9: Output ASCII file showing the temperature values in °C of corresponding DN values of pixels other than zero which is generated after converting DN to Temperature. 19

Figure 10: Properties file showing the constant values used for DN to reflectance conversion...20

Figure 11: Output ASCII file showing the header format and unchanged zero reflectance values of corresponding DN...25

Figure 12: Output ASCII file showing reflectance values of corresponding DN values of pixels other than zero which is generated after converting DN to reflectance. 25

Figure 13: GRASS GIS Layer manager output window showing temperature values (highlighted) in degrees centigrade. Temperature values of the pixels in the left side window are shown whenever clicked upon a certain pixel on the right side thermal band (band 6) by using in GRASS Map Display. 26

Figure 14: GRASS GIS Layer Manager output window showing ground surface reflectance values(Highlighted).Ground surface reflectance values of the pixels in the left side window are shown whenever clicked upon a certain pixel on the right side optical band (band 2-green) by using in GRASS Map Display...26

1. INTRODUCTION

1.1 Background:

Land Surface Temperature (**LST**) is an important parameter in the surface energy budget and it is regulated and controlled by the complex interplay of topography, incoming radiation, atmospheric processes, as well as by the soil moisture distribution and the different land covers and vegetation types. From a satellite's point of view, the "surface" is whatever it sees when it looks through the atmosphere to the ground. It could be snow and ice, the grass on a lawn, the roof of a building, or the leaves in the canopy of a forest. Thus, land surface temperature is not the same as the air temperature that is included in the daily weather report.

The LST measurement generally becomes tedious while adopting manual methodology such as through acquisition of field data. For example if we take the Himalayan Region separating Indian Subcontinent from Tibetan Plateau, the automated weather stations are absent, without which the measurement of temperature of glaciers becomes difficult. Therefore Thermal Remote sensing acts as an indispensable tool for measuring temperature.

Reflectance is one of the important parameter in Remote Sensing using which we can estimate the properties or characteristics of the objects. Reflectance can be defined as the ratio of the radiation reflected from an object to the amount of radiation incident upon that object. Calculation of reflectance from the field work for large areas can be tedious and consumes lot of time. Hence, satellite Remote sensing data can be used for estimating the reflectance of various Earth features. In satellite remote sensing, reflectance is of two kinds: **Top of Atmospheric (TOA) Reflectance** and **Surface Reflectance**. TOA reflectance represents the solar radiation incident on the instrument at top of the atmosphere. Surface reflectance is the reflectance of the features at the ground level. To obtain the surface reflectance from remote sensing, TOA reflectance should be estimated and corrected for atmospheric effects.

Every Digital Image is composed of a two-dimensional array of numbers. Each cell of digital image is called pixel. The number representing the brightness value of a pixel is called **Digital Number (DN)**. DN value ranges from 0 to 2^n, where n represents the no. of bit resolution (8bit, 10 bit etc.).Satellite imagery generally have third dimension known as layers or bands containing different information. By combining the information from all these bands we get the required satellite image in 3D. In the Satellite imagery digital numbers play an important role estimating the temperature and reflectance values combined with the information derived from the satellite sensor like calibration constants, radiance values which can be obtained from the satellite header file.

1.2 Brief overview of LANDSAT programme:

Landsat, formerly called ERTS (Earth Resource Technology Satellites) consists of series of satellites which are operated by NASA with the help of U.S Department of Interior. Till now six Landsat satellites have been launched successfully, out of which LANDSAT-6 was a launch failure. There have been four types of sensors generally included in these satellite missions namely Multi Spectral scanner (MSS) systems, Return Beam Vidicon (RBV) systems, Thematic Mapper(TM) and Enhanced Thematic Mapper (ETM). We have acquired Thermal Infrared (TIR) data and optical data from Landsat Thematic Mapper.

1.3 The Thematic Mapper:

The Thematic Mapper (TM) is an advanced, multispectral scanning, Earth resources sensor designed to achieve higher image resolution, sharper spectral separation, improved geometric fidelity and greater radiometric accuracy and resolution than the MSS sensor. Like MSS, the TM uses a fixed set of detectors for each band and an oscillating mirror. TM data are sensed in seven spectral bands simultaneously. Band 6 senses thermal (heat) infrared radiation. Landsat can only acquire night scenes in band 6. A TM scene has an Instantaneous Field Of View (IFOV) of 30m x 30m in bands 1-5 and 7 while band 6 has an IFOV of 120m x 120m on the ground. The Thematic Mapper instrument is the primary imaging sensor carried by Landsat-4 and Landsat-5. TM is also an Optical Mechanical Scanner, similar to MSS; however being a 2nd generation line scanning sensor, it ensures better performance characteristics in terms of (i)Improved pointing accuracy and stability, (ii) high resolution (iii) new and more number of spectral bands (iv)16 days repetitive coverage(v) high scanning efficiency using bi-directional scanning and (vi) increased Quantization levels. For achieving the bi-directional scanning, a scan line corrector (SLC) is introduced. Between the telescope and focal plane. The SLC ensures parallel lines of scanning in the forward and reverse direction.

1.3.1 TM Bands:

Table 1: TM Band Wavelengths and Resolution (Source:http://landsat.gsfc.nasa.gov/about/tm.html)

Band Number	µm	Resolution
1	0.45-0.52	30 m
2	0.52-0.60	30 m
3	0.63-0.69	30 m
4	0.76-0.90	30 m
5	1.55-1.75	30 m
6	10.4-12.5	120 m
7	2.08-2.35	30 m

1.3.2 TM Technical Specifications:

(Source: http://landsat.gsfc.nasa.gov/about/tm.html)

- Sensor type: Opto-mechanical
- Spatial Resolution: 30 m (120 m - thermal)
- Spectral Range: 0.45 - 12.5 µm
- Number of Bands: 7
- Temporal Resolution: 16 days
- Image Size: 185 km X 172 km
- Swath: 185 km
- Programmable: yes

1.4 LANDSAT Applications:

(Source: http://landsat.gsfc.nasa.gov/about/appl.html)

Landsat observations have gained wide importance and acceptance throughout the science and application communities over the program's life time. Early applications of Landsat were more confined to the science of remote sensing , but today's use made it not only a fundamental source for addressing science equations but also served as a valuable resource for decision makers in diverse fields like agriculture, land use, forestry, water resources and natural resources exploration. Over the past few decades, Landsat has also played an important role in diverse applications such as human population census, growth of global urbanization and deletion of coastal wetlands. NASA's Land-Cover and Land-Use Change Program (LCLUC) uses Landsat data to develop socially relevant interdisciplinary science that can be applied to natural resource management questions, starting with agricultural land use change.

1.5 Objectives:

- To develop computer programs for estimating the Land Surface Temperature (LST) from LANDSAT TM TIR data (open source data).
- To develop computer programs for estimating the ground surface reflectance from LANDSAT TM optical data (open source data).

For calculating the Temperature and Ground surface reflectance certain conversions are involved. Both the conversions are done in two stages as follows:

a) DN values--->Spectral Radiance--->Temperature (as per Chander, et al; 2009)

b) DN values--->Spectral Radiance---->Ground surface reflectance (as per Chander, et al; 2009)

2.1 Need for Converting Digital Numbers to Radiance:

Although Digital numbers have a practical integer value, they do not present the brightnesses in the desired physical units (Watts per square meter per micrometer per steradian). When image brightnesses are expressed as the Digital Numbers, each image becomes a unique and individual entity where there will not be any specific defined relationship to other images or features on the ground. DNs express accurate relative brightnesses within an image but cannot be used to examine brightnesses over time (from one date to another), to compare brightnesses from one instrument to another, to match one scene with another, nor to prepare mosaics of large regions. Further, DNs cannot serve as input for models of physical processes in (for example) agriculture, forestry, or hydrology. Therefore, conversion of DNs to radiances forms an important transformation to prepare remotely sensed imagery for subsequent analyses. Spectral radiance generally means the radiance calculated per unit wavelength. (Campbell and Wayne, 2011).

2.2 Digital Number (DN) to Temperature Conversion:

***Step1**. Conversion of the Digital Number (DN) to Spectral Radiance (L)*

$L = L_{MIN} + (L_{MAX} - L_{MIN}) * DN / 255$

Where

L = Spectral radiance [W/(m^2 sr μm)]

L_{MIN} = 1.238 (Spectral radiance of DN value 1)

L_{MAX} = 15.600 (Spectral radiance of DN value 255)

DN = Digital Number

***Step2.** Conversion of Spectral Radiance to Temperature in Kelvin*

$T_B = K_2/\ln((K_1/L)+1)$

Where

K1 = Calibration Constant 1 (607.76) [W/ (m^2 sr μm)]

K2 = Calibration Constant 2 (1260.56) [K]

T_B = Surface Temperature [Kelvin]

L = Spectral Radiance [W/ (m^2 sr μm)]

Ln = Natural Logarithm

***Step3**. Conversion of Kelvin to Celsius*

$T_B = T_B - 273$

The value of T_B is in degrees Celsius

We need to change L_{MIN} and L_{MAX} for each thermal scene

We can find these values in satellite header file.

2.3 Digital Number (DN) to Ground surface reflectance conversion:

Note: This methodology is generally applied for DN to TOA reflectance conversion. But since the DN values are corrected for the atmospheric effects like scattering, and given as input it the conversion transforms directly to DN to ground surface reflectance. So in this conversion, the QCAL (DN) values are corrected for atmospheric effects and given as input. The DNs of the satellite imagery are corrected for atmospheric effects through radiometric preprocessing of the data under which correction methods are applied. One such method applied for our analysis is Dark Object Subtraction Technique (DOS).

