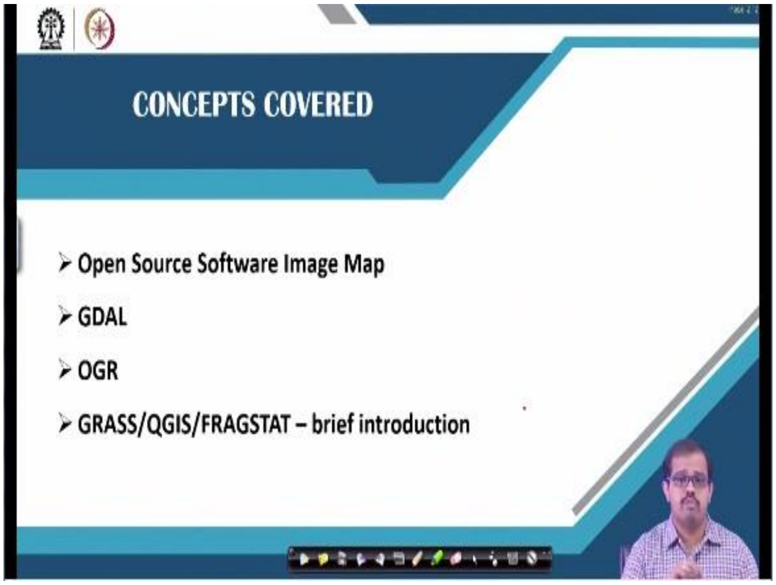
Geographic Information Systems
Dr. Bharath Haridas Aithal
Ranbir and Chitra Gupta School of Infrastructure Design and Management
Indian Institute of Technology – Kharagpur

Lecture – 58
Open Source GIS Software: Introduction

Hello, Namaste. Welcome back to the course on Geographic Information System. In today's class; I said in the previous class we will look at more of open source GIS software. So we looked at the licensing system now we are looking at the software. So some of; one or two of these software’s you may have learnt in the previous two weeks. I would not go into details of those software’s.

But I would give you one very good software which can be used in urban; as an urban analytical tool, extremely good in terms of analyzing urban; I have used it for urban, I have used it for various other measurements in terms of urban data. So it can be used for ecological purposes; it can be used for environmental purposes; it can be used for hydrological purpose. So the way you think you can modify that particular software for finding out different metrics. So I will speak about some of these software’s. Maybe you can use it for your own research code.

(Refer Slide Time: 01:25)



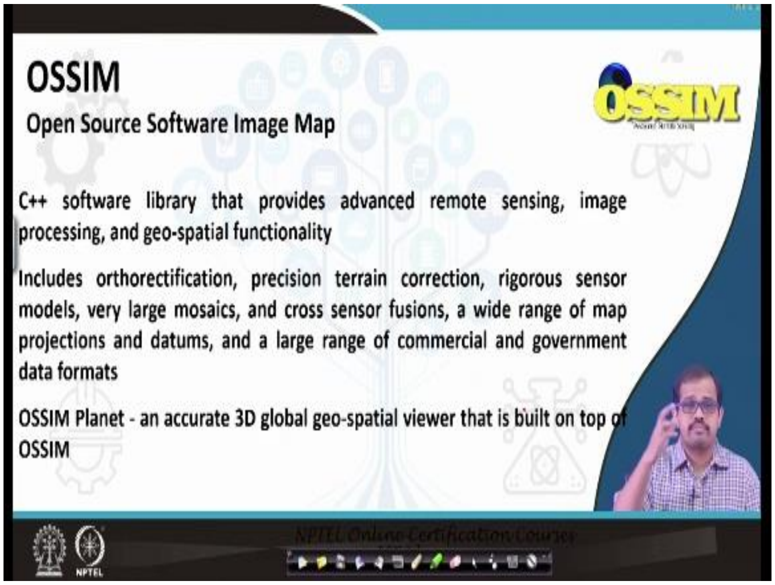
CONCEPTS COVERED

- Open Source Software Image Map
- GDAL
- OGR
- GRASS/QGIS/FRAGSTAT – brief introduction

So let us look at the content. So the first thing I would speak about an Open Source Software Image Map. So this is a kind of software which can be used and modified by any user so that the image map or the analysis; any kind of analysis can be done in a much easier way. Then you have GDAL. GDAL is one of the extremely powerful software which is powering I mean extremely powerful library which is actually powering most of the software's GIS software's that are available today.

And OGR is other software which is complementing GDAL in terms of handling the vector data. So this both are the backbone of most of the GIS software’s available today. And I would also speak about GRASS, QGIS and FRAGSTAT. I will give you a brief introduction and a small example of how we look at the software as it is.

(Refer Slide Time: 02:19)



OSSIM

Open Source Software Image Map

C++ software library that provides advanced remote sensing, image processing, and geo-spatial functionality

Includes orthorectification, precision terrain correction, rigorous sensor models, very large mosaics, and cross sensor fusions, a wide range of map projections and datums, and a large range of commercial and government data formats

OSSIM Planet - an accurate 3D global geo-spatial viewer that is built on top of OSSIM

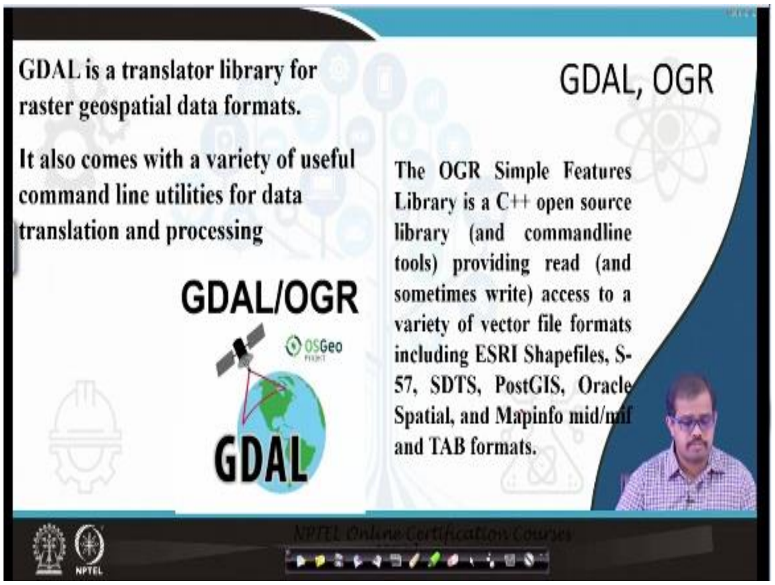
Okay. The first software when we look at as I said is an open source software image map. So you may not have heard about this but it is extremely good software library; it is written in C++, okay. That provides an advanced remote sensing image processing and geo-spatial functionaries. So why I am trying to tell about OSSIM is that, for example we spoke about what you mean by remote sensing, very briefly.

If someone wants to combine the GIS remote sensing and image processing together to produce some good results, so OSSIM can be in one of those software libraries which can be extremely helpful for many of you, so it includes orthorectification, precision terrain, correction, rigorous sensor models, very large mosaics. So if you have trace, today where computer is easy to; I mean it can handle anything.

So that is how you can look at any of those applications that can come in. For example, a cross sensor fusions, wide range of map projections, datums and a large range of commercial and government data formats. So all are these can be handled using OSSIM. So if someone is trying to develop something or someone is not able to maybe look at the software functionalities, if you; when you are looking at data management or developing an image processing tool or looking at image processing as an effective tool.

So you can look at OSSIM so that your analysis becomes much faster and more productive. So when I look at OSSIM as a planet it is an accurate 3D global and geo-spatial viewer that is built on top of OSISM. So please look at it OSSIM planet is very good in terms of visualization. It is a 3D global visualization please look at it. If you look at the way, it has evolved today OSSIM is one of those libraries that has to be looked at when you are trying to do research using image processing and its applications.

(Refer Slide Time: 04:36)



GDAL, OGR

GDAL is a translator library for raster geospatial data formats.

