Journal of GeoPython

Volume 1, June 20, 2016

Editor: Martin Christen
FHNW - University of Applied Sciences and Arts Northwestern Switzerland

Journal of GeoPython (Online)

ISSN 2297-928X

9 772297 928008

Journal of GeoPython (CD-ROM)

ISSN 2297-9336

9 772297 933002

Table of Contents

Fast Data

Moving beyond from Big Data's map-reduce

Adam Lev-Libfeld, Alexander Margolin
Tamar Labs
Tel-Aviv, Israel
adam@tamarlabs.com

Abstract—**Big Data may not be the solution many are looking for. The latest rise of Big Data methods and systems is partly due to the new abilities these techniques provide, partly to the simplicity of the software design and partly because the buzzword itself has value to investors and clients. That said, popularity is not a measure for suitability and the Big Data approach might not be the best solution, or even an applicable one, to many common problems. Namely, time dependent problems whose solution may be bound or cached in any manner can benefit greatly from moving to partly stateless, flow oriented functions and data models. This paper presents such a model to substitute the traditional map-shuffle-reduce models.**

Keywords—fast data; performance; big data scalabuility; filter-split-dehydrate

I. INTRODUCTION

The map-reduce model is the de facto standard for processing large amounts of data (mostly known as Big Data). This technique is based on the idea that for every given problem our dataset holds some form of the solution, which can be found by permutation of the data items, and fusion of that permutated data in a way that eliminates all unnecessary information in the data set [1][3]. This set of assumptions is generic by design and was created as such to service the needs of emerging data warehouse concept [2][3]. As often happens once a method gains popularity, there will also be some who will use it without consideration of appropriateness, applicability or even usefulness, same is the case for map-reduce, where companies use the Big Data buzzword to instill trust in clients or investors, and even scare off the competition.

As industry acknowledges these changes, players are moving towards more tightly constrained tools to mitigate the effects of using map-reduce in an inappropriate manner [4], these are usually surface solutions to the underlying problematic assumptions on the nature of the problems they are trying to solve, and the data they hold. The solution proposed in this article is part of a new approach to looking at problems. This approach, although already in use by many (mostly under the names "Stream Processing"[6][7]) has yet to find a leading paradigm and is currently a collection of tools and techniques [4][5][6] more than an architectural model to be used to the benefit of the solution. This article presents one such paradigm.

II. ASSUMPTIONS AND REALITIES

The map-reduce paradigm is based on several assumptions which, in order to satisfy, software engineers go to great lengths in changing their system. These changes might hinder the performance and maintainability of the system as a whole, in a manner that will overcome any advantages the system may have gained from using map-reduce.

In order to deal with these inaccurate assumptions, one must first identify them, three of these assumptions are:

A. Completeness of Data

It is assumed that the answer is, or can be derived from the data we have at any given moment.

B. Independence of Data Set Calculations

It is assumed that any action performed on the data set is independent from any other in any way, and can be done concurrently with no side effects.

C. Relevancy Distinguishability

It is assumed that we can and will distinguish which data is relevant and which is irrelevant to the calculation at any given moment.

Once we call attention to these assumptions, it is easy to see why they were so appealing (and confusing). In the real world we usually face something very similar, but inherently different:

D. *Contextual Completeness*

The data we have is complete only in the context in which it arrived. As our context is time bound, and time is constantly progressing, the information may never be complete and we may never have all the information needed to form a perfect[1] answer, or even a correct one.

E. *Partially Dependent Calculation*

There exists a *series* of calculations, each takes as input the output of its' predecessor, for which data (input) needed is less dependent in every step. i.e. the further you go down the chain of calculations, the more independent and less state aware each calculation is, as state has been partially resolved by previous calculations.

F. *Emergent Relevancy*

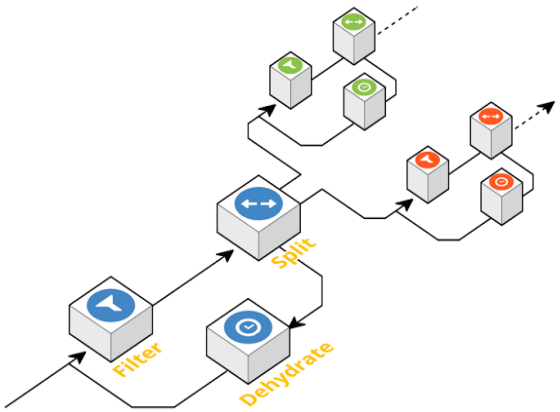
In essence the complement of context completeness, it is not only uncertain that we will have complete information to form an answer, but it is also possible that we would not be able to distinguish between several options which would lead to a correct answer and which will not. Having that said, upon trying different answers we can deduce which answer is best, in a manner that will eventually produce the best possible answer (that is - the least irrelevant).

III. Handling Uncertinty

In order to deal with these complex realities one might, as many do, erect mechanisms to "clean" the uncertainties out of the system. These mechanisms tend to have significant cost, in either computation time system complexity or both. We suggest a formalized approach that, by taking context and uncertainty into account enables the creation of cleaner (and simpler) system design, which is easier to maintain, extend and scale.

The new approach consists of three main elements: Filter, Splitter and Dehydrator.

These three elements work in unison to perform the same objective of the map-reduce, filters acting as reducers and splitters as mapping/shuffling operations. The objective of the dehydrator element is to create a controlled feedback loop; effectively prolonging past context so greater meaning could be derived from latest information.



IV. Filter

The filter acts as a *reducing* measure to the incoming flow of data, enabling following elements to be more specific as they only have to process data which is already assumed to be (at least somewhat) relevant.

Stateless Filters

These can be applied to a single input, or to a whole data array in parallel. This filter's only input is the data element itself, and it has no regard to the results of neighboring filters.

Aggregation (top X) Filters

The stateful[2] version of the stateless filter, the input for this filter is both a data element and some aggregation of the previous elements, this aggregation could vary from a simple internal count to a full ordered shared cache. Using several tiers of these filters can minimize shared state; each tier aggregates internally and passes a "soft" decision to the next. This soft output may also vary to consist anything from a numerical result with a notation of margin of error to a set of several elements (top X elements), only one of which is an optimal answer to the next stage of computation.

[1] Such answer may be more than we want. One of the ways we overcome this obstacle is by better defining the error margins allowed for an answer to be considered "correct".

[2] Exactly how "much" stateful this element will be can be controlled by moving shared state locally, and duplicating the data elements on the "edges" of the shared data-structure, this will lower co-dependency of the different elements in your system and enable higher scalability (in the cost of space over time, of course).

V. SPLIT

A Splitter will be used to *map* data elements to their appropriate next stage(s) through the system. This channels some of the data once contained within some form of state of a system to the topology, as the receiving stage can be sure that certain conditions apply by it being invoked.

Stateless Splitters

As in the case of the stateless filter, this splitter applies its logic only to the data it takes as input. Based on the results of the logic, the splitter routes the input (or some derivative of it created by the logic) to the next element.

Aggregation (memory) Splitters

This splitter has an internal or shared state[2], which is effected by previous inputs or results, and the splitter uses to route results of its logic to the following elements of the system.

VI. DEHYDRATE

Dehydrators are simplistic time machines. They transport data elements that arrived prematurely in terms of their context right to the future where they might be needed, without loading the system while waiting. This concept is achieved by attaching a time-indexed data store to a clock, storing elements as they arrive to the dehydrator and re-introducing them as inputs to the system once a predetermined time period has passed. In order to determine that time period one must take into account three issues:

1. At what point we consider a data element too old and retire it?
2. At what resolution do we want elements to be "retried" as input and how this resolution shift with element age?
3. Are there any exceptions and special circumstances required by the system logic that may change 1, 2 or both for a certain element?

Once these three factors are determined, constructing a scheduling process is as simple as adding a new split process which attaches each data element with it's appropriate dehydration time, and sends it to the data store.

VII. USE CASE : REAL TIME GEO-MATCHING

We shall now bring as example the case of a system built for matching "Question" elements tagged with a geospatial position (latitude, longitude and radius) and "Candidates" who are contently report on location. Both new questions and location notification from users are constantly flowing into the system. Once a candidate intersects with a question, the question had to, under some logic, be sent to that user, or if there is more than one user, to the most suitable one. A question could not be sent twice to the same candidate. If there were no candidates within the question active area, but some were on (or near) its edge the system may request a location update from these candidates. These requests are limited in number and frequency.

The original system had all the characteristics we mentioned before, as the client in this specific instance was a start-up, focus was set on speedy proof of concept (POC) and short time to market. The engineer in charge decided on using a centralized DB for the POC and this went on into the final system.