Step 1: *Spectral Radiance Scaling Method (Conversion of DN to Radiance)*

The formula used in this process is as follows:

$L= [(L_{MAX}- L_{MIN})/ (QCAL_{MAX}-QCAL_{MIN})] * (QCAL-QCAL_{MIN}) + L_{MIN}$

Where:

L =cell value as radiance [W/ (m^2 sr μm)]

QCAL = Quantized calibrated pixel value [DN]

L_{MIN} = spectral at-sensor radiance that is scaled to $QCAL_{MIN}$ [W/ (m^2 sr μm)]

L_{MAX} = spectral at-sensor radiance that is scaled to $QCAL_{MAX}$ [W/ (m^2 sr μm)]

$QCAL_{MIN}$ = the minimum quantized calibrated pixel value corresponding to L_{MIN} (DN)

(typically = 1)

$QCAL_{MAX}$ = the maximum quantized calibrated pixel value corresponding to L_{MAX}

(typically = 255)

Step 2: *Conversion of Radiance to Ground surface Reflectance*

$P = [(pi*L*d^2)/(ESUN* \cos \theta_s)]$

Where P =Unit less Planetary Ground surface reflectance

Pi= 3.14159

L= Spectral radiance at sensors aperture (from earlier step) [W/(m^2 sr μm)]

d = Earth-Sun distance in astronomical units

ESUN = mean solar exoatmospheric irradiances [W/(m^2 μm)]

θ_s = solar zenith angle(degrees)

The values of the constants mentioned above can be obtained from the satellite header file.

For the above mentioned methods for converting DN values to Temperature values computer programs have been developed.

2.4 Radiometric Pre-processing:

Radiometric pre-processing involves the correction of brightness values of the image for sensor malfunctions or to compensate atmospheric degradation. Any remote sensing sensor typically records two types of brightness values. One is the brightness derived from the Earth's surface and the other is due to the atmospheric scattering. For example if we consider a pixel having a brightness value of 58 it might be due to the addition of the brightness values due to surface reflectance(48) and part of it due to atmosphere scattering(10). So here if we subtract 10 from the brightness value of the pixel we can get the actual brightness value of the pixel which is derived only due to surface reflectance correcting it for atmospheric effects.

There are many methods available for correcting additive or remove haze and atmospheric effects under radiometric pre-processing of remote sensing data such as Dark Object Subtraction (DOS) technique, COST method proposed by Chavez (1996), other softwares like MODTRAN, ATCOR etc. Out of all these techniques DOS method is one of the simplest and quickest method for correcting atmospheric effects. In our methodology we have used DOS technique for correcting atmospheric effects.

2.5 Dark Object Subtraction Technique:

Dark object subtraction techniques is one of the easiest and compact technique used for correcting atmospheric effects under the radiometric pre-processing of remote sensing data. This strategy can be applied by identifying a very dark object or feature in a scene. Such objects in general can be a water body or shadows cast by clouds or by large topographic features. If we examine the infrared portion of the spectrum, both water bodies and shadows should have brightness at or very near to zero. This is because of the reason that clear water absorbs strongly in the near infrared spectrum and very little infrared energy is scattered to the sensor from those pixels which are effected by shadow. (Campbell and Wyne, 2011)

When such areas are examined, it is observed that the lowest brightness values of these pixels are not zero but have some larger value. These values typically differs from band to band of the obtained satellite imagery. For example for Landsat band 1, the value might be 12; for band 2, the value might be 7; for band 3, 2; and for band 4, 2. (James Campbell and Randolph H. Wyne, 2011) These values, assumed to be contributed from the atmospheric scattering. They are subtracted from all the digital numbers of the particular scene and that band. Thus in every band the lowest value is subjected to be zero, which indicates the dark tone of the object which is the correct tone of the objects under the absence of atmospheric scattering. Such kind of dark objects are regarded as pseudo invariant objects because their spectral properties do not change significantly over time. This strategy forms one of the direct methods involved for correcting digital values due to atmospheric degradation. Hence it is called as Dark object subtraction (DOS) technique. It is also called as Histogram minimum method (HMM) since each value is set to its minimum by correcting it for the atmospheric effect and histogram is plotted.

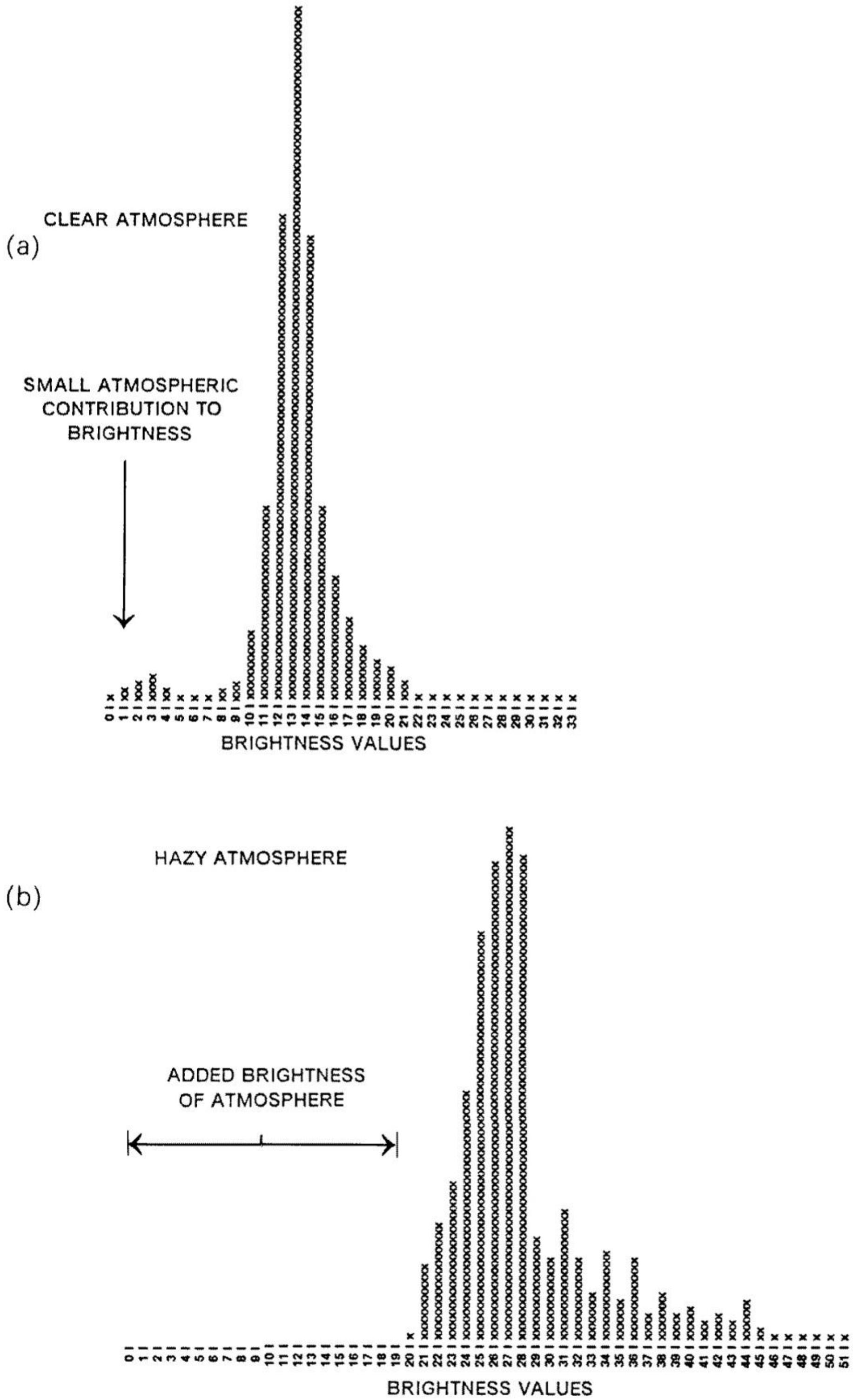


Figure 1: Histogram minimum method for correction of atmospheric effect (Campbell and Wyne, 2011)

The lowest brightness value in a given band is taken to indicate the added brightness of the atmosphere to that band and is then subtracted from all pixels in that band. (a) Histogram for an image acquired under clear atmospheric conditions; the darkest pixel is near zero brightness. (b) Histogram for an image acquired under hazy atmospheric conditions; the darkest pixels are relatively bright, due to the added brightness of the atmosphere.

2.5.1 Advantages of DOS:

This method has the unique advantages of its simplicity, directness, and almost universal acceptance where the information is exploited from the image itself.(Campbell and Wyne, 2011).This technique is in its basic form and can also be applied by making higher modifications such as regression analysis proposed by Chavez (1965). This technique is most simple in its style and its lucidity helps in the application to correct for atmospheric effects.

2.5.2 Disadvantages of DOS:

DOS technique can be considered as an approximation because the atmospheric effects not only change the position of the histogram but can also change the shape of the histogram (i.e all the brightness values are not effected equally there are some variations) (Campbell and Wyne, 2011).Atmospheric effects can cause the darker pixels to become brighter and the brighter ones to darker. So the application of uni-correction to all the pixels present provides only a rough adjustment of the digital values. It is also said that the DOS technique is only capable of resolving additive effects of atmospheric scattering but not multiplicative effects.