It also comes with a variety of useful command line utilities for data translation and processing

The OGR Simple Features Library is a C++ open source library (and commandline tools) providing read (and sometimes write) access to a variety of vector file formats including ESRI Shapefiles, S-57, SDTS, PostGIS, Oracle Spatial, and Mapinfo mid/mif and TAB formats.

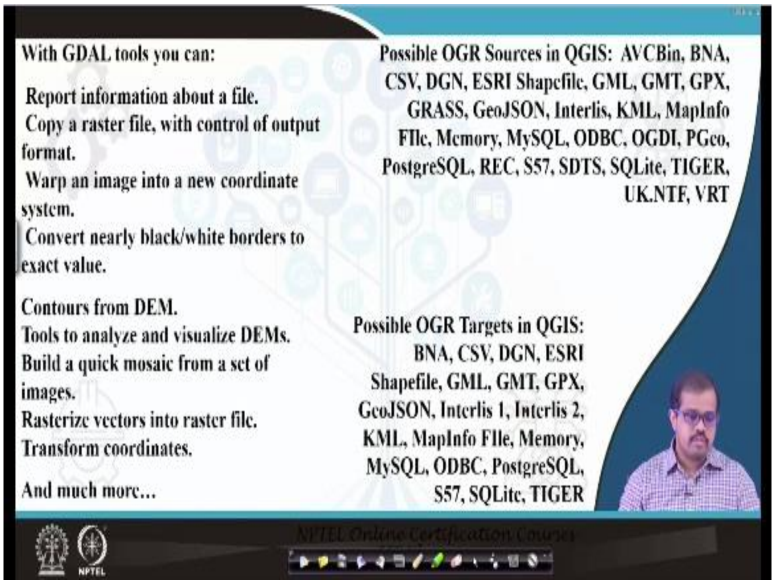
So the next thing is GDAL. So GDAL is basically a translator library. Okay. I could say it as a translator library for raster geo-spatial data formats. So when I say raster geo-spatial data formats it handles effectively the raster model, okay. So it also comes with a variety of useful command line utilities for data I mean translation processing. And when you look at this GDAL, take it any software whether you are using ERDAS whether you using any of those raster processing very good capable raster processing software.

The raster library is formed through the GDAL translator itself, okay. So the entire handling of the raster data is through GDAL, okay. So when you look at OGR, OGR complements GDAL with a vector as a vector translator, okay. OGR Simple Features Library is a C++ open source library written in C++ and providing read access to variety of vector files formats including whether it Shapefiles.

Most of you use a shapefile, SDTS, PostGIS, Oracle Spatial, Mapinfo mid/mif or a TAB format. So all of these can be easily handled by OGR, so now take any vector software that has been that is in the market today whether it is proprietor open source. The basic library behind all of this software is OGT. Okay. Basic library behind raster software is nothing but GDAL. So whether you are trying to buy a proprietary license basically the thing that at the brain behind us it is GDL and OGR, okay.

But only thing is that in a proprietary software you; it is much easier for you to access all the tools easily and most importantly it looks maybe more interesting in terms of when you look at it as a visual enterprise. But when you are looking at open source software it may be a bit difficult that you have to get accustomed for a few days. But once you are accustomed open source software is also as beautiful as your proprietary software’s.

(Refer Slide Time: 07:01)



With GDAL tools you can:

Report information about a file.
Copy a raster file, with control of output format.
Warp an image into a new coordinate system.
Convert nearly black/white borders to exact value.

Contours from DEM.
Tools to analyze and visualize DEMs.
Build a quick mosaic from a set of images.
Rasterize vectors into raster file.
Transform coordinates.

And much more...

Possible OGR Sources in QGIS: AVCBin, BNA, CSV, DGN, ESRI Shapefile, GML, GMT, GPX, GRASS, GeoJSON, Interlis, KML, MapInfo FIle, Memory, MySQL, ODBC, OGDI, PGeo, PostgreSQL, REC, S57, SDTS, SQLite, TIGER, UK.NTF, VRT

Possible OGR Targets in QGIS: BNA, CSV, DGN, ESRI Shapefile, GML, GMT, GPX, GeoJSON, Interlis 1, Interlis 2, KML, MapInfo FIle, Memory, MySQL, ODBC, PostgreSQL, S57, SQLite, TIGER

So what can you do basically if someone asks me with GDAL, so GDAL can do everything that is on the planet for a raster module, so it can report information about a file copy a raster file with control of an output format, wrap an image into a new coordinate system. Okay, we have seen this how; I think Sachidanand spoke about how a new coordinate system it can be transformed.

Then convert nearly black or white borders to the exact values, okay. So this also can be handled effectively by GDAL tools. It can derive contours from DEM tools to analyze and visualize DEMS. Build a quick mosaic and rasterize vectors into a raster file. Transform coordinates anything that you asked for as far as raster data; raster data module is concerned. So, all of these things can be done by GDAL.

And when you look at a OGR, OGR has huge number of formats that it can support whether it has AVCBin whether it has BNA, CSV, CSV is your Comma Separated Value format. Then you have DGN, then you have Shapefile, GML. Now a well-known language GML is being used in a lot of contexts extremely flexible language, then you have extension and then you have GMT, GPX, GRASS. GRASS has its own thing through OGR sources in QGIS.

Then you have MySQL, SDTS, SQLite, NTF, TIGER so all of these things can be handled very effectively through the OGR in QGIS. Okay. When you are looking at OGR targets any of these

I mean the way it is saved whatever it has whatever the target you need they; the well-known targets every target or every extension is possible with OGR that is a backbone of vector data module in QGIS.

So when; so just took; put thing in your mind GDAL and OGR are two libraries which are actually powering, GDAL is powering a raster data model and OGR is powering a vector data model. So with combination of this the other tools are being built and you have a beautiful interface that has been shown to you so that you do not go into command line and use it though that is more flexible, okay. So this basically both of these are handling entire GIS software's. So if you understand both of these you do not need to buy any GIS software, okay. So it is much easier than buying a GIS software.

(Refer Slide Time: 09:57)

GRASS GIS - The World Leading *Free GIS* Software, one of the world's biggest open source project, official project of the Open Source Geospatial Foundation

Welcome to GRASS GIS

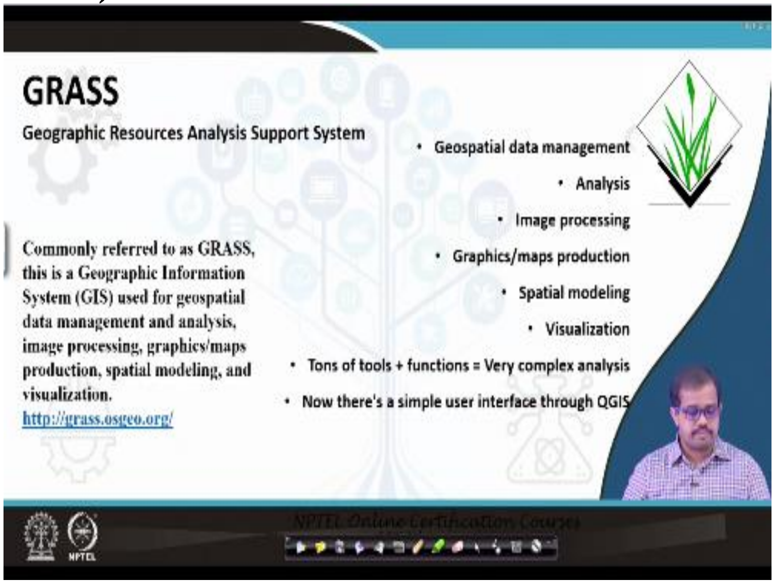
Geographic Resources Analysis Support System

Commonly referred to as GRASS, this is free Geographic Information System (GIS) software used for geospatial data management and analysis, image processing, graphics/maps production, spatial modeling, and visualization. GRASS is currently used in academic and commercial settings around the world, as well as by many governmental agencies and environmental consulting companies. GRASS is an official project of the Open Source Geospatial Foundation.