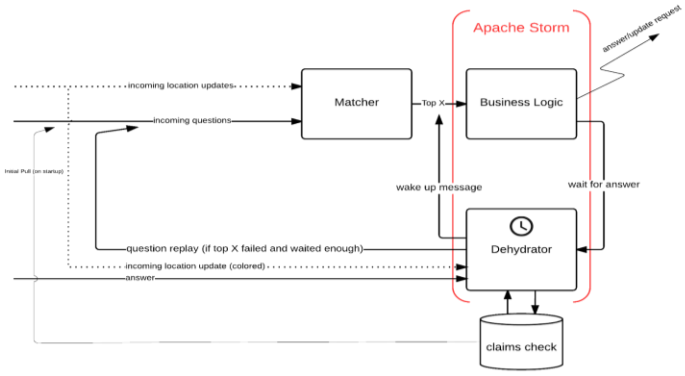
The system suffered of inability to scale and severely reduced performance even in relatively low concurrent user count.

Originally, the system design revolved around two parts:

1. A monolithic database in which all relevant data is stored (see assumption C), and it is assumed that all data needed for a certain response, if it ever arrived, will be available within it (see assumption A)
2. A set of workers managed by a job queue, each serves requests independently as they come in to form a response based on the described DB and send it back to the relevant user (see assumption B).

The inputs to the system, as mentioned above were either Question or Candidate elements; both are arriving constantly and are of varying relevancy, e.g. questions that no longer need an answer, users that are not near any question or have since moved away, and even candidates who, once matched, will not provide an answer (see assumption F).

After analyzing the available system, its problems and challenges, the suggested system structure was as such:



As can be seen in the diagram, this is a fairly straightforward single tier filter-split-dehydrate setup, with both the filter and the splitter in their aggregator variants.

The Matcher element is a cached filter - caching items using an in-memory RTree and produces upon incoming request the corresponding intersection of possible responses, intersecting an incoming question with known user locations and vice versa.

The Business logic is a memory splitter - mostly independent but shares some statistics with fellow splitters, deciding when to send user a question, a request to update location, or to just forward a question event to the dehydrator to be retried on a later date (or even retired altogether).

Holding all that shared data in an in memory database (Redis [10]) is a crucial part of the system implementation. It allowed for the different filter and splitter instances to start independently and only access the state it needed (see assumption E) in minimal latency.

In Addition, the Dehydrator holds questions as the system waits to a user response, and replays/retires each question, based on business logic. This enables for a question's context to be prolonged (see assumption D) with minimal load overhead on the system.

Upon deployment of this solution, effective performance rose by over 9 orders of magnitude with the system at its final setup supporting of millions of requests per second on a single server.

VIII. Conclution

The filter-split-dehydrate model, with its stateless, aggregatory, single-tiered and multi-tiered variants is a versatile model that enables a clean software design reminiscent of map-reduce, and adapted to a partial data reality, in which data elements flow into the system in an stochastic fashion. The likes of this model is already being used in the industry to various ends and was proven to, once used correctly to have positive results on performance and scalability.

Acknowledgments

We would like to thank Vioozer Israel [9] for allowing us to use their system diagram for the use case and data in the course of this work.

References

[1] Andrew Pavlo , Erik Paulson , Alexander Rasin , Daniel J. Abadi , David J. DeWitt , Samuel Madden , Michael Stonebraker, A comparison of approaches to large-scale data analysis, Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, June 29-July 02, 2009, Providence, Rhode Island, USA

[2] D. A. Patterson. Technical Perspective: The Data Center is the Computer. Commun. ACM, 51(1):105–105, 2008.

[3] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In OSDI '04, pages 10–10, 2004.

[4] http://spark.apache.org/streaming/

[5] http://storm.apache.org/

[6] Rajiv Ranjan, "Streaming Big Data Processing in Datacenter Clouds", IEEE Cloud Computing, vol.1, no. 1, pp. 78-83, May 2014, doi:10.1109/MCC.2014.22

[7] Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Karthik Ramasamy, Jignesh M. Patel, Sanjeev Kulkarni, Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, Nikunj Bhagat, Sailesh Mittal, and Dmitriy Ryaboy. 2014. Storm@twitter. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD '14). ACM, New York, NY, USA, 147-156. DOI=http://dx.doi.org/10.1145/2588555.2595641

[8] Beckmann, Norbert, et al. The R*-tree: an efficient and robust access method for points and rectangles. Vol. 19. No. 2. ACM, 1990.

[9] http://www.vioozer.com/

[10] http://redis.io/

ArcGIS scripting

Generating unique hydrogeological maps

Réka Pogácsás
Department of Geoinformatics
Geological and Geophysical Institute of Hungary
Budapest, Hungary
pogacsas.reka@mfgi.hu

***Abstract*— The Department of Geoinformatics is responsible - among others - for designing task-specific applications and programs for other departments at the Geological and Geophysical Institute of Hungary. A good example for the cooperation is our new script for a project at the Department of Hydrogeology. This department maintains an up to date cadaster of all drilled wells in Hungary. The cadastral information system consists of the document sheets of each object which contains the metadata and basic information (for example: unique id, coordinates, depth and location) and two additional maps involving detailed information about the environment of the objects. The aim of the script was to create a multifunctional application (within ArcMap) which can store the input data and generates the required map and output document automatically. One of the challenges during the scripting phase was to compose this output document according to the strict statutory regulation. For the automation of the map generation further modules were needed, such as PIL – Python Image Library. As structure and the content of the output document was strictly defined none of the pre-defined ArcGIS solutions could have been applied.**

Keywords*— *ESRI, Python, Arcpy, geotool

I. INTRODUCTION

The Geological and Geophysical Institute of Hungary has a wide range of projects and functions, including national tasks. In several cases this means that a great amount of data has to be processed, which can only be done through scripts right on time. At the Department of Geoinformatics the aim is to simplify the work of the colleagues. This tool was created for the Department of Hydrogeology where they use it for the national hydrogeological log-book. The method is fixed by the law, which means that during the development we needed to follow strict rules. According to a ministerial decree (KvVM 101/2007. (XII.23), this department is responsible for several functions including the issue of the Hydrogeological log-book [1]. This documentation is made for order and also as data report. The layout is also fixed (including the scale, size and content of the map, the used fonts and several other details). Every day there are more than 20 orders of hydrogeological log-books, and earlier all maps had been drawn by hand [1]. The exported page has to contain the selected map type of the well and its predefined attributes. On the one hand our aim was to automate this workflow as much as possible (including the output maps as well) but on the other hand our intention was also to create a spatial database. As a national Institute the

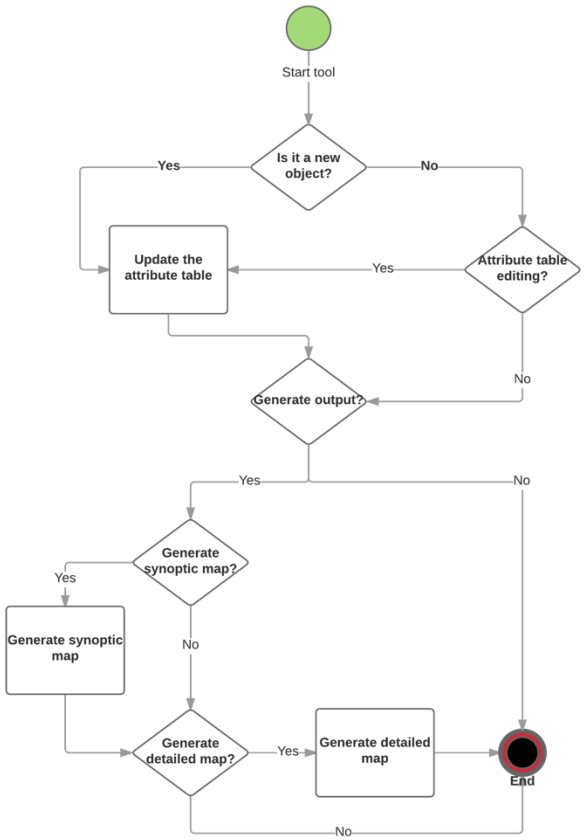


Fig. 1: The workflow of the script

usage of ArcGIS software was pre-defined therefore the tool was scripted according to the ArcGIS requirements.

IX. Method and workflow

To develop ArcGIS tools, python programming language is required, as well as the special python library called Arcpy [3]. The great advantage of the Arcpy module is to facilitate linking multiple tools and toolsets to each other as well as using the result data for further processes [2, 4]. The developed python script can be added as a tool to ArcGIS, and at the same time can include the same parameters as the built-in applications such as help file and validations [7]. After assessing the technical requirements and the needs of the Department we started to develop the tool. "Fig. 1." to make it more flexible our tool contains multiple choices because it has to work not only with newly registered but also with earlier recorded objects.

A. Creating the report file

First of all, we defined a method for the form. This was done by the report file editor of ArcMap, where the need for unique data can be indicated. The report was exported to an image file (.tif). The result of the map generation is also an image file format (.png). To merge the two outputs layers the script applies the Python Image Library (PIL). The process for this is to convert the output map (which file extension is png) to transparent background, and after resizing both images, frame them together.