DOS was found as the least accurate of the methods evaluated in their study, by Moran et al. (1992), but they recognized that its lucidity and the prospect of pairing it with other strategies might form the basis for a practical technique for operational use. The subtraction of a constant from all pixels in a scene will have a larger proportional impact on the spectra of dark pixels rather than on brighter pixels, so the users should be cautious in using it for scenes in which spectral characteristics of dark features, such as water bodies or perhaps coniferous forests, for example, might form important dimensions of a study. (Campbell and Wyne, 2011).

Despite such concerns, the technique, both in its basic form and in the various modifications such as regression technique proposed by Chavez(1975) and also the extension of these concepts like variance-covariance matrix technique, covariance matrix method [CMM] described by Switzer et al., 1981.) has been found to be satisfactory for a variety of remote sensing applications (e.g., Baugh and Groeneveld,2006; Nield et al., 2007).

2.6 Input Parameters and their dependence:

The input parameters in both the conversions are the digital number values (DN), where the DN values are corrected for atmospheric effects like haze, scattering etc. in the ground surface reflectance estimation but in the estimation of land surface temperature they are directly given as input. This is because in the case of thermal band data acquisition no such atmospheric errors persist. Other than the DN values certain sensor parameters like maximum and minimum radiance values, Quantized calibrated pixel values ($QCAL_{MIN}$,$QCAL_{MAX}$)calibration constants (k_1 and k_2), mean solar exoatmospheric irradiances are used in the estimation of LST and ground surface reflectance whose values vary depending upon the type of sensor used. Other than the sensor parameters certain constants like solar zenith angle, Earth-Sun distance purely depends upon the date of acquisition of the imagery under consideration.

3. STUDY AREA AND DATA USED

3.1 Study Area:

We have selected Mumbai as our study area. Mumbai formerly known as Bombay, is the capital of Maharashtra. It is regarded as the most populous city in India. It is a metropolitan city with a population of around 20.8 million. Mumbai lies on the west coast of India and has a deep natural harbour. Mumbai was named as an Alpha world city in 2009. It is also one of the wealthiest cities in India and has highest Gross Domestic Product (GDP) in South, West or Central Asia. The seven islands that came to constitute Mumbai were home to communities of fishing colonies.

Mumbai is the commercial and entertainment capital of India. It is also one of the world's top 10 centres of commerce. The city houses prominent financial institutions such as the Reserve Bank of India, the Bombay Stock Exchange, the National Stock Exchange of India, the SEBI and the corporate headquarters of numerous Indian companies and multinational corporations. It is also considered as a home to some of India's premier scientific and nuclear institutes like BARC, NPCL, IREL, TIFR, AERB, AECI, and the Department of Atomic Energy.

To check how well our algorithm works we have implemented our methodology for estimating the Land surface temperature and ground surface reflectance values of Mumbai region. The temperature estimation through this methodology helps in obtaining the temperature values of places which are obscure or difficult to find through field data. In the same way the reflectance values helps in establishing the characteristics of the features observed.

3.2 Data Used:

We have used LANDSAT TM data for our study and analysis. LANDSAT data has been chosen because it is open source data and freely available and can be downloaded from the USGS website (www.earthexplorer.usgs.gov). A LANDSAT TM imagery of Mumbai Region with Entity ID: LT51480472009344KHC00 and coordinates (Lat: 19°04'33"N, Lon: 072°52'39"E) has been taken for our analysis. The imagery was acquired on 10th December 2009, with path: 148 and Row: 47. The acquired Landsat TM Imagery consists of seven bands namely (Band 1 to Band 7), out of which Band 6 is Thermal Band and remaining bands are Optical bands.

We have used Thermal Band in our study for the estimation of Land Surface Temperature (LST), and the Optical Bands for the estimation of Ground Surface Reflectance. Since there are 6 optical bands we have chosen the Band 2(Green) for the estimation of Ground Surface Reflectance because of its characteristic of high spectral reflectance. The remaining bands are Band 1(Blue), Band 3 (Red), Band 4(NIR), Band 5(MIR) and Band 7(Far IR).

The satellite header file in general consists of all the information regarding the acquisition and other radiance details of various bands. The header file is generally written as MLT at the end and it is a note pad file in the .txt format. The acquired data header file is denoted by the name "LT51480472009344KHC00_MTL". All the constants involved in the conversion of DN to Temperature and DN to ground surface reflectance can be found out from this header file and can be given as input parameters. All the sun elevation angle and sun azimuth angle are given in degrees which need to be converted to radians while using in the programming concept.

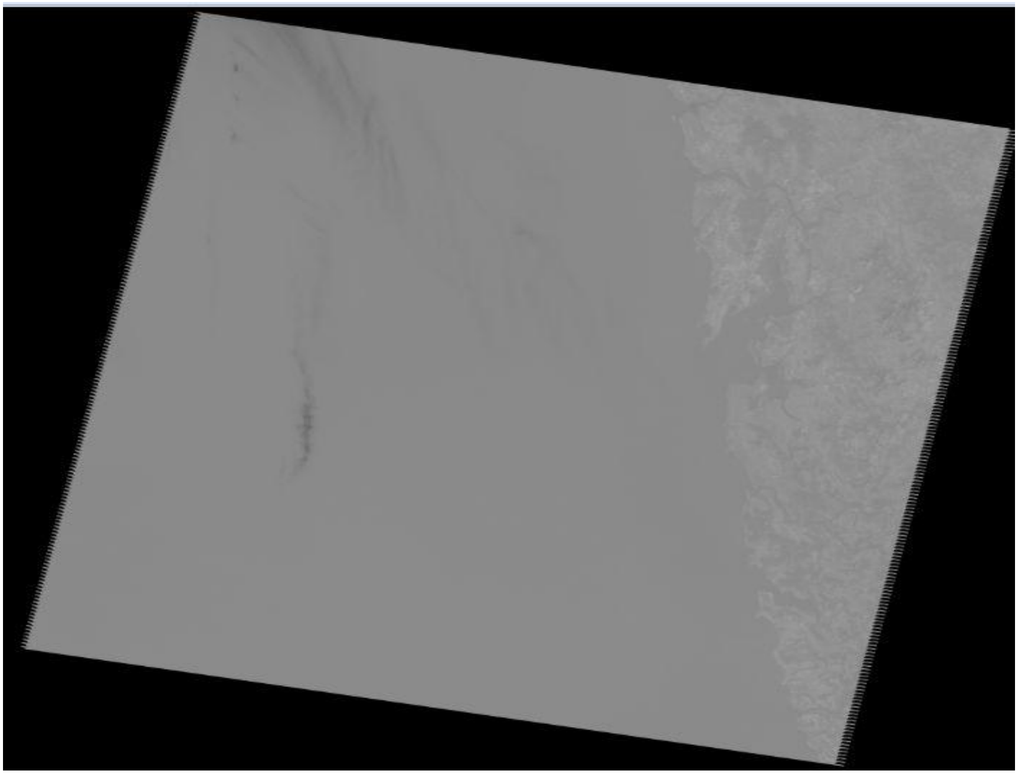


Figure 2 : Band 6 (Thermal Band of Data used)

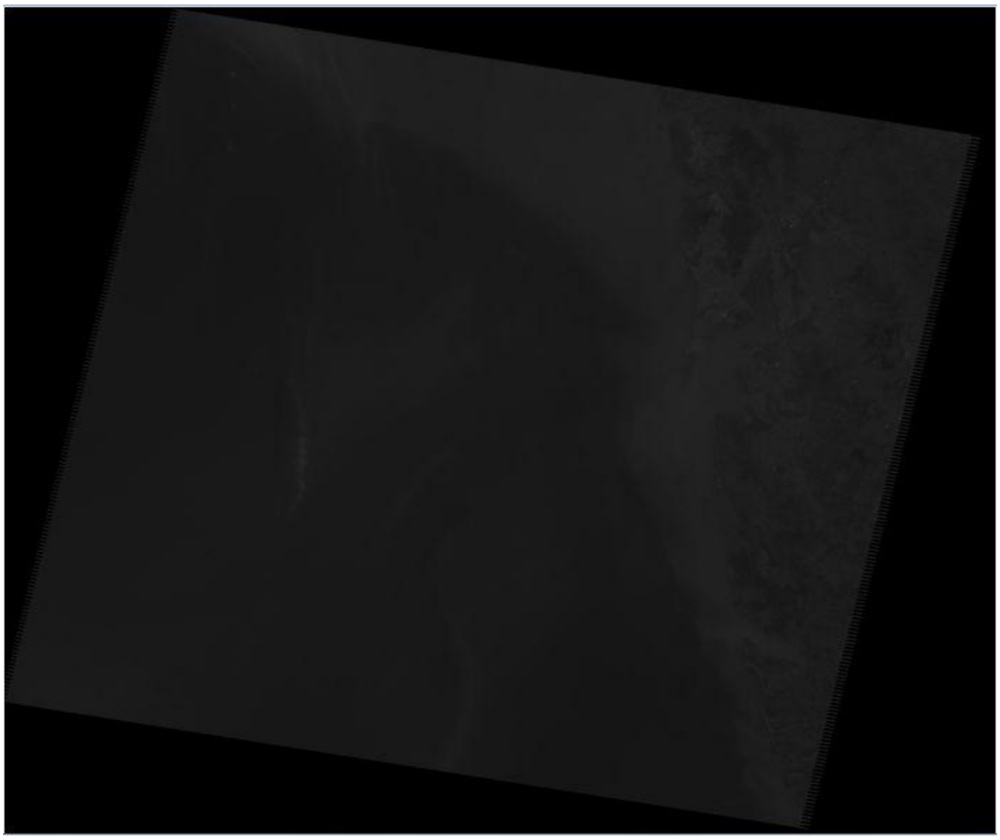


Figure 3 : Band 2 (Green) (Optical band of the data used)