So GRASS GIS is one of those extremely good raster processing software; open source software available today. And GRASS GIS as a world over a period of time and it has the world's biggest funded open source software program on earth today. So when you look at GRASS GIS, GRASS GIS is very effective and most importantly GRASS GIS is a complete software whether you are looking at working with Python or whether you are working directly as a standalone software GRASS GIS can handle anything, okay.

So it offers an environment of wxPython wherein you can start coding by yourself use the modules and open source that have been already been built with the GRASS GIS and develop your own context whenever you are using it.

(Refer Slide Time: 10:49)



GRASS

Geographic Resources Analysis Support System

Commonly referred to as GRASS, this is a Geographic Information System (GIS) used for geospatial data management and analysis, image processing, graphics/maps production, spatial modeling, and visualization.

http://grass.osgeo.org/

- Geospatial data management
- Analysis
- Image processing
- Graphics/maps production
- Spatial modeling
- Visualization
- Tons of tools + functions = Very complex analysis
- Now there's a simple user interface through QGIS

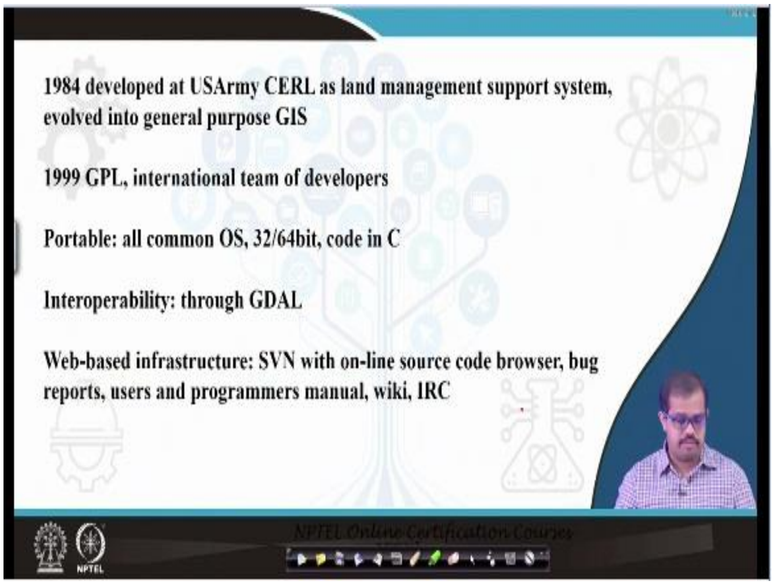
So it is abbreviated as GRASS, GRASS is nothing but Geographic Resource Analysis Support System. And when you look at it, it is commonly referred as GRASS, just as GRASS.

So it is; when I say GRASS software it is one of those effectively use; I mean GIS software of; basically a very effective in raster data processing. And when you are looking at geographic information system it can look at anything under the Sun that can be processed using a GIS software. Anything that can be modulated in terms of GIS software can be done with GRASS and when you are looking at whether at spatial modeling, whether you are looking at the production of graphics, maps or when you are looking at analysis basically image analysis.

So it is extremely useful software available today. So I would suggest if students to have a look at this particular website and probably even look at its functionality you will be assured that you will not go back to any proprietary software for terms of using it with raster processing. Then you have geospatial data management, analysis, image processing, graphical maps, spatial modeling, visualization then any complex analysis and most importantly a support to Python.

So you can use it; its code and; I mean modify it the way you need and present your research output, okay. So; and also most importantly is that the QGIS is well-known so it has a support with QGIS in; from the QGIS you can call GRASS and use it as one of the raster processing tool inside from the QGIS itself.

(Refer Slide Time: 12:39)



1984 developed at USArmy CERL as land management support system, evolved into general purpose GIS

1999 GPL, international team of developers

Portable: all common OS, 32/64bit, code in C

Interoperability: through GDAL

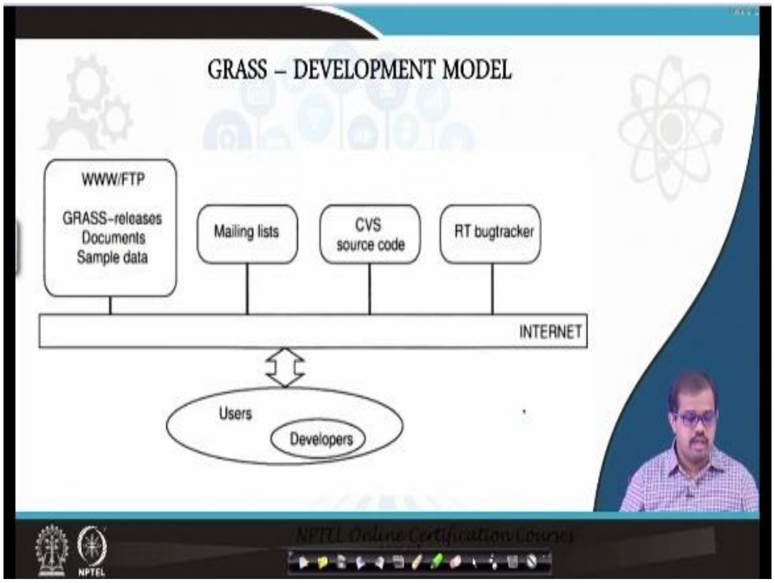
Web-based infrastructure: SVN with on-line source code browser, bug reports, users and programmers manual, wiki, IRC

So it was developed in 1984 for the US Army as a land management support system and evolved as a general-purpose GIS. In 1999 GPL, international team of developers who are handled this. It has as actually portable any of the common OS, it can be used whether it is 32 or a 64-bit; only thing is that please see what kind of architecture your particular system supports, so whether it is a 32 bit or 64 bit and use this particular software then normally it is most of it is coded in C.

It is interoperability through GDAL. It extends interoperability to any software's through GTL, okay. And more importantly now today; in today's scenario you need something that is a web based infrastructure. So it supports SVN and online source code browser which with bug reports users and programmer’s manual wiki, IRC etcetera so that you have both web based infrastructure that is already; there community of users who is actually looking at it.

The source code that is available, and most importantly any bug that is being reported is being addressed or phasely manner, very fast in fact in terms of hours also. The turnaround time may be sometimes even less than 24 hours, so that is how GRASS is; has become extensively used in a very large community a worldwide today.

(Refer Slide Time: 14:15)



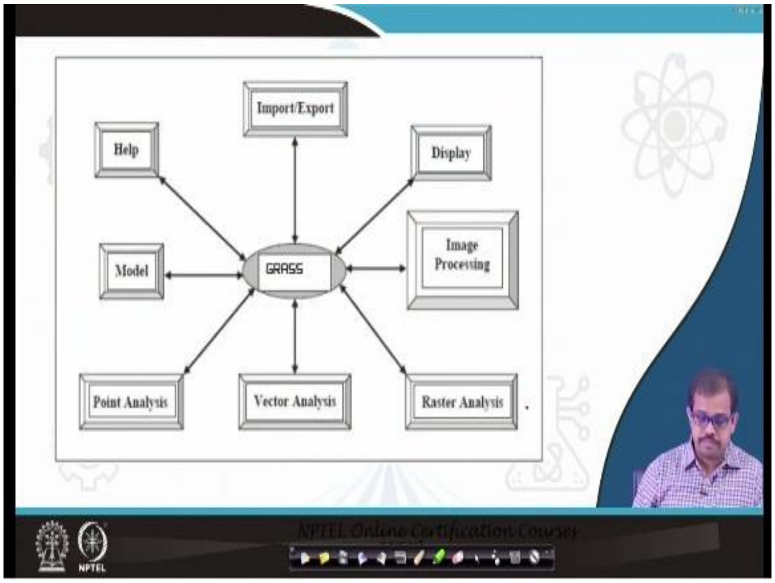
GRASS – DEVELOPMENT MODEL

So this is what is a development model of GRASS, so we had GRASS; first thing is we had developers and users which who are using they have an interface through internet to through a bugtracker. There is an RT bugtracker where most of the bugs are reported. And this is actually handled by the patches who actually look at these bugs and patch that. Then you have CVS source codes available then you have mailing list.