B. Updating the attribute table

As written above there was a need to register the wells in a geospatial database. When the user starts the tool there is a form that needs to be filled in with basic data. To get the input information the "arcpy.GetParameterAsText" method is used in order to record the inputs as variables [8]. The user-given information contains among other things for example the coordinates of the well. Since the chosen geospatial database is a Personal Geodatabase by ESRI where the set of data is called feature class (the objects are called features), and all the features have attributes that are stored in the attribute table. The "arcpy.da.UpdateCursor" method is used to write the records into the database [2, 3].

```
...
query = "TelNev = '" + telepulesnev + "'" # 
defines the query with the telepulesnev input 
for the selection

      arcpy.SelectLayerByAttribute_management
(fc, "NEW_SELECTION", query) # new selection 
being done
                  fields = 
arcpy.ListFields(fc, "", "") # gets all fields 
to a list for the loop
                  rows = 
arcpy.SearchCursor(fc)
                  for row in rows:
                        for field in 
fields: # loops through all fields
                              if field.name
== "EOVY":
      arcpy.AddMessage("Koordinatak szamitasa
-  Telepules Y koordinataja")

      coord_telY = row.getValue(field.name) # 
sets coord_telY variable from 
Telepules_kozpontok.Tel_Y field
                                    return
coord_telY

   ...
```

As seen this is within the Arcpy module the "da" means data analysis. This refers to any basic operations on the dataset, and in this tool it is the most important method. With this method the geometry field of the object can also be updated, so the location is corrected. This is needed because the visual placement with the cursor might not be accurate. The wells are numbered by settlements. The newly registered well gets a registration identification from the current year and the next available number. Therefore "arcpy.da.UpdateCursor" is also applied to edit the well cadaster after searching the right city or town with "arcpy.da.SearchCursor". These methods iterate through the selected attribute table as a "for" loop. The document has to include the map sheet number where the well is located. The used grid reference system is the Hungarian civil edition of the Gauss Krüger system [5]. All the map sheets are merged into one raster file that is permanently added to the map document. Finding the right sheet is done by a spatial selection, called "arcpy.SelectLayerByLocation_management", where the map sheet number is recorded as a variable and written into the wells attributes. A function was defined for writing all the user defined attributes to the attribute table:

- Settlement name
- County name
- Address
- Cadastral number (In Hungary if this starts with 0 the object is outer of the settlement, in the attribute table this also had to be indicated, the script finds how the number starts, and updates the field)
- Special name
- Year of drilling
- Scale (needs to be defined for the detailed map)
- Outer or Inner of the Settlement
- Cadastral number of the well (number of wells at the settlement, at every new well of the selected settlement the number gets increased by the script)
- Registration number of the well (ordinal number of all registered wells at the year/current year)
- X coordinate of the location
- Y coordinate of the location (Both locations are given in Hungarian reference system, in which the unit is meter.

Two difficulties appeared. First in Hungarian comma is used instead of decimal point although Arcpy uses point, thus it has to be changes in a function. Additionally the Hungarian Geodetic Reference (EOV) switches the axes, therefore EOV Y is the X coordinate and EOV X is the Y [6]. This is also fixed by functions.

- Elevation
- Number of map-page

Furthermore we were asked to calculate the azimuth and distance between the object and settlement center, and set the record in the attribute table of the wells. Two functions were written for there, using simple trigonometric relations.

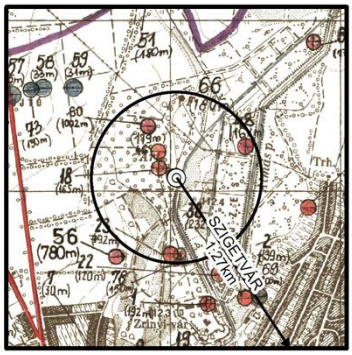


Fig. 2: The graphical objects on the synoptic site-plan

C. *Generating the site-plans (maps)*

The site plan (also mentioned as map) generation is the most complicated part of the script. Two types can be created, the detailed site-plan is simplier. The user defines a scale as an input. The script changes the symbolization of the selected well to the required appearance (the object has to be selected before running the script while this is the only way to change the symbolization of the well on the fly). An additional style layer (a .lyr file) is imported permanently. Than the script changes the view to layout (this is used for exporting maps to image files), sets the scale to the user-defined value, and moves the selected object to the center of the map before exporting it to a .png file ("arcpy.mapping.ExportToPNG"). The synoptic site-plan has graphical objects although Arcpy is not able to draw them, so they are generated as permanent feature classes. First the symbol of the well is being changed the same way as at the detailed ones. An arrow pointing to the center of the settlement where the well is also obligatory in case the center is not seen on the output map. While the cut-out size is 2000 x 2000 meter, the script measures the coordinate difference between the settlement center and the object. "Fig. 2." in case it is more than 1000 meters a line is drawn between them in a new permanent feature class. The arrow has to have signed with the settlement name and the distance, so the attributes has to be written to the attribute table. This also gets the styling from a pre-created style layer. Additionally a circle with a radius of 500 meters has to be drawn around the object as a buffer with transparent fill. After these elements the exportation works the same as with the detailed site-plan. The final step is merging the exportation to the report file as previously written above. Finally the tool was optimized, as the help file was added. To minimize typing mistakes at the validation tab of the tool we defined the list of the possible settlements and counties.

D. *Special case: well groups*

One expectation could not be completely solved; managing a group of wells. It is possible to register some wells as a groups, their attributes are almost identical only the coordinates differ. In this case the tool has to be used to fill the attribute table one by one object without generating an output file. After selecting every members of the group at the same time the hydrogeological-log can be exported. The site-plan is generated the same way as with single wells, however there are no circles around them, and the arrow is also unnecessary. The difficulty is caused by the report file, as the built-in functions of ArcMap do not allow by script to select which page (which wells data) of the report should be exported. It generates the report of all the wells in the group (if 10 objects are selected then 10 pages are generated) ordered by the sequence of selection. Thus in the tool the user can define a number referring to the order of the selection.

X. Results and possible further development

Some difficulties appeared during the development mostly due to ESRI environment. For example the creation of the graphical elements. These disadvantages caused that the scripts length reached 1200 lines, and has to use several permanent layer and style files. To step forward it would be important to integrate this system into the relational database (MsSQL) of the institute, and find the duplicated elements and make it possible to search for wells at the database. There is also a possibility to dissociate the filling of the attribute table and the map generation. All in all a useful tool for ArcMap was developed which according to the latest surveys could significantly increase the effectivity of the site-map generations workflow. Some security backups are also nested in the script and if needed numerous of error messages help the users in case of problems. "Fig. 3." the greatest advantage is that the use of the tool is simplified as much as possible therefore anyone with basic knowledge of ArcGIS can easily apply it.

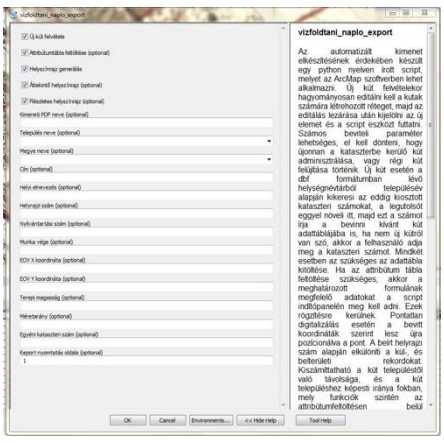


Fig. 3: The starting interface of the tool is simple

REFERENCES

[1] http://mfgi.hu/en/node/863 (2016. 05.)

[2] S. Toms, Arcpy and ArcGIS – Geospatial analysis with python. Birmingham, Packt Publishing, 2015.

[3] Z. Dobesova, "Programming language Python for data processing" in 2011 International Conference on Electrical and Control Engineering (ICECE), Yichang, 2011, pp. 4866-4869.

[4] H. Ali Sallah, "Development of a prototype ArcGIS- web-based decision application WATERDSS: For water pollution management" in 2015 7th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Bucharest, 2015, pp. E-1-E-6.

[5] I. Hazay, A. Tárczi Hornoch, A Gauss-Krüger koordináták számítása, Budapest: Akadémiai Kiadó, 1951.

[6] Sz. Mihály, A magyarországi geodéziai vonatkozási és vetületi rendszerek leíró katalógusa in Geodézia és Kartográfia, vol. 4. Budapest, 1994, pp. 198-203.

[7] Paul A. Zandbergen, Python Scripting for ArcGIS, Redlands, CA: ESRI Press, 2013.

[8] J. P. Rigol-Sancheza, N. Stuart, A. Pulido-Boscha, ArcGeomorphometry: A toolbox for geomorphometric characterisation of DEMs in the ArcGIS environment, Computers & Geosciences, vol. 5. part A, pp. 155-163.