4. ALGORITHM IMPLEMENTATION

4.1 Landsat Scene Details:

The acquired Landsat TM scene (both in band 6 and band 2) comprises 7061 rows and 8001 columns. In the programming part initially we have converted the thermal band (band 6) and optical band green (band 2) into their ASCII formats respectively using the conversion tools present in the Arc Tool Box in the Arc GIS 9.3 software. These ASCII files are respectively stored with (.txt) extension. All the values present in the ASCII file represents the respective DN values of the pixels, also it consists of header file which is common for every ASCII format. The ASCII file must consist of header information containing a set of keywords, followed by cell values in row-major order. The file format is

```
<NCOLS xxx>
<NROWS xxx>
<XLLCENTER xxx | XLLCORNER xxx>
<YLLCENTER xxx | YLLCORNER xxx>
<CELLSIZE xxx>
{NODATA_VALUE xxx}
row 1
row 2
.
.
.
row n
```

(http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=ascii_to_raster_%28conversion%29)

4.2 Step by step procedure for implementing algorithm:

Step 1: Obtain the bands from the downloaded data, check for the thermal band (band 6) and optical bands (band1-5 and 7)

Step 2: For the estimation of Land surface temperature, convert the band 6(.tiff file) which is in the Raster format to the desired ASCII format (.txt) by using the conversion tools option (Raster to ASCII) under Arc Tool Box in Arc GIS 9.3 software as shown in Fig 4.

Step 3: For the estimation of ground surface reflectance check for the green band (band 2) (.tiff file) using their wavelengths from table 1 and give it as input similar to the above mentioned case and convert it into the ASCII format as shown in Fig 5.

Step 4: Now the obtained ASCII format files are given as input in the corresponding java code written separately for estimation of land surface temperature and ground surface reflectance by specifying their path in the code.

Step 5: Run the code and the output files (output 1- temperature values, output 3-Ground surface reflectance values) are stored respectively in a note pad file which will be in the ASCII format.

Step 6: These ASCII format files are in turn converted to their Raster formats by navigating as File→import raster data→ESRI ASCII grid import in GRASS GIS 6.4.3 RC 2, (open source software) where the pixel values of the obtained raster format represent their corresponding temperature and ground surface reflectance values.

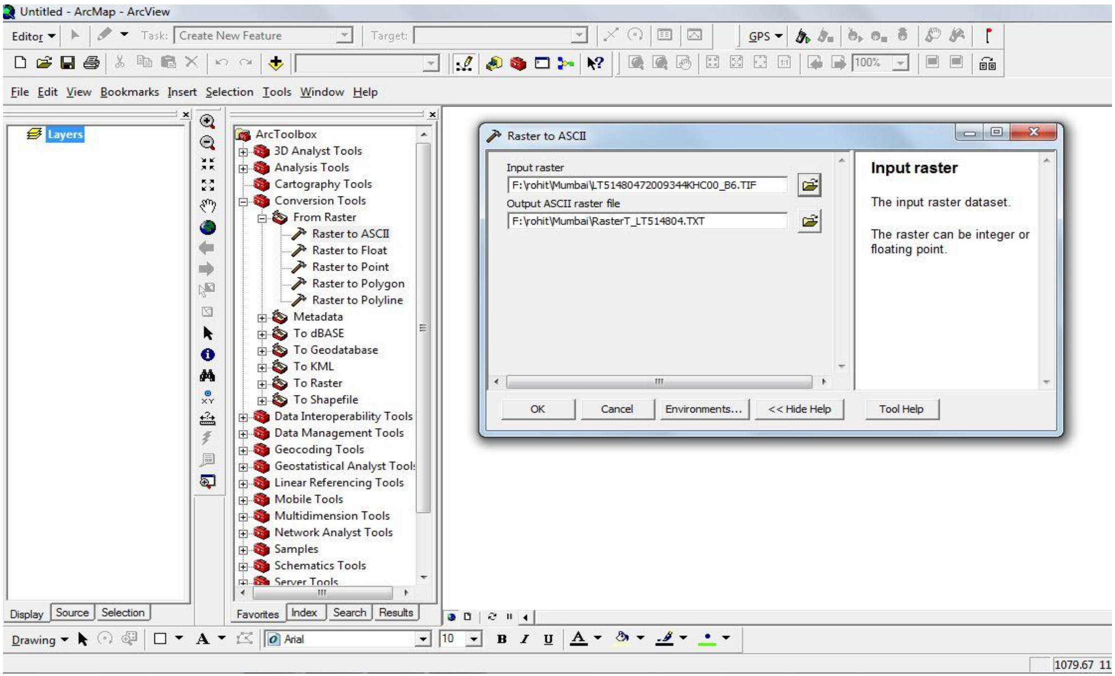


Figure 4: Raster to ASCII conversion of Thermal Band (Band 6)

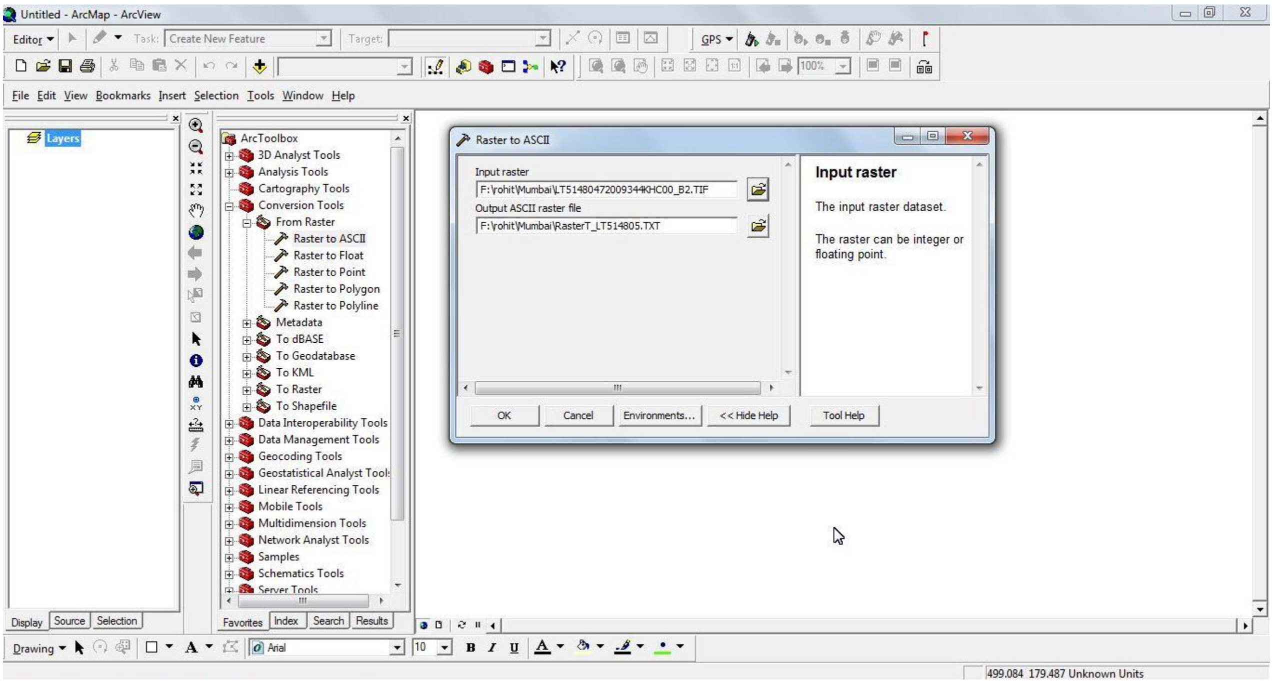


Figure 5: Raster to ASCII conversion of Optical Band Green (Band 2)

4.3 ASCII Input:

The bands (thermal band and optical green band) are converted from raster to ASCII format and the obtained ASCII files are given as input, by giving the path of the file ("F:rohit/Mumbai/rastert_lt514801.txt") for thermal band and ("F:rohit/Mumbai/rastert_lt514802.txt") for optical green band. We have given a properties file which consists of all the constant values involved in the conversion process and imported it into the JAVA code. We have also imported a Java Archive JAR folder (commons-io-2.4).

4.3.1 About Commons IO

Commons IO is a library of utilities to assist with developing IO functionality. There are six main areas included:

- Utility classes - with static methods to perform common tasks
- Input - useful Input Stream and Reader implementations
- Output - useful Output Stream and Writer implementations
- Filters - various implementations of file filters
- Comparators - various implementations of `java.util.Comparator` for files
- File Monitor - a component for monitoring file system events

(Source: http://commons.apache.org/proper/commons-io/)

The ASCII file consisting DN values is given as input by giving the path of the file. The constants have been given in a separate file called Properties File to access use and them for calculation purpose in the formula. When the code is run it reads each and every DN value and converts it to the Radiance and then to Temperature values and displays them in the form of a matrix of the given order. Same is the case of Reflectance, where each and every DN value in the matrix is converted to Radiance, which are again converted to Reflectance values.

The code for both conversions has been written in JAVA language and complied and Run in Net Beans 7.1.1, which is a premium software package of Java for executing and compiling the code.