Mailing list here if you just join your mailing; that mailing list, you can see huge number of users who are actually replying to every aspect of research needs or any researchers working on. So if someone wants to or he stuck at certain point of time just addressing this mailing; I am sure that someone will reply to you with saying; the solution that is; that may be available or maybe in the direction towards the solution which you may have to work.

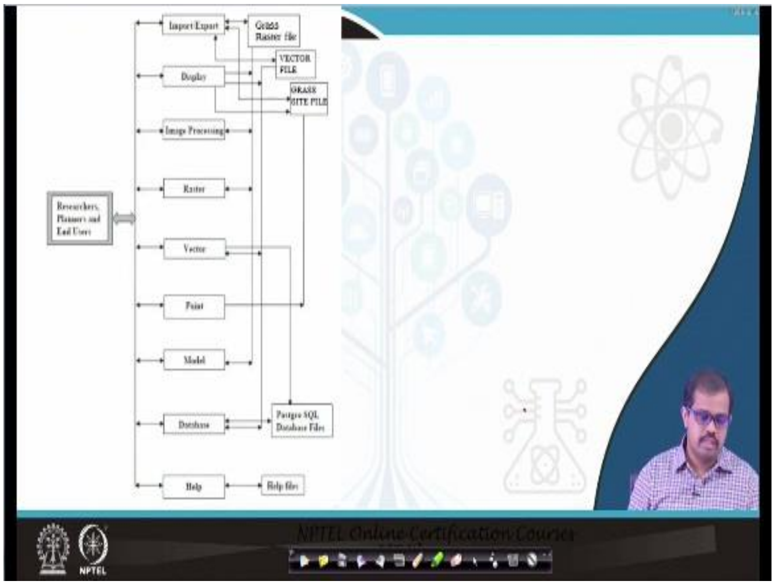
So it is extremely good in terms of mailing list. And then you have GRASS-releases that is a documented sample data that is available already available you can download it when you are downloading the software or installing the software, so that is how the entire model is quite good in terms of development and usage.

(Refer Slide Time: 15:37)



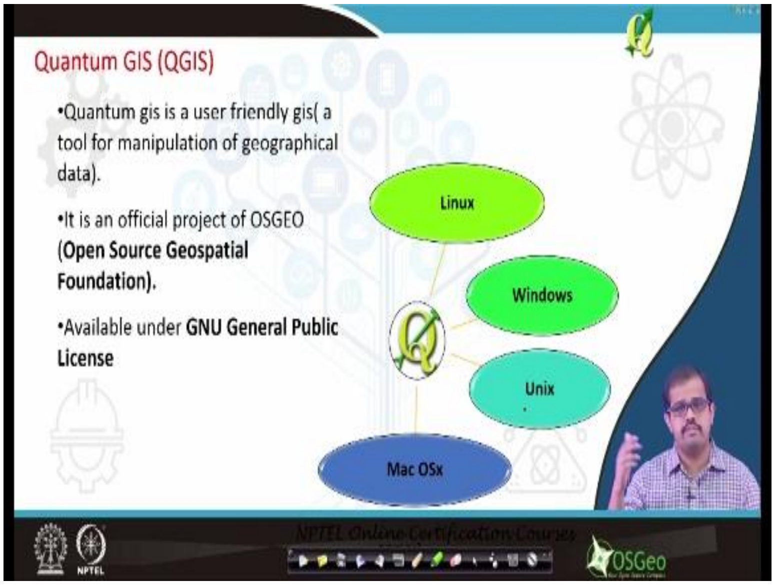
And when you look at GRASS as a software, you have you can develop it as, as I said where you can I import or export any kind of raster or vector formats; you can display all your raster or vector data; you can process an image; you can look at raster analysis; you can look at vector analysis; you can look at point data analysis then most importantly you can; once you have developed all your data you can even modulate. So that is how the GRASS is extremely useful as entire as a raster processing software, okay.

(Refer Slide Time: 16:09)



Now when you are looking at a researcher or a planner you can now probably look at import-export; you have a display, you have image processing and as I said, it also can handle the database that is through PostgreSQL database files more effectively than most of the software proprietary software's that are available today. So this is just to tell you the entire module of how GRASS GIS is built. So you ask for anything that you have to work, so most of the things are available with GRASS GIS.

(Refer Slide Time: 16:44)



Quantum GIS (QGIS)

- Quantum gis is a user friendly gis(a tool for manipulation of geographical data).
- It is an official project of OSGEO **(Open Source Geospatial Foundation).**
- Available under **GNU General Public License**

Linux

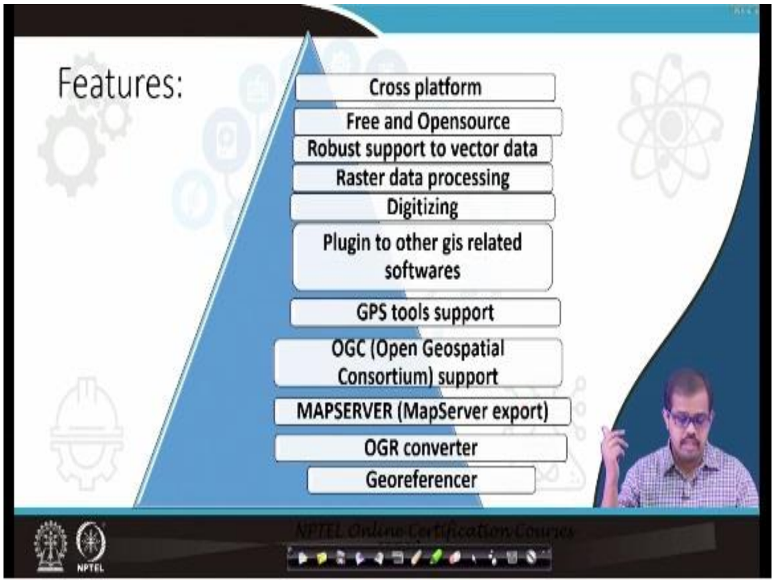
Windows

Unix

Mac OSx

So the next thing is the Quantum GIS is this you have seen. So it is a user friendly GIS I would say it is a collection of tools or a software which primarily is an official project of OSGEO that is a open source geo-spatial foundation. It is under the GNU Public License. So I have spoke about the license already it is under the GNU Public License. And when you look at QGIS which is available with Linux; it is available with Windows; it is available with Unix, Mac OSx and if; others if any, so most of these users can use QGIS.

(Refer Slide Time: 17:25)



Features:

- Cross platform
- Free and Opensource
- Robust support to vector data
- Raster data processing
- Digitizing
- Plugin to other gis related softwares
- GPS tools support
- OGC (Open Geospatial Consortium) support
- MAPSERVER (MapServer export)
- OGR converter
- Georeferencer

NPTEL Online Certification Courses

And when you look at QGIS as a feature, okay. Software's there are certain features of a particular software why you like that particular software. For the first feature that the most important thing is cross-platform. For example, may be in your college you have a Windows platform and at home you may using a Linux platform. So whatever you have developed there can be easily cross-platform here, okay.

So it is not that you are stuck with only one and you have to do with that. So it is cross-platform you can install in any kind of platform that or an operating system that you can have. Most important is the free and open source available free of cost, okay. Not even licensing or a distributing cost and most important it is open source. That is why it is for a helpful for most of the researchers around the world.

Robust support to vector data, vector data sets prime point, handling vector data or is; where QGIS excels, okay vector data model. So when you are probably you have seen how well it can handle most of the vector data in last two weeks of this particular course. Then you have raster data processing, this is with the help of GRASS. So as we spoke previously GRASS is very effective in terms of handling raster data so this combines GRASS with itself.

So the raster data processing is also quite effective. Digitizing, so like you do in any other GIS software proprietary GIS software it can do 2D and 3D digitizing in much easier way the same way that you do in most of the other software. Plug-in to other GIS related software. Yes, it has lot of plug-ins. You can plug-in with other GIS software related software.