Hyper spectral Image processing for Forest types Mapping and forest health monitoring: A case study in the buffer zones of iron mining belts of Saranda forest, Jharkhand, India

Narayan Kayet[1] Abhisek Chakrabarty[2]
[1]Research Scholar, Department of Remote Sensing & GIS, Vidyasagar University, Midnapore, INDIA
[2]Assistant Professor, Department of Remote Sensing & GIS, Vidyasagar University, Midnapore, INDIA

***Abstract*—Hyperspectral remote sensing is a very useful tool for forest health mapping, different forest types and also for natural resource management. In this work, the NASA-EO1 Hyperion sensor data is used for forest type and forest health monitoring for a case study in the Saranda forest of Jharkhand, India. Saranda forest is covered with different kinds of valuable trees and is rich in mineral deposits. Growing anthropogenic activities within and near the forest lands are causing a threat to the forest health and well-being. The forest area in the buffer zone of mining fields is under high-stress conditions may show signs of dry or dying plant material. Forest classification based on Hyperion data reveals the existing forest types, their overall health, and vigor in the forested region. Such designation helps to detect pest and blight conditions in forest particularly for assessing areas of timber harvesting and reforestation planning. Mining activities in the iron ore belt of the Saranda forest of Jharkhand in the Karo and Koina river basin have high potential to induce forestry health problem. Improper mining of minerals is often liable to damage the forests health. Forest regeneration activities introducing new species in the forest region can bring certain changes in the tree patterns over the years in a particular area in the forest.**

The study involves Hyperspectral EO-1data collection, pre-processing geometric and radiometric correction (bad band removal, cross track illumination correction, stripe removal and reduction of atmospheric and solar flux effects using FLAASH Correction) and generation of spectral signature from the image based on Minimum Noise Fraction (MNF) Pixel Purity Index (PPI) n-dimensional visualize and techniques. To distinguish the different forest type viz. deciduous forest, evergreen forest and mixed forest spectral classification Spectral Angle Mapper (SAM) technology are applied. This enables to compare the spectral signature of the forest with the USGS spectral library files using the software ENVI. Based on the study results final map of forest classification and their area statistics is prepared, which indicates the grades of forest health with acceptable accuracy. The results were considered separately with the USGS spectral library of forest types and forest health monitoring map. The output image and area statistics showed the capability of the method in remotely monitoring the health of vegetation and forest type.

***Keywords*— Hyperspectral Remote sensing, FLAASH Correction, Narrowband Vegetation indices MNF, PPI, n-dimensional visualizer, Spectral analysis, SAM, Forest type and health monitoring.**

XI. INTRODUCTION

The forest areas near the iron ore mining belts in Jharkhand are covered with different types of trees that are susceptible to the changed environs due to mining. The health of the trees of each species responds differently to various anthropogenic activities within or near the forest lands. The forest area in the buffer zone of mining fields, which is under high stress conditions, may show signs of drying or dying plants. The quality of the canopy can be expressed in terms of efficiency of using sunlight. Hyperion EO-1 sensors provide images the earth surface features in hundreds of narrow adjoining spectral bands. These can be used for earth surface features mapping. Maps generated from these data have a very high-ranking accuracy. These data are maintained in a spectral library as the reflectance spectra. Such conditions could be revealed through Hyperion data analysis. NASA's EO-1 satellite is equipped with hyperion spectrometer that has the sampling distance of 10 nm within the 7.7 km swath path which provide 242 spectral bands within the 400-2500nm (355.59-2577.08 nm, to be precise) wavelength of EM spectrum (EO-1/ Hyperion science data user's guide).Hence the Hyperion data obtained from the EO-1 provide scopes of precise classification of land covers. Thus there is immense scope of using this for forest zoning and classification based on tree species, agricultural crops classification, identification of forest biochemical properties, identification of chlorophyll content and plant nitrogen content, etc (Thenkabail, P.S et al, 2001).Once such classification based on Hyperion data is carried out, it is possible to generate spatial map that shows the overall health and vigor of a forested region. It could be possible to detect pest and blight conditions in a forest as well as assessing areas for timber harvesting (Tuominen, J et al, 2009).The study aims to express the quality parameters of standing vegetation health in terms of hyperspectral signatures. Such signature will provide the quality of vegetation in terms of presence of bio-chemicals, plant water content etc. Decision tree classifier tool of ENVI will be used to classify the pixels of a Hyperion image for necessary information acquisition for forest management purposes (EO-1 Hyperion Vegetation Indices Tutorial ENVI 5.3).Forest health is characterized by leave and canopy qualities, forest type, bio chemical characteristics of the plants, species diversities, fire proneness, deforestation ratio, bio-physical conditions etc. Recent developments in hyperspectral data acquisition from satellite borne spectrometers have opened new areas of research that could bring revolutionary changes in the current approaches of forest management (Franklin, S. E. 2001).Number of researchers have assessed vegetation stress from a derivative chlorophyll index and leaf area index estimation using compact airborne spectrographic image (Wu, C., Han, X., Niu, Z., & Dong, J. 2010). Detection of vegetation stress by remote sensing techniques is based on the assumption

that vegetation stress factors interfere with photosynthesis or the physical structure of the vegetation and affect the absorption of light energy and thus alter the reflectance spectrum of vegetation (Zarco-Tejada, et al. 2009). Cho, M.A et al. 2008 used airborne Hyperspectral data to develop photochemical reflectance index (PRI) for distinguishing the species wise variations of leafs. Thus using airborne hyperspectral data it is possible to estimate distribution of different species in the forest. The false color aerial photographs and multispectral sensor based satellites image provide red edge related observables (Mc Carton et al, 1997). By the red edge, the different between the reflectance maximum at NIR and similar to minimum at visible red edge involved indices like as normalized different vegetation index (NDVI) and leaf area index (LAI) were applied. International Institute for Geo-Information Science and Earth Observation (ITC) study on tree species classification using hyperspectral data has been on the pixel-level tree classification, classification map at the separate tree near can be directly connected to the tree's biophysical properties and it can be used for applied forest management determinations (Vauhkonen, J, et.al. 2011) and higher classification accuracies have been described likened to pixel level (Dalponte et al., 2013).

The study aims to express regarding spectral signature generate the various quality parameters of standing vegetation health and forest type classification. Such signature will provide the quality of vegetation in terms. This work to assist the mine managers to carry out environmentally friendly mining and help the department of forests to administer the forest conservation and development through information based monitoring of action plans. This proposed methodology is applied to Saranda forest state of Jharkhand, INDIA.

XII. STUDY AREA

The present study is the Saranda forest, and its surrounding area is located the West Singhbhum district of the Indian state of Jharkhand (Figure 1). It is famous for Asia's largest Sal forests and is an important elephant habitat. The Saranda forest of Jharkhand is endowed with some rich iron ore deposits. The location of the forest is within latitude 22° 3' 7.98" - 22° 14' 0.67"N and longitude 85° 21' 31.52" - 85°25'53.18"Ewith elevation of 850m above the mean sea level (MSL).Saranda forest is fed by two major rivers, Karo and Koina. The catchment of these rivers comprises of a drainage system with stream order up to six. Over the last few decades, in this region, many iron ore mining towns have emerged, e.g. Gua, Chiria, Megataburu and Kiriburu.

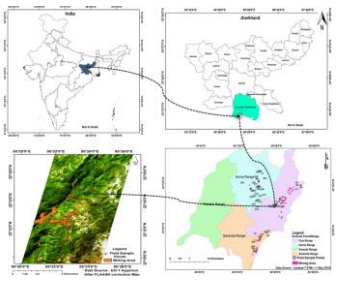


Figure 1: Location map of the study area

XIII. MATERIAL AND METHODS:

A. *Data used:*

Precise detection, identification, and assessment of forest health monitor and forest type mapping in around the Gua mines, Saranda forest, Jharkhand were attempted by using hyperion EO-1 data (2006) & Top sheet (2005). Open file EO-1hyperion data from NASA and District Resource Maps of Survey of India were utilized for this purpose. GPS Survey and photographs are taken for collecting the ground truth data using the field validation.

B. *Data pre-processing:*

Hyperspectral data poses challenges to image interpretation, because of high data volume, redundancy in information, the need for calibration, and dimensionality problem. The raw Hyperion image is having individual errors arising due to atmosphere or malfunctioning of sensor mechanism. The errors are said to be caused due to calibration differences in the detector array (Goodenough, D. Get. al., 2003).

Bad band removal: The delivered USGS Hyperion product level LIR has 242 bands, of which 44 were not calibrated. The main reason for not calibrating the entire band was the decreased sensitivity of the detectors within the non-calibrated spectral regions. Out of the 242 collected bands, bands 1-7(356-417nm) and 225-242(2406-2578nm) were not calibrated (Christian, B., & Krishnayya, N. S. R., 2007).58-78 to overlap region, 77 and 78 were eliminated because of the higher noise level present in those bands. There are water vapor absorption bands which need to be eliminated and are identified as bands 120 -132,165-182 and bands 221-224. (Sahoo & Pradhan,).Hyperion has 242 Spectral bands ranging from 355.59-2577.08 nm. Due to poor signal to noise from some of these bands, only 198 were processed (Datt, B. et al., 2003).

Geometric correction: Geometric distortion is often to be removed from remotely sensed data. There are two main approaches to removed geometric disorder from remotely sensed data. One is a systematic approach, and another is the non-systematic approach. Some of these errors can be corrected by using ephemeris of the platform and previously known internal sensor distortion characteristics (Homer, C. et al. 2001). Other errors can be rectified by matching image coordinates of physical feature recorded by the image to the geographic coordinates of the same feature collected from a map or using global positioning system (GPS).