4.4 Softwares Used:

4.4.1 NetBeans:

NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others. The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by third party developers. (http://en.wikipedia.org/wiki/NetBeans)

We have created two projects in NetBeans namely dntotemp and dntoreflectace which contains source files (.java) namely DnToTemp.java and Dntoreflectance.java

4.4.2 GRASS GIS:

Grass GIS commonly referred to as Grass (Geographic Resource Analysis Support System) is one of the open source geographical information systems softwares which are extensively used for geospatial data management and analysis, remote sensing image analysis, image processing, graphics/map production, spatial modelling, and visualization.(http://grass.osgeo.org/). Our main motto behind using Grass GIS is because it is an open source software freely available and images can be viewed easily. Whereas other GIS softwares like Arc Map are licensed and not freely available. Moreover, Grass GIS software is compact and intuitive. Grass GIS is an official project of the Open Source Geospatial Foundation. Grass GIS is currently used for academic needs, departmental and organisational activities generally by the governmental agencies. Some of the other open source remote sensing and GIS softwares that can be used for digital image analysis are QGIS, ER Mapper, Hyper spec etc.

4.4.3Arc GIS:

ArcGIS is a Software suite (collection of computer programs) consisting of a group of geographic information system (GIS) software products produced by ESRI. It is a system that was designed to work with maps and the geographic information. It is mainly used for

- creating and using maps
- compiling geographic data
- analysing mapped information
- sharing and discovering geographic information
- using maps and geographic information in a range of applications and
- Managing geographic information in a database.

Figure 6 : Image showing the different products of Arc GIS.
(http://www.ds2corp.com/Services/GeospatialServices.aspx)

4.4.3.1 Arc GIS Products:

Arc GIS includes products in each of the following categories:

a) Desktop GIS:

Arc GIS Desktop is a collection of GIS software products for standard desktop computers. These products can be used to create, import, edit, query, map, analyse and publish geographic information. There are four products in the ArcGIS Desktop collection:
Arc Info, Arc Editor, ArcView, Arc Reader, ArcGIS Engine, ArcGIS Explorer

b) Server GIS:

Arc GIS Server is a Server-based GIS technology used for centrally hosted GIS computing. GIS software can be centralized in application servers. Enterprise GIS users connect to central GIS servers using Desktop GIS, Web browsers, mobile computing devices, and digital appliances.

c) Mobile GIS:

- ArcGIS Mobile
- Arc Pad

d) Online GIS:

ArcGIS Online

e) Esri Data:

- Community Data
- Street Map, Esri Data & Maps

5.1 Process involved in Digital Number to Temperature Conversion:

The process involved in the Digital Number to Temperature conversion consists of two steps like loading the constants as properties file in the java code as shown in Fig 7 and running the java code in Net Beans to obtain the desired output.

5.1.1 Constants used for Digital Number to Temperature Conversion:

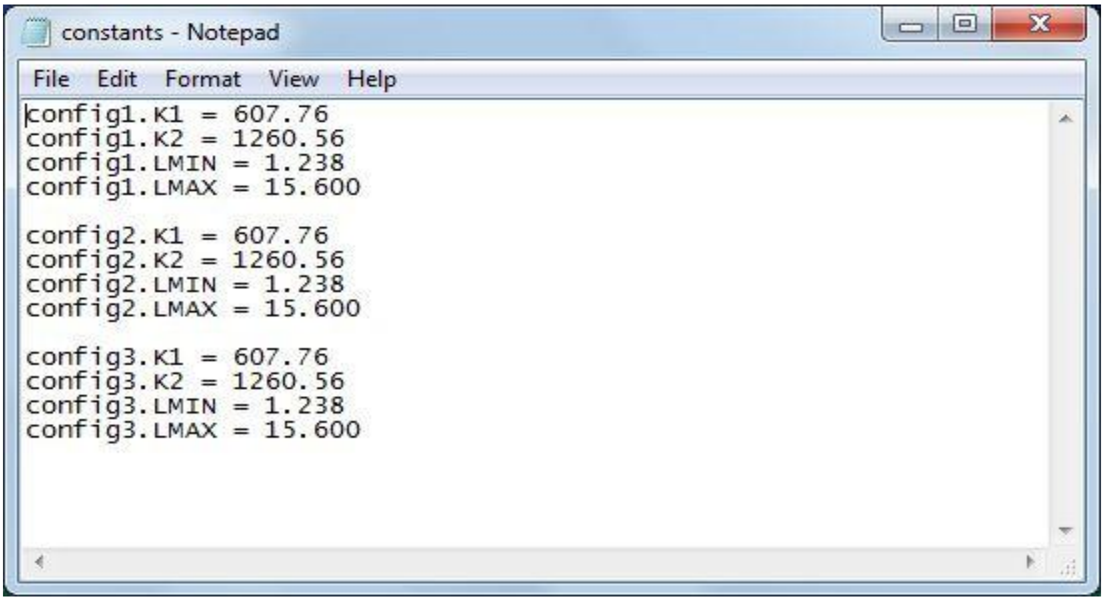


Figure 7: Properties file showing the constant values used for DN to temperature conversion

The above shown file consists of all the constant values. Here we have written the set of constants thrice by giving them as config1, config2, config3 because if we want the computation for different set of values we can directly edit config1 in the program to config2 or config3 and run the file. You can even give different values of K_1,K_2,L_{MIN},L_{MAX} in each set (config1,config2,config3) according to the type of satellite sensor used and input whichever set you want(config1 or config2 or config 3) in the program and run the file.

While writing the code care has been taken that the temperature values are computed only for those pixels having DN values other than zero. This is because DN value of Zero represents those pixels that are out of area of interest.

5.1.2 Java code for Digital Number to Temperature Conversion:

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.util.List;
import java.util.Properties;
import java.util.StringTokenizer;
```

```
import org.apache.commons.io.FileUtils;
public class DnToTemp {
    //Calibration Constant 1
    public static float K1;
    //Calibration Constant 2
    public float K2;
    //(Spectral radiance of DN value 1)
    public float LMIN;
    //(Spectral radiance of DN value 255)
    public float LMAX;
      public void loadProperties()
    {
        InputStream is = this.getClass().getClassLoader().getResourceAsStream("constants.properties");
        Properties p = new Properties();
        try {
            p.load(is);
            K1 = Float.parseFloat(p.getProperty("config1.K1"));
            K2 = Float.parseFloat(p.getProperty("config1.K2"));
            LMIN = Float.parseFloat(p.getProperty("config1.LMIN"));
            LMAX = Float.parseFloat(p.getProperty("config1.LMAX"));
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }


    }
    public void readImage() throws IOException
    {
       List<String>lines=FileUtils.readLines(new File("F:rohit/Mumbai/rastert_lt514801.txt"));
        System.out.println("Lines read");
        FileWriter fw = new FileWriter("C:/Users/rvg296/Desktop/output1.txt");
        //fw.flush();
        PrintWriter out = new PrintWriter(fw);
        for(int i=0; i < lines.size(); i++)
        {
            if(!lines.get(i).contains("//") && !lines.get(i).contains("/*"))
```

```
        {
          //System.out.println("Line 1: "+lines.get(i));
          StringTokenizer s = new StringTokenizer(lines.get(i));
          StringBuffer sb = new StringBuffer();
          while(s.hasMoreTokens())
          {
            int DN = Integer.parseInt(s.nextElement().toString());
            //System.out.print(DN+" ");
            if(DN!=0)
            {
              float L = convertNumberToSpectralRadiance(DN);
              float TB = convertSpectralRadianceToTemperature(L);
              sb.append(TB+" ");
              System.out.print(TB+" ");
            }
            else
            {
              sb.append(0+" ");
              System.out.print(0+" ");
            }
          }
          //output to the file a line
          out.println(sb);
          System.out.println("\n");
        }
        //close the file (VERY IMPORTANT!)
      }
      out.close();

    }
    public float convertNumberToSpectralRadiance(int DN)
    {
      //Spectral radiance
      float L = LMIN + (LMAX - LMIN) * DN / 255;
      return L;
```

```
    }
    public float convertSpectralRadianceToTemperature(float L)
    {
        //Surface Temperature
        float TB = (float) (K2/Math.log(K1/L+1));
        //Conversion of Kelvin to Celsius
        TB = TB - 273;
        return TB;
    }
    public static void main(String[] args) {
        try
        {
            DnToTemp s = new DnToTemp();
            s.loadProperties();
            s.readImage();
            System.out.println("Please check the text file Output1 on Desktop");
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

When we right click one the program and select run file or alternatively press Shift+F6, the program runs and starts printing the output on the console screen, where all the temperature values are separately stored in a note pad file (.txt extension) on the desktop.

5.1.3 Result:

The output window shown in Fig 8, Fig 9 is an ASCII file generated which consists of all the values of temperature in degrees Celsius corresponding to their DN values that have been given as input.