And also it has a repository of plug-ins which you can download and install which you have already seen and can; so that is how it is very different from many of the other software’s. And GPS tools, so, many proprietary software do not offer a support to GPS in terms of when you have collected data and you have to put it on the right platform it may not be possible. So, GPS tools support is available in QGIS.

Then you have OGC that is Open Geospatial Consortium support which again is extremely good, which is actually the basic part of any of these open source software. And the; if someone is looking at today's context most of the thing is that you have to develop it on the web. Anyone who are looking at today's developments it is through a web service, okay whether it is a map or a feature; so you are looking at a web service.

So here it is; it supports map server which can handle the data the both raster and vector data very effectively and you can utilize it as a web service so that anyone who wants to use your data visualized with your data visualized with your code without having any kind of hardware or a software can easily look at that just by logging into that particular URL with a web browser support; as simple as that.

Anyone with a mobile phone can get into that. So that is why today's context is to; is developing towards that, the decision support is happening towards developing and more importantly the portable kind of decisions that can be made at every instant of time without having a large amount of hardware and software. So that is where we are moving towards and yes QGIS provides support for that also. Then you have OGR converter.

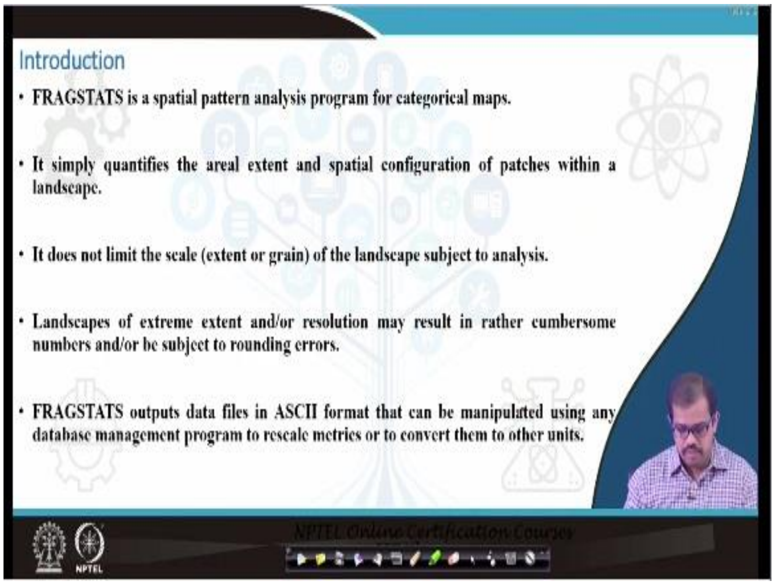
So open geospatial raster converter you can convert most of your formats as I said is interoperability can be there and it is provided through GDAL. Then you have Georeferencer, so you have used georeferencer I would not get into all of these details.

(Refer Slide Time: 21:30)

FRAGSTAT

Okay the other, the last kind of software that I would like to develop is FRAGSTAT.

(Refer Slide Time: 21:36)



Introduction

- FRAGSTATS is a spatial pattern analysis program for categorical maps.
- It simply quantifies the areal extent and spatial configuration of patches within a landscape.
- It does not limit the scale (extent or grain) of the landscape subject to analysis.
- Landscapes of extreme extent and/or resolution may result in rather cumbersome numbers and/or be subject to rounding errors.
- FRAGSTATS outputs data files in ASCII format that can be manipulated using any database management program to rescale metrics or to convert them to other units.

So just to tell you what is FRAGSTATS, it is a spatial pattern analysis program for categorical maps. For example, let us say, I have developed a map of land use of a particular region. Let us say the region is Kharagpur. So now I want to understand I have a map of 1990, I have a map of 2000, I have a map of 2010 and I have a map of 2019. Now I want to understand how Kharagpur has grown over a period of time, in 1990 it was probably only IIT-Kharagpur and maybe some settlement just next to it.

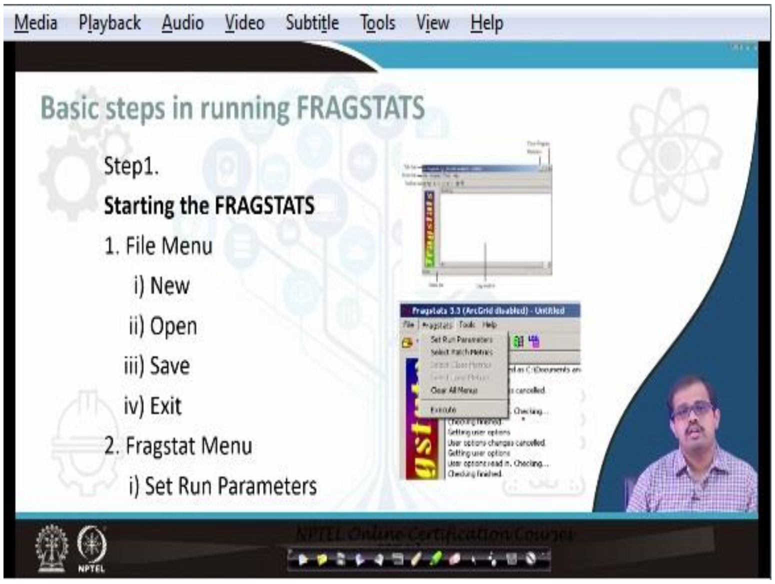
Now once in 2000 there it may have grown over entire the railway block, okay. In 2010 it has going beyond the railway block and maybe reaching the next location. And in; maybe in 2019 it has developed. So how it has developed? What kind of development has happened? Okay, whether it is developing in patches; whether it is developing classes; whether it is developing as a landscape, okay.

What kind of development, whether you (()) (22:40), whether you can call it as a cluster development; so such kind of analysis can be done for any urban context. So this can be done using FRAGSTAT. So it simply quantifies the areal extent and spatial configuration of patches within the landscape or landscape analysis in the landscape. So any kind of thing you can do. It does not limit the scale of a landscape subject to analysis; that is the best part of FRAGSTAT.

So many of the software for example there is something called (()) (23:11) which is actually limits the scale which FRAGSTAT does not. Then you have landscape of extreme extent or resolution may result in rather cumbersome numbers or to be subject to the rounding errors. So please be careful. So if it has a very coarse resolution you may not; none of these will be very effective in analyzing your data.

If it is a medium resolution or very high resolution data okay very high spatial resolution data then you are; most of these whatever I am speaking about the tool is extremely useful in urban planning. FRAGSTAT outputs normally in ASCII file format that can be manipulated using any database management program to rescale metrics to convert them to other any of the other units, okay.

(Refer Slide Time: 24:01)



Basic steps in running FRAGSTATS

Step1.

Starting the FRAGSTATS

1. File Menu
   i) New
   ii) Open
   iii) Save
   iv) Exit
2. Fragstat Menu
   i) Set Run Parameters

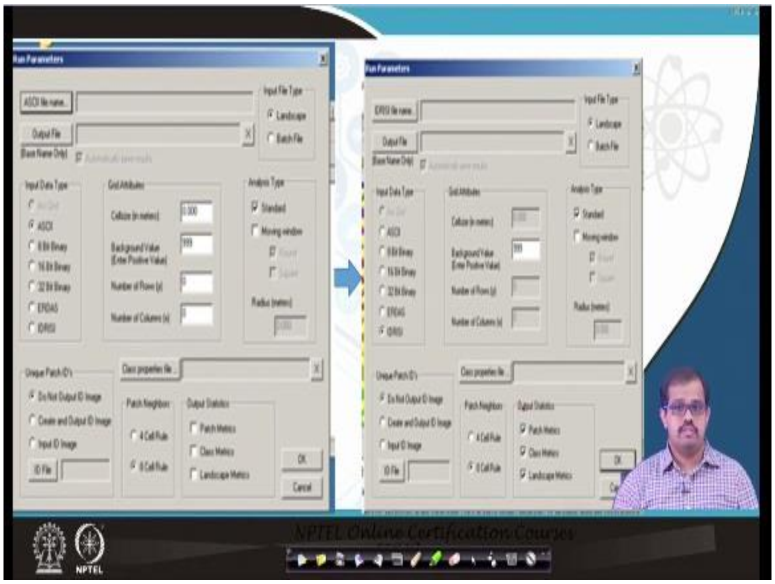
```
Checking finished.
Getting user options
User options changes cancelled.
Getting user options
User options read in. Checking...
Checking finished.
```

So I will just give you a brief about how FRAGSTATS works. For example, FRAGSTAT can be just type in the Google you can download FRAGSTAT, FRAGSTAT is start is a very small program maybe in terms of MBs you download it. And when you double click it you get in the portable format you get even the installable formats. I am using the version 3.3 which allow it the most the recent; there are recent versions of 4, now it is 5.