Radiometric correction: Cross track Illumination correction: Cross track Illumination correction is used to remove variation in the cross track illumination of the image. Cross track Illumination change may be used to vegetating effects, instrument effects or scanning or other non-uniform illumination effects. This algorithm calculates mean values of an extended track Polynomial function, and fit mean values remove this error. (San, B. T. et al. 2011).

De-stripping: Hyperion L1R data show a severe striping effect by imprecise co-calibration of individual detectors on the focal plane array. The first 12 Visible near Infrared bands and so many

Short Wave Infrared bands are affected by striping and bad columns. It is evident that an uncorrected striping effect will lead to faulty interpretation of the result of the data (Rasel, Sikdar MM, et al 2015). Vertical stripe and replacing the DN value of the affected column by the average of DN values of the adjacent columns (George, R., et al 2014). Hyperspectral data are affected by different noise sources which can be grouped into two main classes: random noise and fixed pattern noise. Photon noise (also called shot noise) and thermal noise are two examples of random noise in Hyperspectral images while striping noise is a typical example of fixed pattern noise, often associated with push-broom sensors (Mora, A., & Nelson, P. 2014).

FLAASH atmospheric correction: Atmospheric correction is only reducing the effects of the atmospheric components (water vapor, dust, gasses, etc.) on the electromagnetic radiation reflected or emitted from the surface. Convert surface reflectance of Hyperion data uses the Fast line of sight atmospheric analysis of spectral hypercube algorithm, available in Envi software. This algorithm developed by Spectral Sciences, Inc, under the sponsorship of the US Air force Research Lab. The calibration model based on the theory of atmospheric radiation is according to the physical process of radiation transfer, building by radiation transfer equation, using a theoretical formula to proceed with atmospheric correction (Ganesh, B.P, et al 2013).FLAASH starts from a standard equation for spectral radiance at a sensor pixel, L; that applies to the solar wavelength range (thermal emission is neglected) and flat, Lambert a material or their equivalents. The equation is as follows:

$$L = \left(\frac{Ap}{1-peS}\right) + \left(\frac{Bpe}{1-pes}\right) + La \qquad (1)$$

$$Le = \left(\frac{(A+B)\,pe}{1-peS}\right) + La$$

Where: P is the pixel surface reflectance, Pe is an average surface reflectance for the pixel and a surrounding region ,S is the spherical albedo of the atmosphere, La is radiance backscattered by the atmosphere, A and B are coefficients that depend on atmospheric and geometric condition but not on the surface, The first equation corresponds to radiance that is reflected from the surface and the second equations are to radiance from the surface that is scattered by the atmosphere into the sensor (Gao, B. C ,et al 2009).The radiance spectra of Hyperion L1R dataset some grasses and absorption features are present. The absorptions can be visualized in the spectral region from 400-2500 nm. There are two gasses and water is absorbed in vegetation radiance spectra. At 760nm, oxygen absorption gasses are present. Carbon-di-oxide is present 1900-2055nm. Water vapor absorption is 940-1140nm and 1375-1900nm. This absorption gasses and water vapor are the largest effects on the radiance data. FLAASH atmospheric correction removes this absorption feature. FLAASH MODTRAN has outperformed other radiative transfer code especially in the water region 940 and 1130nm and CO2 at 2055nm (Matthew, M. W.et al., 2002). Aerosol information is required to make accurate atmospheric correction of Hyperspectral data. Over the ocean, aerosol optical depths and aerosol models can be retrieved from measured data at 0.86μm and longer wavelengths where water surfaces are usually dark (Yuan, J., & Niu, Z. 2008).There is certain input needs to be generated for FLAASH atmospheric correction models in ENVI. In the case of vegetation, the signature of the radiance in the VNIR and SWIR region is dominated by the high atmospheric absorption region. After FLAASH atmospheric correction the vegetation spectra was absorbed within the wavelength between 1900-2100nm because considering the CO2 Mixing Ratio is 390ppm (Figure 2).

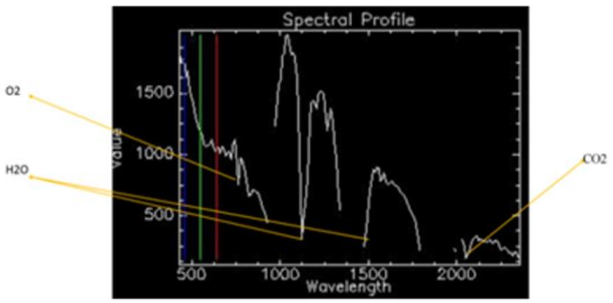


Figure 2: O2, H2O, CO2 absorption gases in radiance spectra

The difference in the positions of water vapor absorption could be attributed to the process of treating the water vapor models. Water vapor depends on seasonal latitude and temperature model. The column water vapor amount for each pixel in the image must be determined.

C. Forest health Mapping Techniques

The forest health vegetation analysis tool will generate a spatial map that shows the overall health and vigor of a forested area. It is good at detecting pest and blight conditions in a forest. The tool uses the following vegetation index categories:

1. Broadband and narrowband greenness, to show the distribution of green.
2. Leaf pigments, to show the concentration of arytenoids and anthocyanin pigments for stress levels.
3. Canopy water content, to show the concentration of water.
4. Light use efficiency, to show forest growth rate.

Narrowband Vegetation indices:

In the forest health discrimination work vegetation indices are usually extracted from the data. Vegetation indices are usually combination of surface reflectance at two or more wavelength design to highlight a particular property of vegetation. Each of the vegetation property is design to represent a particular vegetation property (Table-1).

Narrowband index	Table -1 Narrowband vegetation Index		
	Index	*Algorithm*	*Reference*
Greenness	(i)Modified Red Edge Normalized Difference Vegetation Index (ii)Vogelmann Red Edge Index 1	$mNDVI705 = (750nm - 705nm) / (750nm + 705nm - (2 * 445nm))$ $VREI1 = 740nm / 720nm$	(i)Datt,(1999) (ii)Zarco-Tejada et al., (2001)
Light Use Efficiency	Structure Insensitive Pigment Index	$SIPI = (800nm - 445nm) / (800nm - 680nm)$	Penuelas et al.,(1995)
Leaf Pigments	(i)Carotenoid Reflectance Index 1 (ii)Anthocyanin Reflectance Index 1	$ARI1 = \frac{1}{550nm} - \frac{1}{700nm}$ $CRI2 = (1/510nm)(1/550nm)$	(i)Gitelson et al. (2002) (ii)Gitelson et al.(2001)
Canopy water content	Normalized Difference Infrared Index	$NDII = 819nm - 1649nm / 819nm - 1649nm$	Hardisky et al. (1983)

Greenness Index: Greenness vegetation indices are generally measure the vigor and green vegetation. They measure the various aspects such as: chlorophyll concentration, canopy area and canopy structure. Greenness vegetation indices are based on the measuring the reflectance peak in NIR region. Red wavelengths where the chlorophyll absorption is strongest are used as a reference.

A. Modified Red Edge Normalized Difference Vegetation (mNDVI705): Modified normalized difference vegetation index is modified mNDVI705. It is differing from modified nNDVI705 by incorporating a correction of the leaf spectral reflectance. It is sensitivity to vegetation red edge and also small changes to foliage content, gap friction and senescence (Datt 1999). It is applied in agriculture; forest monitoring and vegetation stress detection. The equation is given below:

$$mNDVI705=(750nm\ 705nm)/(750nm + 705nm\ (2 * 445nm)) \quad (2)$$

The values of mNDVI705 index range since 1 to 1, with the common range for green vegetation falling between 0.2 and 0.7.

Vogelmann Red Edge Index 1 (VREI1):

Vogelmann red edge index 1 is a narrowband reflectance measurement. This index is sensitive to combined effects of leafage chlorophyll attentiveness, water body content and canopy leaf zone (Zarco-Tejada et al., 2001). VREI1 index application on vegetation growth, candidness of agriculture, vegetation output modeling. The equation is given below:

$$VREI1 = 740nm / 720nm \quad (3)$$

The value of VREI1 index range from 0 to 20. With common range for green vegetation falling between is 4 to 8.

B. Light Use Efficiency Index:

Light use efficiency vegetation indices providing the efficiency of which vegetation is able to use incident light for photosynthesis. It is correlates with carbon uptake efficiency and growth rate. However, by the use of light use efficiency vegetation indices measure the growth rate and production of vegetation.

Structure Insensitive Pigment Index (SIPI):

Structure insensitive pigment index is a reflectance measurement unit where maximize the sensitivity to the ratio of bulk carotenoids to chlorophyll while minimizing the effect of variation in the canopy structure. SIPI use three bands blue, NIR and maximum chlorophyll peak at 680nm (Penuelas et al. (1995). SIPI is defined by the following equation:

$$SIPI = (800nm - 445nm) / (800nm - 680nm) \quad (4)$$

This index value of range from 0 to 2, healthy green vegetation is from 0.8 to 1.8.