5.1.4 Output file on desktop (Initial part)

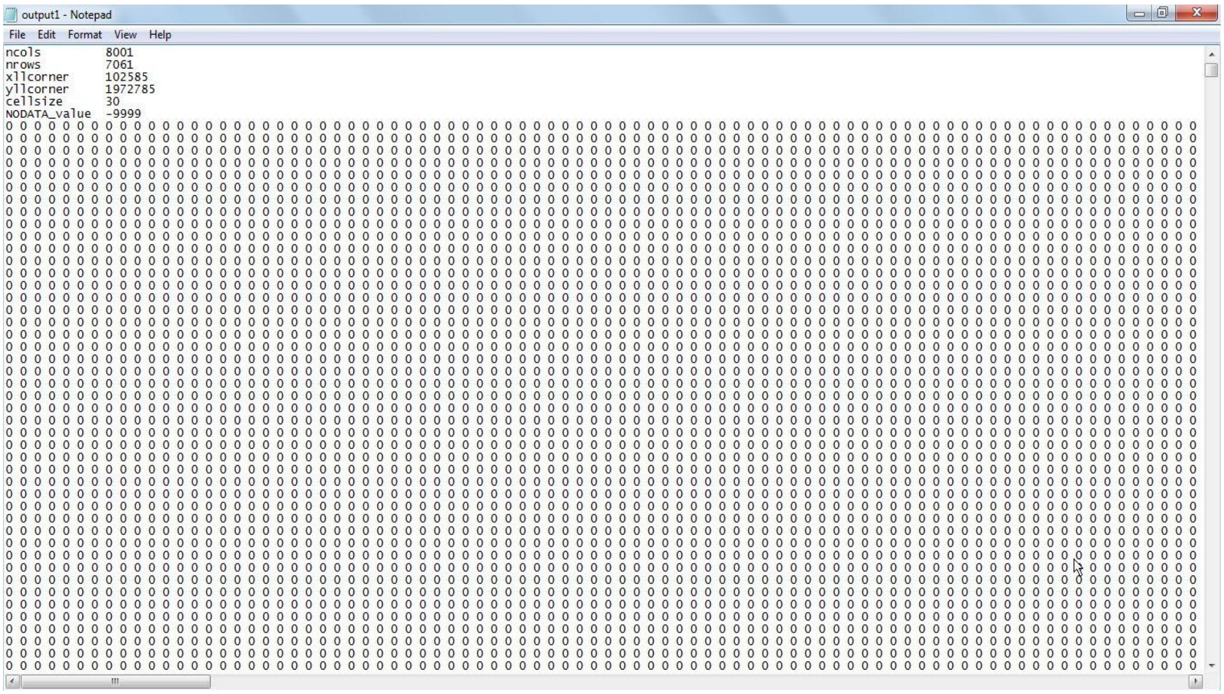


Figure 8: Output ASCII file showing the header format and unchanged zero temperature values of corresponding DN.

5.1.5 Output File on Desktop (other part)

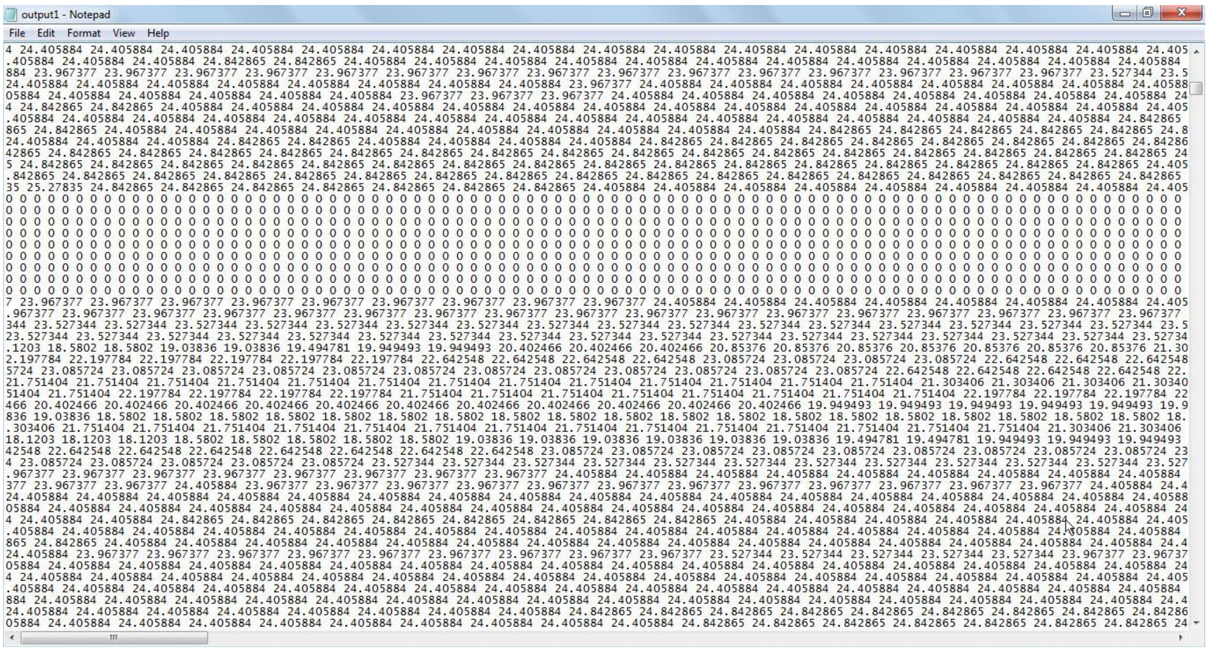


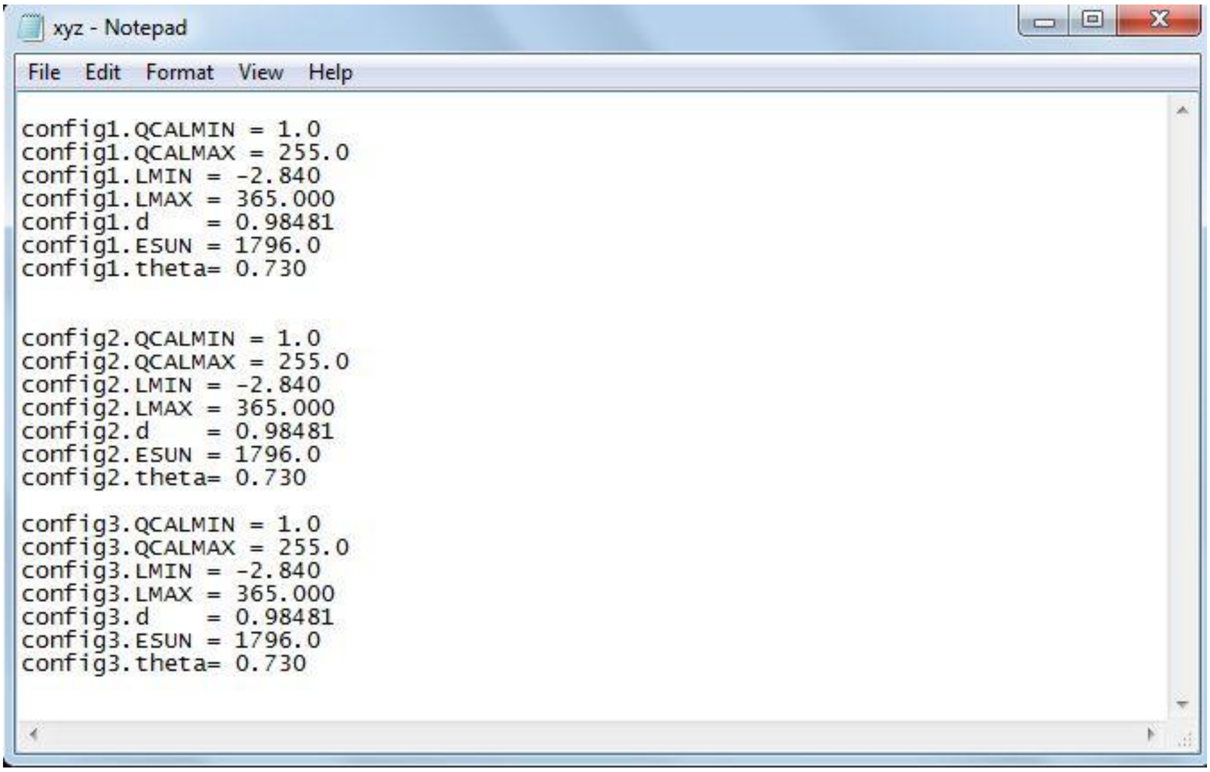
Figure 9: Output ASCII file showing the temperature values in ˚C of corresponding DN values of pixels other than zero which is generated after converting DN to Temperature.

Note: The output file generated consists of as many as 8001 columns and 7061 rows. Since the file is large only a sample or part of the file has been shown as output here. . Here the values other than zero represents the temperature values of the corresponding DN values of pixels.

5.2 Process involved in Digital Number to Reflectance Conversion

The process involved in the Digital Number to ground surface reflectance conversion consists of two steps like loading the constants as properties file in the java code as shown in Fig 10 and running the java code in Net Beans to obtain the desired output.

5.2.1 Constants used for Digital Number to Ground Reflectance conversion




```
config1.QCALMIN = 1.0
config1.QCALMAX = 255.0
config1.LMIN = -2.840
config1.LMAX = 365.000
config1.d    = 0.98481
config1.ESUN = 1796.0
config1.theta= 0.730

config2.QCALMIN = 1.0
config2.QCALMAX = 255.0
config2.LMIN = -2.840
config2.LMAX = 365.000
config2.d    = 0.98481
config2.ESUN = 1796.0
config2.theta= 0.730

config3.QCALMIN = 1.0
config3.QCALMAX = 255.0
config3.LMIN = -2.840
config3.LMAX = 365.000
config3.d    = 0.98481
config3.ESUN = 1796.0
config3.theta= 0.730
```

Figure 10: Properties file showing the constant values used for DN to reflectance conversion

In this conversion the JAVA code is written in such a way that it checks the input ASCII file and finds the least DN value from it other than zero and subtracts it (DOS concept) from every DN except zero. So it uses the corrected DN values for the calculation of ground surface reflectance.

The above shown file consists of all the constant values. Here we have written the set of constants thrice by giving them as config1, config2, config3 because if we want the computation for different set of values we can directly edit config1 in the program to config2 or config3 and run the file. You can even give different values of $QCAL_{MIN}$, $QCAL_{MAX}$, L_{MIN}, L_{MAX}, d, ESUN, theta in each set (config1,config2,config3) and input whichever set you want(config1 or config2 or config 3) in the program and run the file.