So all these versions are there more modified versions better versions with large number of metrics are available. But I am very comfortable with 3.3 so I am giving you an example of with FRAGSTAT 3.3. So this has a huge amount of metrics that you can use and these metrics. See these metrics are not only made for urban analysis; this can be then used to our ecological analysis, it can be used for; and one basically it was done for ecological analysis.

Now it is being used as an urban analytical tool also it can be used for environmental analytical tool, it can be used for hydrological analytical any kind of analysis that you want to do in terms of looking at patches configuration etcetera on the landscape that can be done with FRAGSTAT. So first thing is the file menu which actually gives you how do you open; how do you create a new particular instance in order to run the FRAGSTAT then you have open, save etcetera. The most important part here is Set Run Parameters, okay.

(Refer Slide Time: 25:38)



When I say run parameters this is how it looks, okay. It can take an ASCII file name; it can take an aggressive file name, okay. So when, when you are actually putting in the run parameters you have to give an ASCII file or an aggressive file, okay. Then you can even give an 8-bit binary here or a 16-bit binary or 32-bit binary. Then you have ERDAS format can be directly used and IDRISI receive format can be directly used.

So this is a input kind of data. Based on this data this would change, okay. Now once you have given this, let us say you have 100's of the same kind of data that you have to use, okay; same analysis that has to be repeated over maybe you are looking at concentric circles maybe squares, so if you want to do it you can always automated with the batch file, okay create a batch file and automated which can be used as an instance here, okay.

If you use this button radio button, then it will create a batch file. But otherwise you can use it as a input data only. Now then you have Grid attributes. The cell size in meters is your resolution of your satellite data, okay. If it is land site so you may enter a test 30 otherwise, you may enter whatever its resolution is. Then you back; you have background values then you have number of rows and columns.

If you look at metadata of your satellite image that is available, then it gives you a number of rows and number of columns. So that has to be entered here. Then you have analysis type one is a Standard and the one is a Moving window. So there are two ways of looking at it. Standard is normally used when you have; when you have running parameters which is quite standard to the analysis of whatever you are trying to do.

But Moving window gives you a different kind of understanding that how you way each and every parameters across your research domain. So this also can be looked at in a different way. So probably if you want to look at it please look at some of the literatures that are available. Then most importantly you have to; input here as a class property file. So when I say the class property file so what basically it is, is a description of whatever the raster image that you are entering here as a input data or a vector image that you are inputting as a input data.

So what basically that is for example when I; I am inputting it as a land use data. So I may have created a land use with urban, vegetation, water others. So with this you have to put in the class properties files as what are the different, you know classes that are available. How many classes have there? Whether it is 4 class, 5 class, 6 class, okay and what is the name of that class. For example, the first one is urban; the second one is water; third one is vegetation; fourth one is others and rest is nothing but null.

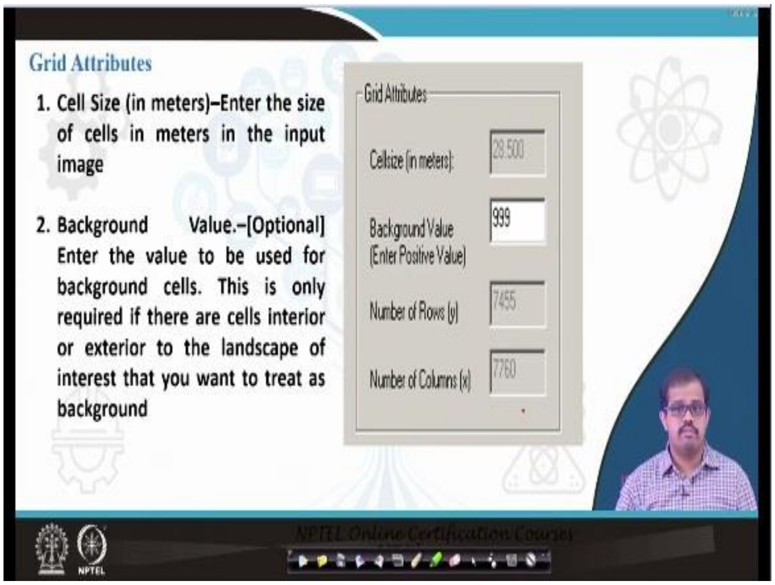
So this has to be inputted as a class properties file in this particular tab or you can even develop that particular file. Then you have patched neighbors how the rule has to be whether it is looking at the 4 cell neighbors the 4 neighbors or whether it is looking at the 8-cell neighbours. Then finally the output statistics whether you want to look at patch. For example, if in; if you are looking at urban analysis, let us take a city; not city of today.

Let us say a tier III city which is actually developing. Now in the tier III city you may have a core area, okay. There may be some towns away from the core area 3-4 towns. Even inside the municipality boundary there will be other classes also that has been developed. So which means they are in patches. So if you are looking at how many patches are there; how it is actually converting; how it is actually changing over a period of time.

For example, if there is 100 patches at today, so you say that this is kind of a patchy growth. And higher the number of patches it means to say that there is clutter there are many other classes that are there in a land use configuration. But if the number of patches starts decreasing and then becomes a single patch it means entire landscape that you are analyzing has become a single urban patch; it means everything has converted to urban, right.

So that is one of the; kind of analysis that you can do, that is why you look at the patch metrics. But when you are comparing across a class so you are looking at class metrics; if you are looking at overall understanding of a landscape that you are inputting then you are looking at landscape metrics. So these are different ways of you; of things that has to be inputted in terms of understanding over a FRAGSTAT.

(Refer Slide Time: 30:30)



Grid Attributes

1. Cell Size (in meters)–Enter the size of cells in meters in the input image
2. Background Value.–[Optional] Enter the value to be used for background cells. This is only required if there are cells interior or exterior to the landscape of interest that you want to treat as background

Grid Attributes

Cellsize (in meters): 28.500

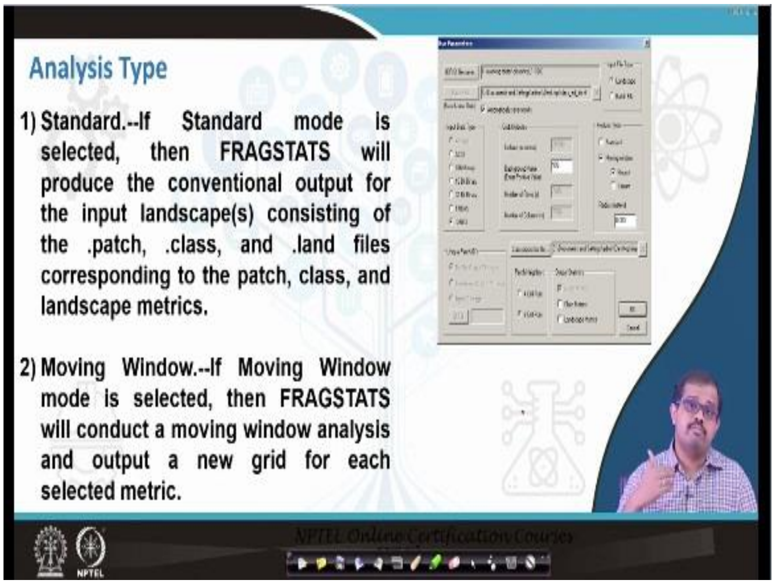
Background Value (Enter Positive Value) 999

Number of Rows (y) 7455

Number of Columns (x) 7760

So this is what is; what I have mentioned so you can look at it all of these information so when leisure, okay.