C. Leaf Pigments Index

Leaf pigment vegetation indices are measured the amount of stress related pigment in the vegetation. In the stress vegetation there are higher concentration of carotenoids and anthocyanins. Carotenoids are the pigment that prevents the vegetation from high light condition. Anthocyanin pigment contents are high in new leafs.

Carotenoid Reflectance Index 1 (CRI1):

Carotenoids reflectance index measure the amount of carotenoids instead in the canopy. It is calculating by difference of two bands that is sensitive to carotenoids amount (Gitelson et al. 2002). It is defined according to the equation:

$$CRI2 = (1/510nm)(1/550nm) \quad (5)$$

The value of this index ranges from 0 to more than 15. The common range for green vegetation is 1 to 12.

Anthocyanin Reflectance Index1 (ARI1): Anthocyanin reflectance index is sensitive to anthocyanin amount in the vegetation. Weakening vegetation contains higher concentrations of anthocyanins, so this index is one measure of stressed vegetation. (Gitelson et al.2001). ARI1 is defined according to the equation:

$$ARI1 = \frac{1}{550nm} - \frac{1}{700nm} \quad (6)$$

D.Canopy water content Index

Water content vegetation indices are design to estimate the canopy water content. However, water content is an important vegetation property that controls the vegetation growth and also

correlates with vegetation health. The use of water content vegetation indices needed high spectral resolution data.

Normalized Difference Infrared Index (NDII):

NDII index is sensitive to change canopy water content of plant canopies. It used two difference water absorption bands it is 857nm and 1241nm (Hardisky et al. (1983)).This index application on crop, agricultural, forest canopy, and vegetation stress detection. NDII is defined by the following equation:

$$NDII = 819nm–1649nm / 819nm–1649nm \quad (7)$$

This index value of range from -1 to 1. The common range for green vegetation is 0.02 to 0.6.

D. Hyperspectral analysis

The following procedures of Hyperspectral analysis were employed, including the Minimum Noise Fraction (MNF) transformation for reducing spectral data, the Pixel Purity Index (PPI) for identifying those extreme or spectrally pure pixels, and the n Dimensional Visualizer for determining the endmember directly from the image. Spectral Angle Mapper (SAM) was applied to estimate abundances of each endmember to produce final map.

Minimum Noise Fraction (MNF):

Minimum noise fraction (MNF) can be reduced the inherent dimensionality of the dataset and reduce noise from the dataset. MNF also reduce the computational requirement for subsequent processing (Denghui, Z., & Le, Y. (2011). MNF also decor relates and rescales the noise in the dataset. The first step in MNF has transformed the data in which noise has unit variance, and there is also no band to band interrelationship. Secondly, MNF can compute/process the principle component analysis for noise whitened data. The MNF transformed two cascaded principle components analysis (Chen, C. M et al., 2003). MNF inversion produces much smaller spectral angles than those derived in transformed space (Mundt et al., 2005). The first ten inverse MNF bands contain 95% of the total information.

Pixel Purity Index (PPI):

Image transformation techniques typically used statistical analysis and reduce the dimensionality of the data one of the transformation is called Principle component analysis or principal component transformation. The Pixel Purity Index (PPI) is a means of finding the most “spectrally pure,” or extreme, pixels in Multispectral and Hyperspectral images (Chang, C. I., & Plaza, A. 2006).The most spectrally pure pixels typically correspond to mixing endmembers.

n-dimensional visualizer:

N-Dimensional Visualization is applied after correcting the image. Previously gather the spectrally pure algorithms using PPI or MNF transformation. Where ‘N’ is the number of bands, the coordinates of the bands in ‘N’ space consisting of ‘N’ values that are simply the spectral radiance and reflectance values in each band of a given spectrum and reflectance values in each band of a given pixels. The distribution of bands in ‘N ‘space can be used for estimation of some spectral endmembers and their pure spectral signature. The ‘N’ dimensional visualization is applied after gathering the data though MNF or PPI algorithms. The pre-clustering result is attempting to find in the corner in spectral scatter plot. Since the purest pixel is to find in the neighborhood of the data cloud. The n-Dimensional visualizer allows for interactive rotation of data in n-D space, selection of group into different classes (Boardman& Kruse, 1999).

Spectral ClassificationTechniques:

Classification and feature extraction method have been commonly used for many years for the mapping of forest type and vegetative cover of the multispectral dataset. However, conventional classification algorithm cannot be applied to hyperspectral data due to the high dimensionality of the data. Several mapping techniques were used in the present study to map the abundance found in the study area.

Spectral Angle Mapping (SAM):

Spectral Angle Mapper is an algorithm, which is widely used for Hyperspectral image correction. Spectral Angle Mapper is a supervised image correction process. A pixel with minimum spectral angle comparison with reference spectra is assigned to the pixel vector. The algorithm determines the spectral similarity between two spectra by calculating the angle between the spectra and treating them as vectors in space with dimensionality equal to the number of bands (Kurse et al., 1993). A pixel with minimum spectral angle is a comparison to the real pixel is considering actual spectra of image spectrum. Spectral Angle Mapper is a statistical comparison method which is done through pixel Angle analysis. Spectral Angle Mapper is always not predictable where there is an image overlapped problem. Spectral Angle Mapper is based on the similarity between two spectra. Spectral Angle Mapper can calculate the angle between two spectra through algorithms testing. The spectral similarity between the image spectrum (t) and reference spectrum (r) can be expressed as an average angle (ø) between the two spectra for each channel (Van Der Meer, F. 2006) (Figure 3).

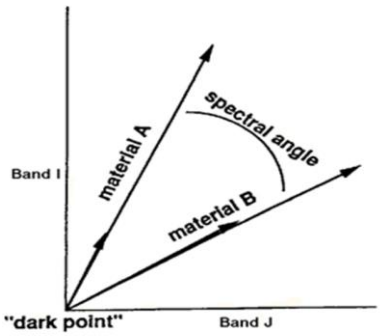


Figure 3: Representation of Reference Angle

The SAM algorithm generalizes this geometric interpretation to nb -dimensional space. SAM determines the similarity of an unknown spectrum t to a reference spectrum r , by applying the following equation (CSES, 1992):

$$a = \cos^{-1}\left(\vec{t} * \vec{r} \,/\, \|\vec{t}\| * \|\vec{r}\|\right)$$

$$a = \cos^{-1}\left(\sum_{i=1}^{nb} t_i r_i \,/\, \left(\sqrt{\sum_{i=1}^{nb} ti^2}\right)^{1/2}\left(\sqrt{\sum_{i=1}^{nb} ri^2}\right)^{1/2}\right) \quad \textbf{(8)}$$

Where, t = Unknown spectra; r =Reflectance spectra; nb = number of bands. Overall integration methodology is given in Figure 4.

Accuracy Assessment:

Accuracy assessment is an important work in classification validation system. Remote Sensing technology is a great source of thematic map presentation system; although accuracy assessment assists how far the classification represents the real world. There is some method to conduct accuracy assessment 1) confidence building assessment; 2) model based interface; 3) design based interface. Accuracy assessment can work some sampling method-1.Random sampling; 2 Systematic sampling; 3.Stratified random sampling; 4.Stratified systematic unaligned sampling; 5.Cluster sampling. Accuracy assessment can produce user accuracy, producer accuracy, total accuracy and also kappa coefficient value. The producer accuracy of the classified pixels compared to ground truth (Congalton, 1991) by the use of Eq.

$$\hat{K} = \frac{N\sum_{i=1}^{k} xii - \sum_{i=1}^{k}(x_{i+} \times x_{+i})}{N^2 - \sum_{i=1}^{k}(x_{i+} \times x_{+i})} \quad \textbf{(9)}$$

In which; r = number of rows in the error matrix, xii= number of observations in the ith row and column, x+i= total number of observations in the ith column, and N = total number of observations. The overall research methodology is shown in Figure 4.

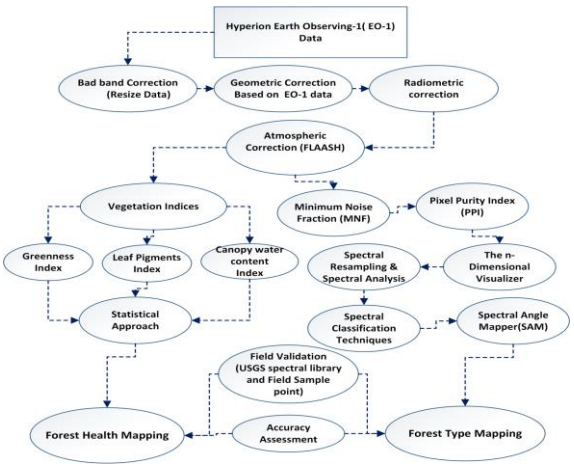


Figure 4: Overall methodology

XIV. RESULTS AND DISCUSSION

A.Result of Atmospheric FLAASH correction

FLAASH is an effective atmospheric correction process where all bands are corrected to follow their proper geometric and radiometric characteristics. FLAASH can also remove the de striping of the image, path radiation of the image, and various systematic and non-systemic effects. Pre FLAASH correction and FLAASH correction statistics value are shown in Table 2. Image and Spectral variability of the Before FLASSH Correction and after FLASSH correction are shown in figure 5.