Note: Here the value of theta should be given in radians. The degrees should be converted to radians and should be given as input in the properties file shown above.

5.2.2 Java Code for Digital Number to Ground surface reflectance conversion

Note: The code for DN to Ground reflectance conversion has been written in such a way that the minimum (except zero) from the DN values is selected and subtracted from all other DN values (except zero) and the resultant DN values are given as input for calculation of ground surface reflectance which forms the basis of DOS technique mentioned earlier in 2.4

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import java.util.List;
import java.util.Properties;
import java.util.StringTokenizer;
import org.apache.commons.io.FileUtils;
public class Dntoreflectance {
    //Minimum quantized pixel calibrated pixel value
    public float QCALMIN;
    //Maximum quantized pixel calibrated pixel value
    public float QCALMAX;
    //(Spectral radiance of DN value 1)
    public float LMIN;
    //(Spectral radiance of DN value 255)
    public float LMAX;
    //Earth-sun distance in astronomical units
        public float d;
        //Mean solar exoatmospheric irradiance
        public float ESUN;
        //Solar Zenith angle
        public float theta;
    public void loadProperties()
    {
        InputStream is = this.getClass().getClassLoader().getResourceAsStream("xyz.properties");
        Properties p = new Properties();
        try {
            p.load(is);
            QCALMIN = Float.parseFloat(p.getProperty("config1.QCALMIN"));
            QCALMAX = Float.parseFloat(p.getProperty("config1.QCALMAX"));
            LMIN = Float.parseFloat(p.getProperty("config1.LMIN"));
            LMAX = Float.parseFloat(p.getProperty("config1.LMAX"));
                         d   = Float.parseFloat(p.getProperty("config1.d"));
                        ESUN = Float.parseFloat(p.getProperty("config1.ESUN"));
```

```
                theta= Float.parseFloat(p.getProperty("config1.theta"));
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

}
public float getMinimum() throws IOException
{
    List<String> lines = FileUtils.readLines(new File("F:/rohit/Mumbai/rastert_lt514802.txt"));
    float min = 0;
    for(int i=0; i < lines.size(); i++)
    {
        if(!lines.get(i).contains("//") && !lines.get(i).contains("/*"))
        {
            StringTokenizer s = new StringTokenizer(lines.get(i));
            while(s.hasMoreTokens())
            {
                int x = Integer.parseInt(s.nextElement().toString());
                if(min == 0)
                {
                    min = x;
                }
                else
                {
                    if(x!=0 && x<min)
                    {
                        min = x;
                    }
                }
            }
        }
    }
    System.out.println("Min: "+min);
    return min;
}
public void readImage() throws IOException
```

```
{
    List<String> lines = FileUtils.readLines(new File("F:/rohit/Mumbai/rastert_lt514802.txt"));
    FileWriter fw = new FileWriter("C:/Users/rvg296/Desktop/output3.txt");
    fw.flush();
    fw.flush();
    PrintWriter out = new PrintWriter(fw);
    float minimum = getMinimum();
    for(int i=0; i < lines.size(); i++)
    {
        //System.out.println("Line 1: "+lines.get(i));
        if(!lines.get(i).contains("//") && !lines.get(i).contains("/*"))
        {
            StringTokenizer s = new StringTokenizer(lines.get(i));
            StringBuffer sb = new StringBuffer();
            while(s.hasMoreTokens())
            {
                int QCAL = Integer.parseInt(s.nextElement().toString());
                //System.out.print(QCAL+" ");
                if(QCAL!=0)
                {
                    QCAL = (int) (QCAL - minimum);
                    float L = convertNumberToSpectralRadiance(QCAL);
                    float px = convertSpectralRadianceToReflectance(L);
                    sb.append(px+" ");
                    System.out.print(px+" ");
                }
                else
                {
                    sb.append(0+" ");
                    System.out.print(QCAL+" ");
                }
            }
            out.println(sb);
            System.out.println("\n");
        }
    }
      out.close();
```

```
    }
    public float convertNumberToSpectralRadiance(float QCAL)
    {
        //Spectral radiance
        float L = ((LMAX-LMIN)/(QCALMAX-QCALMIN))*(QCAL-QCALMIN)+LMIN;
        return L;
    }
    public float convertSpectralRadianceToReflectance(float L)
    {
        //Reflectance
        float px = (float) ((float) 3.14159*L*Math.pow(d,2)/ESUN*Math.cos(theta));
        return px;
        }
    public static void main(String[] args) {
        try
        {
            Dntoreflectance s = new Dntoreflectance();
            s.loadProperties();
            s.readImage();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

When we right click one the program and select run file or alternatively press Shift+F6, the program runs and starts printing the output on the console screen, where all the reflectance values are separately stored in a note pad file (.txt extension) on the desktop.

5.2.3 Result:

The output window shown in Fig 11, 12 is an ASCII file which consists of all the reflectance values corresponding to their DN values of the pixels that have been given as input.

5.2.4 Output file on desktop (initial part):

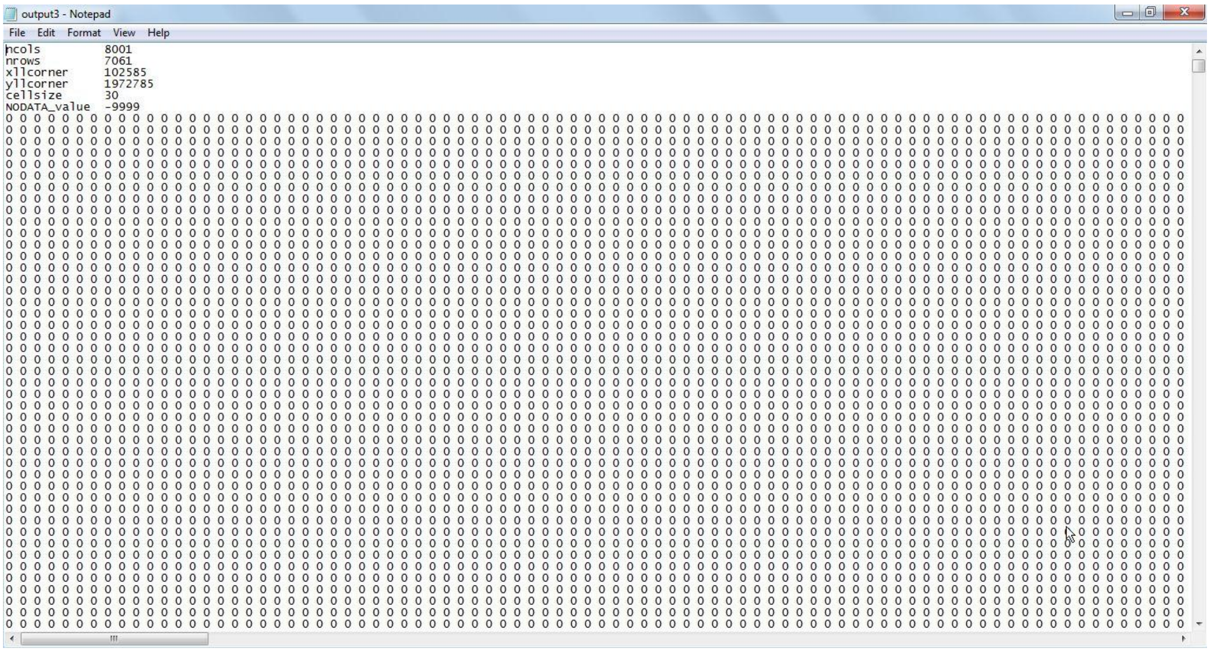


Figure 11: Output ASCII file showing the header format and unchanged zero reflectance values of corresponding DN

5.2.5 Output File on Desktop (other part):

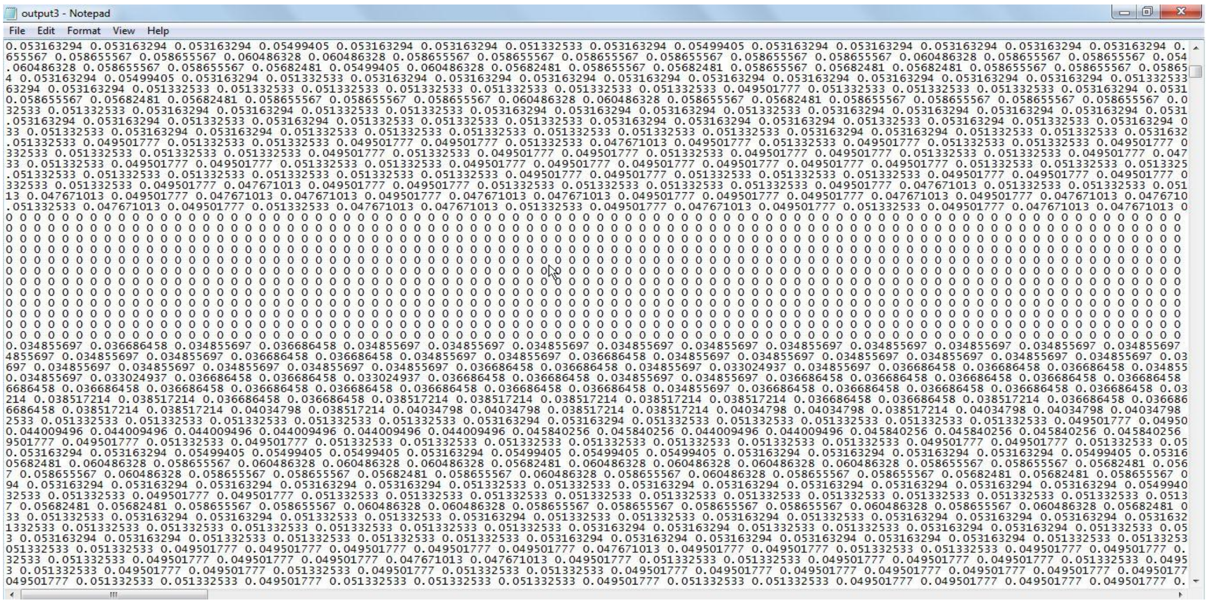


Figure 12: Output ASCII file showing reflectance values of corresponding DN values of pixels other than zero which is generated after converting DN to reflectance.