(Refer Slide Time: 30:39)



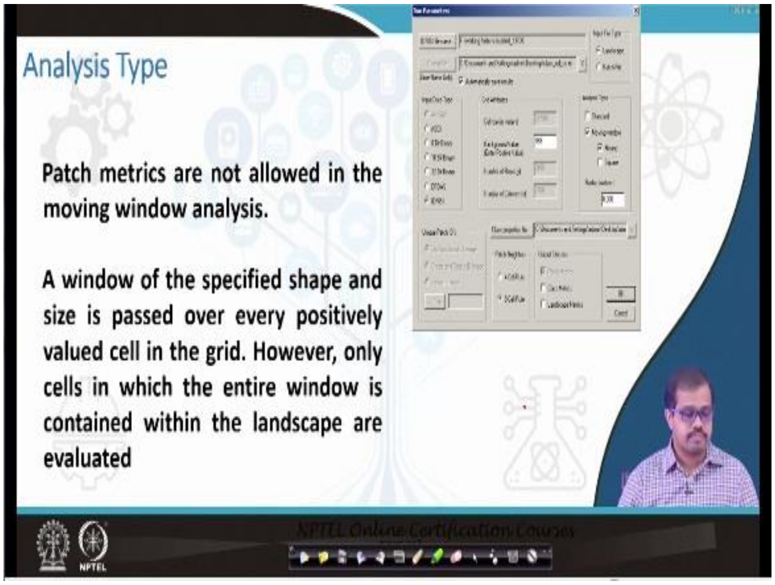
Analysis Type

1) Standard.--If Standard mode is selected, then FRAGSTATS will produce the conventional output for the input landscape(s) consisting of the .patch, .class, and .land files corresponding to the patch, class, and landscape metrics.

2) Moving Window.--If Moving Window mode is selected, then FRAGSTATS will conduct a moving window analysis and output a new grid for each selected metric.

So, yeah, so I said about the standard analysis and the moving window analysis. If it is a standard mode then the FRAGSTATS will produce a conventional outputs that is for input landscape consisting of dot patch, dot class, dot land files okay. If it is a moving window analysis, it understand it conducts a moving window unless in terms of output as a new grid. It produces a new grid for each of the selected metrics.

(Refer Slide Time: 31:07)



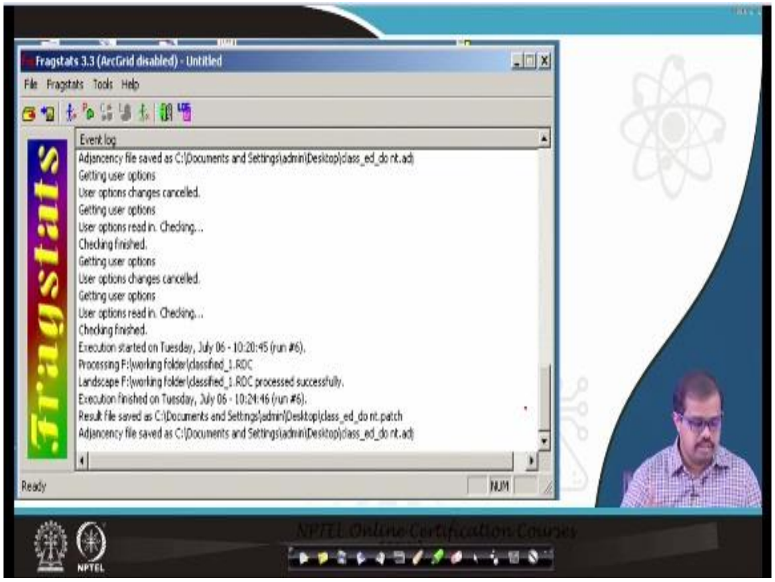
Analysis Type

Patch metrics are not allowed in the moving window analysis.

A window of the specified shape and size is passed over every positively valued cell in the grid. However, only cells in which the entire window is contained within the landscape are evaluated

So these; this is what I have spoke already, not getting to this.

(Refer Slide Time: 31:11)

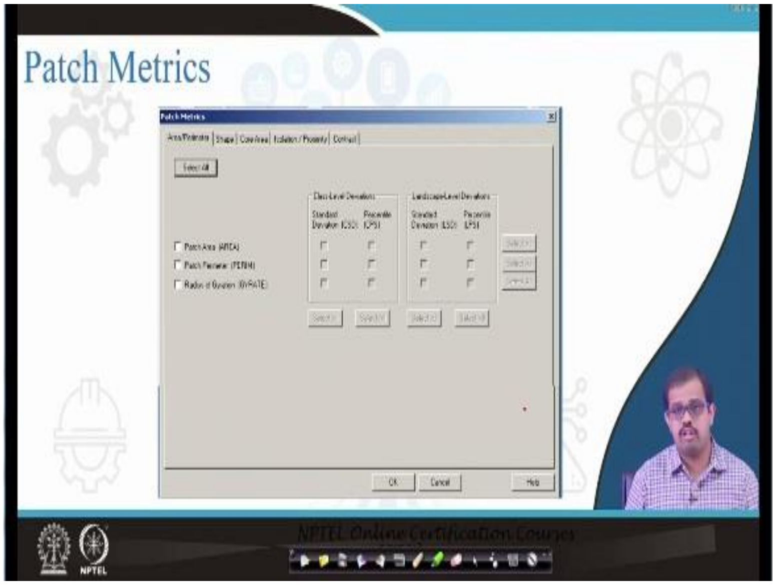


Fragstats 3.3 (ArcGrid disabled) - Untitled
File Fragstats Tools Help
Fragstats
Ready
NUM
NPTEL
NPTEL Online Certification Courses

```
Event log
Adjancency file saved as C:\Documents and Settings\admin\Desktop\class_ed_do nt.adj
Getting user options
User options changes cancelled.
Getting user options
User options read in. Checking...
Checking finished.
Getting user options
User options changes cancelled.
Getting user options
User options read in. Checking...
Checking finished.
Execution started on Tuesday, July 06 - 10:20:45 (run #6).
Processing F:\working folder\classified_1.RDC
Landscape F:\working folder\classified_1.RDC processed successfully.
Execution finished on Tuesday, July 06 - 10:24:46 (run #6).
Result file saved as C:\Documents and Settings\admin\Desktop\class_ed_do nt.patch
Adjancency file saved as C:\Documents and Settings\admin\Desktop\class_ed_do nt.adj
```

This is how we transit processes here for the first file that you will be working with. Then this is where it has finished its run. Then the results have saved somewhere here as patch I have selected as a patch metrics so results are saved here. And you have adjacency file which gives more information about it.

(Refer Slide Time: 31:33)

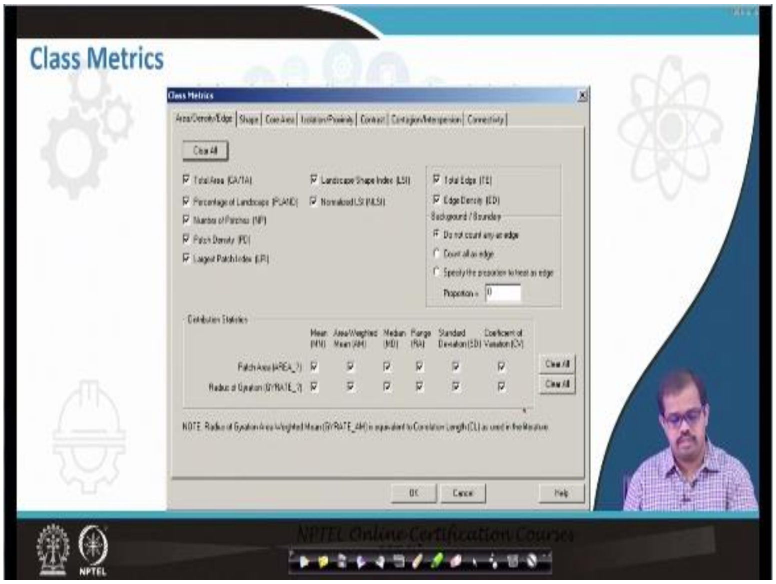


Patch Metrics

☐ Patch Area (AREA)
☐ Patch Perimeter (PERIM)
☐ Radius of Gyration (GYRATE)

Then these are some of the metrics I would not get into all of these details, so whichever metrics you need you can just select it off then you have land metrics.