FLAASH correction	Table -2. Minimum, Maximum, Mean, and standard deviation values for pre and FLAASH correction			
	Min.	Max.	Mean	Stdev.
Pre FLAASH correction	-774	338325	84444.44	66423.37
FLAASH correction	0	70.8979	11.6372	10.21467

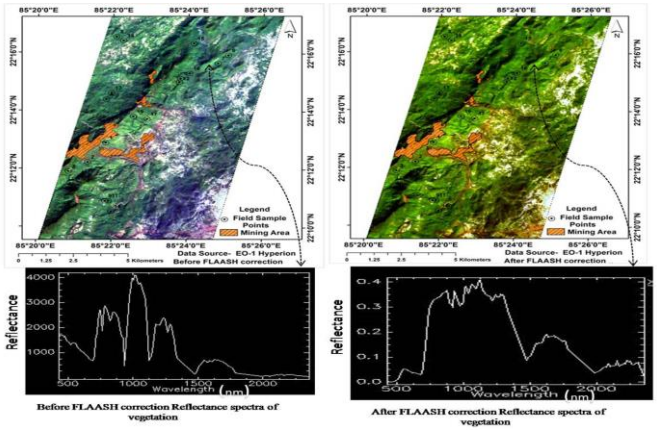


Figure 5: Image and spectra before and after atmospheric correction

B.Result of Forest Health

Vegetation Indies (VIs) were calculated for all 16 test sample pixels. Mean and standard deviation values for both healthy and unhealthy classes are shown in table 3. Separability 'S 'between healthy and unhealthy classes was calculated using the formula (Landgrebe, 2003).

$$S = \frac{u_1 - u_2}{\sigma_1 + \sigma_2} \quad \textbf{(10)}$$

Where Mu and sigma are the mean and standard deviation of the particular class respectively.

Vegetation Index	Table -3. Mean, standard deviation and separability values for each tested				
	Healthy		Unhealthy		
	Mean	Std.	Mean	Std.	S
MNDVI 705	0.66968	0.03259	0.17246	0.08544	4.2124
VREI1	1.44556	1.13552	0.06033	0.05853	2.6083
CRI1	33.22221	7.92681	17.17751	2.24063	1.5780
ARI1	9.65035	4.78392	7.71285	2.29369	0.2737
NDII	-0.49442	0.07740	-1.23946	0.27447	2.1173
SIPI	1.09096	0.02121	0.51795	0.33690	1.6000

In general separability values obtained for greenness vegetation Indies were relatively high. Highest separability values were obtained for the Modified red edge normalized difference vegetation (MNDVI705) index. MNDVI705 correlates well with reduced chlorophyll content, so its good performance could be expected the result is understandable. The test result of Canopy water content index were rather low because canopy structure low. This can clearly be seen from figure 6 where the healthy and unhealthy values are shown for each class.

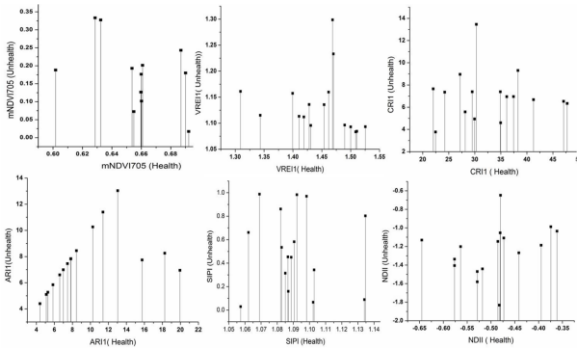


Figure 6: Vegetation Indices (VIs) as well as Health and Unhealthy value are show for each VIs class.

Vegetation health classification was done by the use of three narrow-band vegetation indices which is MNDVI705 (Greenness or Chlorophyll Vegetation Index) CRI_1 (Leaf Pigment Vegetation Index) and SIPI (Light Efficiency Vegetation Index). ENVI software provides nine forest health classes (Figure 7), and the spectral signature of each forest health classes is shown in figure 8.

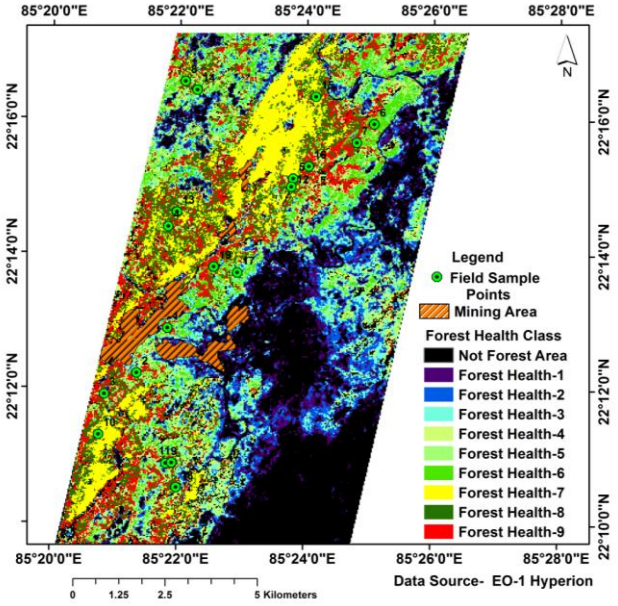


Figure 7: Forest Health Map (Forest health class 9 very Health and 1 class very low health)

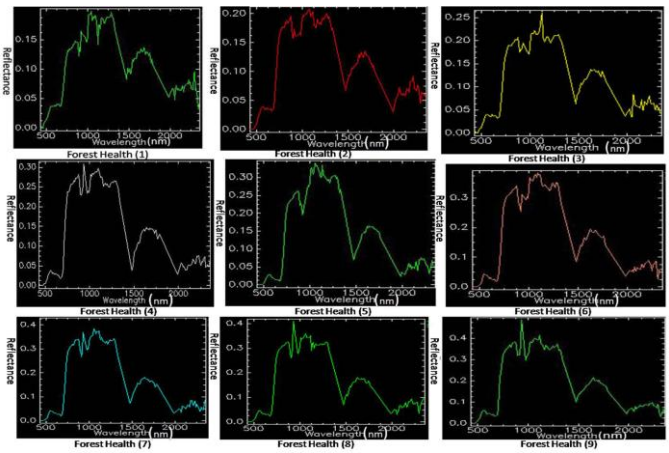


Figure 8: Reflectance spectra of forest health 1 to 9

Distribution of Forest Health:

Most of the area of the study area is covered by strong forest class 9 (2749.38 Acre). It is distributed upper the Gua mines and northeast region of the study area. Healthy forest 1 level area allocated the low portion of the study area and the rest area covered by unhealthy forest (2600.98 Acre), it is distributed basically around the Gua mine area, agricultural area, scarps area, city and mining area and fellow land area (Table 4 and Figure 9).

Forest Health classes	Table -4.Forest health classes area distribution summary
	Area(Acres)
Forest Health-1	2,600.98
Forest Health-2	2,603.20
Forest Health-3	2,603.65
Forest Health-4	2,173.78
Forest Health-5	2,108.02
Forest Health-6	2,105.80
Forest Health-7	2,657.63
Forest Health-8	2,658.96
Forest Health-9	2,749.38

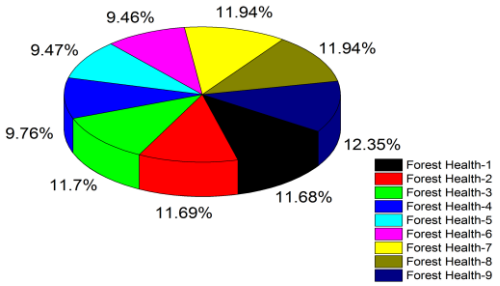


Figure 9: Pie Diagram of Forest Health classes

C.Result of Forest type

Forest type is mainly dependent on various physiological parameters such as climate, temperature, geology, soil, slope, aspect, hill shade direction and much more. The study area is mainly covered by three forest types; which are deciduous, tropical evergreen forest and tropical dry forest. Forest type classification using SAM technique. Spectral signature (Figure 10) collected from the image and matches the spectral signature from USGS spectral library and field sample data in ENVI environment; different dominant forest type has been mapped. Classify the three type of forest that deciduous and tropical evergreen forest and tropical dry forest (Figure11).