Note: The output file generated consists of as many as 8001 columns and 7061 rows. Since the file is large only a sample or part of the file has been shown as output here. Here the values other than zero represents the reflectance values of the corresponding DN values of pixels.

The temperature and reflectance values are obtained by running their respective java codes in NetBeans and these are stored in a note pad file in ASCII file format. These ASCII files are converted to their respective raster formats using conversion tools present in open source remote sensing software like GRASS GIS, Q GIS etc.

If we open the raster files of the temperature and reflectance in the remote sensing image viewers and point out or click on the pixels we can view their corresponding values.

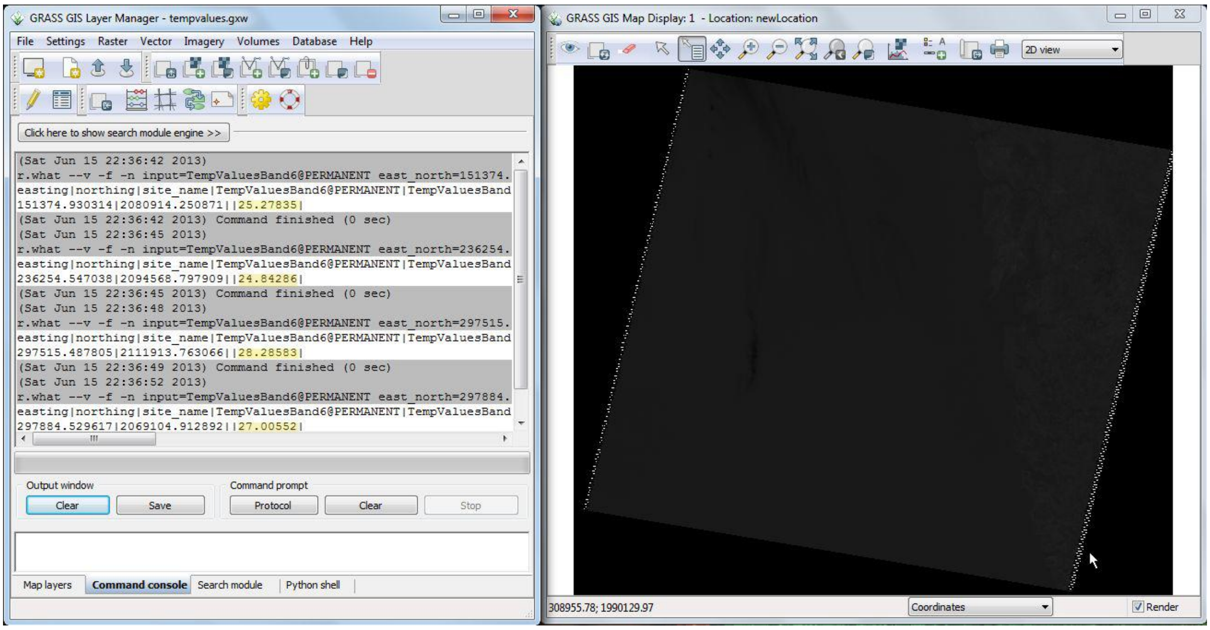


Figure 13: GRASS GIS Layer manager output window showing temperature values (highlighted) in degrees centigrade. Temperature values of the pixels in the left side window are shown whenever clicked upon a certain pixel on the right side thermal band (band 6) by using in GRASS Map Display.

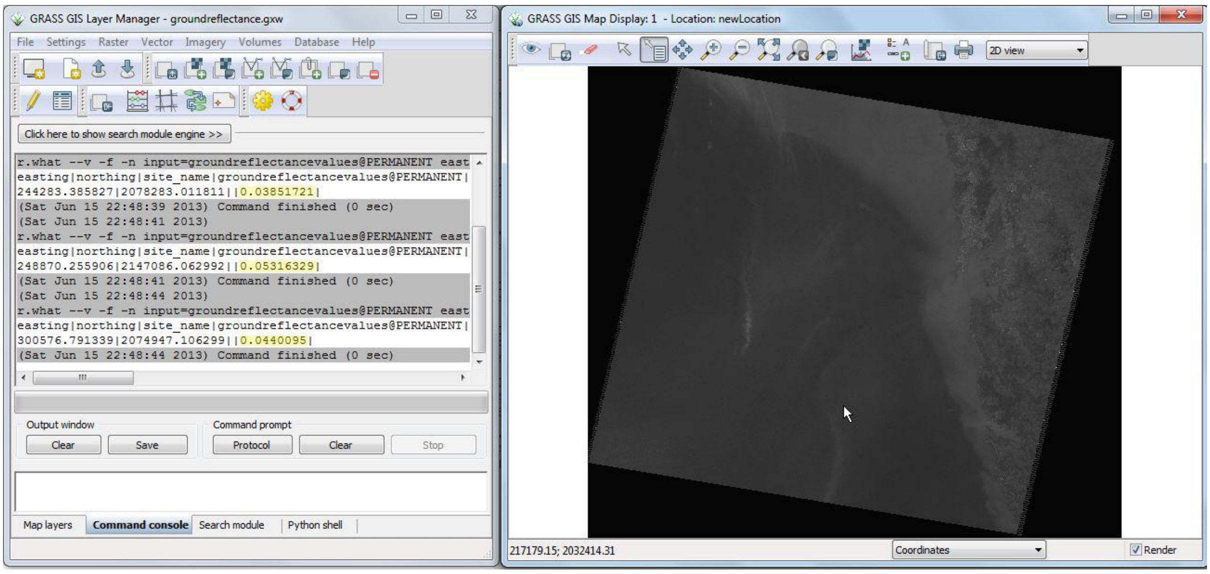


Figure 14: GRASS GIS Layer Manager output window showing ground surface reflectance values(Highlighted).Ground surface reflectance values of the pixels in the left side window are shown whenever clicked upon a certain pixel on the right side optical band (band 2-green) by using in GRASS Map Display.

As stated earlier temperature and reflectance are some of the chief parameters that need to be estimated for various analysis of the areas and also to find out the characteristics of the object or target under consideration. Since manual field measurement makes the task tedious and time consuming, our methodology adopted serves the purpose and makes the estimation of temperature and reflectance values with a greater ease. The temperature values of different features like water bodies, land surface can be estimated. Also the reflectance values estimated helps in finding out of the characteristics of the objects and target under consideration.

Thus the Temperature and Ground surface Reflectance values can be estimated by the combined use of satellite imagery and the computer programming methodology that has been developed, from the program that has been written in JAVA and compiled in NetBeans 7.1.1 using the DN Values and constants. Any ASCII file obtained after converting from Raster consisting of the DN values can be given as input and can be converted easily to the Temperature and Ground surface Reflectance values. The constants that have been given as properties file varies depending on the type of sensor used.

Here for our analysis purpose we have taken Thematic Mapper as our sensor. We can even take other sensors like ETM+ of Landsat or LISS-III of IRS P6 and give their respective constant values (which can be found in the satellite header file) in the properties file, and can estimate the respective temperature and reflectance values.

8. REFERENCES

James Campbell and Randolph H. Wyne (2011), *Introduction to Remote Sensing, Fifth Edition*, THE GUILFORD PRESS, London, New York.

E.Balaguruswamy (2007) *Programming with Java – A Primer,3/e* ,Tata McGraw-Hill,7 West Patel Nagar, New Delhi 110 008.

Herbert Schildt (2007) *Java™: The Complete Reference, Seventh Edition,* Tata McGraw-Hill, United States.

Chander G., Markham B.L, Helder D.L (2009). Summary of current radiometric calibration coefficients for Landsat MSS, TM, ETM+ and EO-1 ALI sensors, Remote Sensing of Environment,113, 897-900.

US Geological Survey, www.usgs.gov

Earth Explorer, www.earthexplorer.usgs.gov

Landsat, http://landsat.gsfc.nasa.gov/

NetBeans, http://en.wikipedia.org/wiki/NetBeans

Commons IO, http://commons.apache.org/proper/commons-io/

Remote Sensing of Environment, www.elsevier.com/locate/rse

Net Beans, https://netbeans.org/

DN to reflectance conversion, http://www.yale.edu/ceo

Landsat 7 Science data Users Hand Book,
http://landsathandbook.gsfc.nasa.gov/data_prod/prog_sect11_3.html

Landsat TM Bands, http://landsat.gsfc.nasa.gov/about/tm.html

Grass GIS, http://grass.osgeo.org/

Arc GIS Desktop 9.3 Help,
http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=ascii_to_raster_%28conversion%29

Professor Dr Yuji Murayama, and Ko Ko Lwin, *Estimation of Landsat TM Surface Temperature Using ERDAS Imagine Spatial Modeler*, Division of Spatial Information Science
Graduate School of Life and Environmental Sciences, University of Tsukuba.