(Refer Slide Time: 31:41)



Class Metrics

Class Metrics

☑ Total Area (CA/TA)
☑ Percentage of Landscape (PLAND)
☑ Number of Patches (NP)
☑ Patch Density (PD)
☑ Largest Patch Index (LPI)
☑ Landscape Shape Index (LSI)
☑ Normalized LSI (nLSI)
☑ Total Edge (TE)
☑ Edge Density (ED)

Background / Boundary
● Do not count any as edge
○ Count all as edge
○ Specify the proportion to treat as edge
Proportion = 0

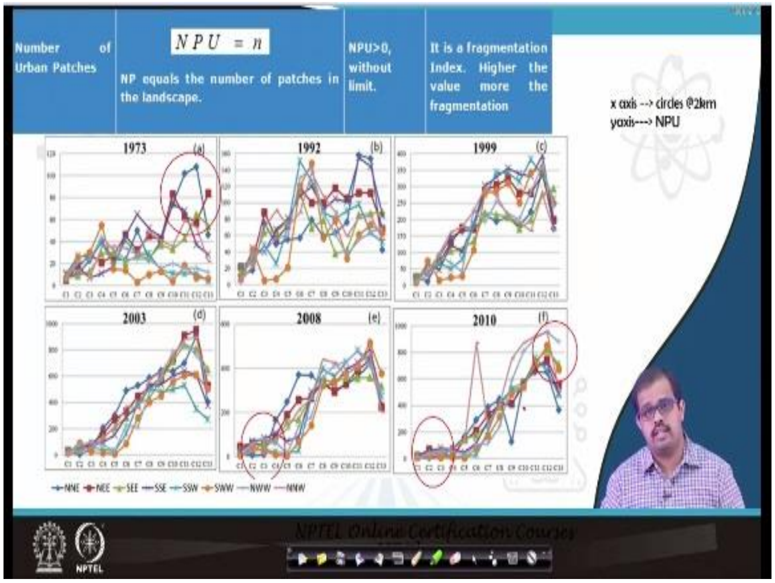
Distribution Statistics

	Mean (MN)	Area-Weighted Mean (AM)	Median (MD)	Range (RA)	Standard Deviation (SD)	Coefficient of Variation (CV)
Patch Area (AREA_?)	☑	☑	☑	☑	☑	☑
Radius of Gyration (GYRATE_?)	☑	☑	☑	☑	☑	☑

NOTE: Radius of Gyration Area-Weighted Mean (GYRATE_AM) is equivalent to Correlation Length (CL) as used in the literature

Then you have class metrics. There you can select whichever the metrics that is appropriate for your analysis, okay.

(Refer Slide Time: 31:50)



Number of Urban Patches

$NPU = n$

NP equals the number of patches in the landscape.

NPU>0, without limit.

It is a fragmentation Index. Higher the value more the fragmentation

x axis --> circles @2km
yaxis---> NPU

So let me give you one example. For example, this analysis done over Bangalore, the growth of Bangalore from 1973 to 2010 from a very small region urban region to the today and agglomeration. So I can, we can even call it as a metropolitan city with maybe in few years. So it has grown out of bounds today. What we did was we looked at the Bangalore as a city then we divided it into four zones, okay.

And also we looked at in terms of concentric circle. Here when you look at it we divided the Bangalore from as a two kilometer concentric circles from the center you have C1 to C30. So Bangalore with a 10 kilometer buffer was 26 kilometers from the center if you measure the radius it was 26 kilometers. So when we divided it you have 13 circles. So every circle was input as input file so which means we created a batch file of 13 different raster files.

So with this we understood the patch metric, the one patch metrics that we have understood is number of urban patches which means say that how many urban patches were there in 1973. How many urban patches were there in 1992, in that particular circle? How many urban patches were there in 1999 in that particular circle and how many urban patches were there in 2010? Now when you see this, okay; and it is with directions.

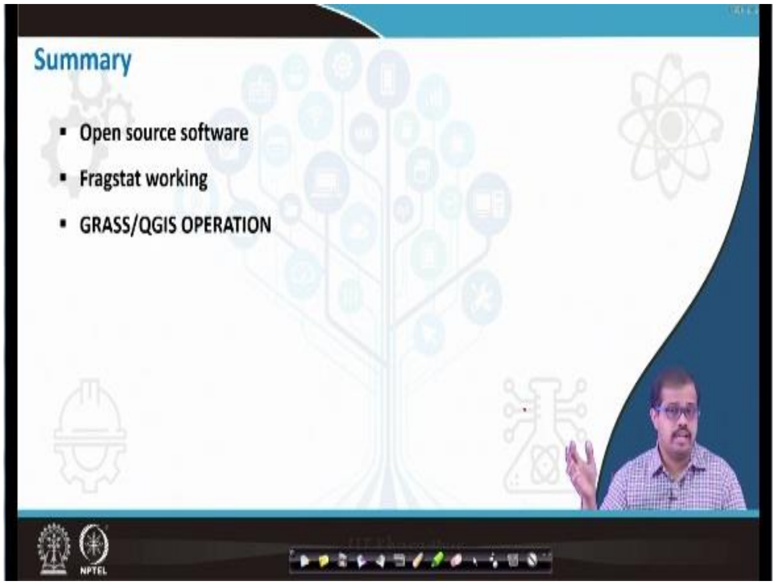
So I have put the directions here that we have considered 8 directions. So when you look at this if let us say that I have taken southwest direction. You can see here the urban patch in the core were very less which means that instead of patch it was completely a single patch and no other classes were there in the core whereas when you looked at the outskirts the patches were less but when you; as you go on the patches increased.

Here the urban area was less but when you; as you went on the urban area increased and the patches also increased. When the patches have increased you can see the core almost closing down to a single patch, okay. Here in 2010 core as already in all directions have closed down to a single patch which means the entire core area of a Bangalore has completely urbanized whereas the outskirts of the city is cluttered format which means it is sprawling away from the city, city end somewhere in the 8th circle or the 9th circle and then it has started sprawling which means the outskirts of the city you can attribute it to various industrial growth.

For example, 1992 we set up the; 1990s we set up our industrial units then the educational institutions came up in various nook and corners of the state of; sorry of the city. Then in 2000 with the IT revolution Bangalore became the Silicon Valley so you had growth in those regions for example the Koramangala region, the white field region etcetera in 2000-2005-2006 onwards. So you can see in those regions actually pulling up in a large way.

Then now the consolidation has started back if you look at 2019 published literature the consolidation of all of this has started and it; now the threshold is being reached as far as urban growth is concerned.

(Refer Slide Time: 35:20)



Summary

- Open source software
- Fragstat working
- GRASS/QGIS OPERATION

So on a summary we looked at the open source; different open source software. We started with OSSIM as an library. Then we looked at GDAL, we looked at OGR how effectively it can handle data. Then we looked at software's GRASS, QGIS operations and most importantly we looked at FRAGSTATS that how we can be used for any research analysis. And finally I gave you an output of how FRAGSTAT can be used for different than urban analysis.

So with this I would end this class. In the next class we will learn more about different aspects of open source and its connected issues and the way it has been developed, okay. Till then have a nice time. Thank you very much.