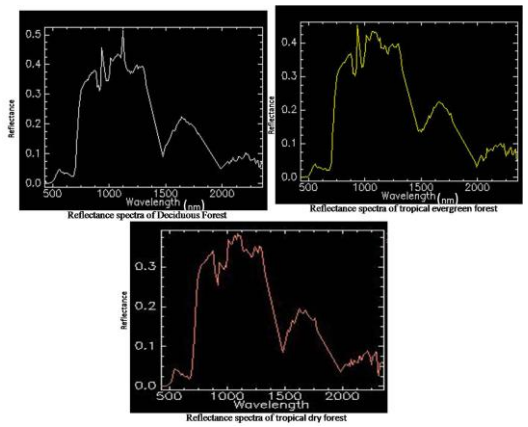


Figure-10- Reflectance spectra of deciduous, tropical evergreen forest and tropical dry forest

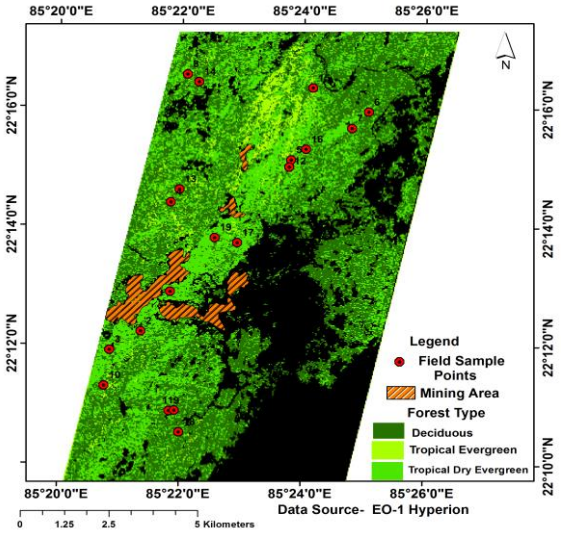


Figure-11 Forest Type Map

Distribution of Forest Type:

The deciduous forest covers 69.859% study area, 27.256% area is covered by tropical evergreen forest, and tropical dry evergreen forest covers 2.885% area. Deciduous forest is the main forest of the study area; tropical evergreen forest are shown mostly north, and north-east direction of the study area and tropical dry evergreen forest are situated within the deciduous forest shown north and north-east direction of the study area.

Accuracy Assessment:

Classification accuracy was done by spectral angle mapper (SAM) classification technique in ENVI environment. SAM was implemented to the collect of the spectral signature of deciduous forest, tropical evergreen forest, and tropical dry forest. Forest type classification is validated by the USGS spectral library and sample field point. Thus, overall accuracy 89 % and 0.8098 kappa coefficient were calculated.

CONCLUSIONS

The article has summarized forest ecosystem including forest type and forest health discriminator. The case study can demonstrate in hyperspectral data. Hyperspectral data has more capability than multispectral data i) classify Forest Type and health more accurately than multispectral data. ii) Estimate biophysical and biochemical properties in higher accuracies. Saranda forest has covered a lot of areas; the study area is covered about 85% forest cover. The forest information products from hyperspectral dataset improve forest inventory.

The study has although a limited of data set unavailability of the spectrometer for canopy leaf pigment measurement and allowed an evaluation of effects of reflectance measurement of individual leafs on top of the canopy of the hyperspectral indices. At the conclusion of the study is that spectral indices are sensitive to the change in the scale of spectral coverage and spectral measurement of the leaf of the canopy. However, the canopy indices are a better discriminator than leaf indices. The carotenoid reflectance measurement gives the almost accurate result. Hyperspectral indices provide more potential and might have new possibilities of different plant species and communities.

ACKNOWLEDGEMENTS

The authors are thankful to SAIL and DFO of Saranda forest for their financial support and providing necessary data. The authors would like to thank Indian Institute of Technology Kharagpur and Vidyasagar University for its constant support and providing wonderful platform for research.

REFERENCES

[1] Adams, J. B., Sabol, D. E., Kapos, V., Almeida Filho, R., Roberts, D. A., Smith, M. O., & Gillespie, A. R. (1995). Classification of multispectral images based on fractions of endmembers: Application to land-cover change in the Brazilian Amazon. Remote sensing of Environment, 52(2), 137-154.

[2] Asner, G. P. (1998). Biophysical and biochemical sources of variability in canopy reflectance. Remote sensing of Environment, 64(3), 234-253.

[3] Asner, G. P., Anderson, C. B., Martin, R. E., Knapp, D. E., Tupayachi, R., Sinca, F., &Malhi, Y. (2014). Landscape-scale changes in forest structure and functional traits along an Andes-to-Amazon elevation gradient.Biogeosciences, 11(3), 843-856.

[4] Baldeck, C. A., Colgan, M. S., Féret, J. B., Levick, S. R., Martin, R. E., &Asner, G. P. (2014). Landscape- scale variation in plant community composition of an African savanna from airborne species mapping.Ecological Applications, 24(1), 84-93.

[5] Baumann, M., Ozdogan, M., Kuemmerle, T., Wendland, K. J., Esipova, E., &Radeloff, V. C. (2012). Using the Landsat record to detect forest-cover changes during and after the collapse of the Soviet Union in the temperate zone of European Russia. Remote Sensing of Environment, 124, 174-184.

[6] Card, D. H., Peterson, D. L., Matson, P. A., & Aber, J. D. (1988). Prediction of leaf chemistry by the use of visible and near infrared reflectance spectroscopy. Remote Sensing of Environment, 26(2), 123-147.

[7] Chan, J. C. W., &Paelinckx, D. (2008). Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. Remote Sensing of Environment, 112(6), 2999-3011.

[8] Chang, C. I., & Plaza, A. (2006). A fast iterative algorithm for implementation of pixel purity index. Geoscience and Remote Sensing Letters, IEEE, 3(1), 63-67.

[9] Chen, C. M., Hepner, G. F., & Forster, R. R. (2003). Fusion of hyperspectral and radar data using the IHS transformation to enhance urban surface features. ISPRS Journal of photogrammetry and Remote Sensing, 58(1), 19-30.

[10] Chen, Q., Laurin, G. V., &Valentini, R. (2015). Uncertainty of remotely sensed aboveground biomass over an African tropical forest: Propagating errors from trees to plots to pixels. Remote Sensing of Environment, 160, 134-143.

[11] Cho, M. A., Sobhan, I., Skidmore, A. K., & de Leeuw, J. (2008). Discriminating species using hyperspectral indices at leaf and canopy scales. The International Archives of the Spatial Information Sciences, 369-376.

[12] Christian, B., & Krishnayya, N. S. R. (2007). Spectral signatures of teak (Tectona grandis L.) using hyperspectral (EO-1) data. CURRENT SCIENCE-BANGALORE-, 93(9), 1291.

[13] Clark, M. L., Roberts, D. A., Ewel, J. J., & Clark, D. B. (2011). Estimation of tropical rain forest aboveground biomass with small-footprint lidar and hyperspectral sensors. Remote Sensing of Environment, 115(11), 2931-2942.

[14] Dalponte, M., Orka, H. O., Gobakken, T., Gianelle, D., &Næsset, E. (2013). Tree species classification in boreal forests with hyperspectraldata.Geoscience and Remote Sensing, IEEE Transactions on, 51(5), 2632-2645.

[15] Datt, B., McVicar, T. R., Van Niel, T. G., Jupp, D. L., & Pearlman, J. S. (2003). Preprocessing EO-1 Hyperion hyperspectral data to support the application of agricultural indexes. Geoscience and Remote Sensing, IEEE Transactions on, 41(6), 1246-1259.

[16] De Carvalho, O. A., & Meneses, P. R. (2000, February). Spectral correlation mapper (SCM): an improvement on the spectral angle mapper (SAM). InSummaries of the 9th JPL Airborne Earth Science Workshop, JPL Publication 00-18 (Vol. 9). Pasadena, CA: JPL Publication

[17] Denghui, Z., & Le, Y. (2011, September). Support vector machine based classification for hyperspectral remote sensing images after minimum noise fraction rotation transformation. In Internet Computing & Information Services (ICICIS), 2011 International Conference on (pp. 132-135). IEEE.

[18] Elmore, A. J., Mustard, J. F., Manning, S. J., &Lobell, D. B. (2000). Quantifying vegetation change in semiarid environments: precision and accuracy of spectral mixture analysis and the normalized difference vegetation index. Remote sensing of environment, 73(1), 87-102.

[19] EO-1 Hyperion Vegetation Indices Tutorial ENVI (http://www.harrisgeospatial.com/docs/hyperionvegetationanalysistutorial.html)

[20] Féret, J. B., &Asner, G. P. (2014). Mapping tropical forest canopy diversity using high- fidelity imaging spectroscopy. Ecological Applications, 24(6), 1289-1296.

[21] Ganesh, B. P., Aravindan, S., Raja, S., & Thirunavukkarasu, A. (2013). Hyperspectral satellite data (Hyperion) preprocessing—a case study on banded magnetite quartzite in Godumalai Hill, Salem, Tamil Nadu, India.Arabian Journal of Geosciences, 6(9), 3249-3256.

[22] Gao, B. C., Montes, M. J., Davis, C. O., & Goetz, A. F. (2009). Atmospheric correction algorithms for hyperspectral remote sensing data of land and ocean. Remote Sensing of Environment, 113, S17-S24.

[23] George, R., Padalia, H., & Kushwaha, S. P. S. (2014). Forest tree species discrimination in western Himalaya using EO-1 Hyperion. International Journal of Applied Earth Observation and Geoinformation, 28, 140-149.

[24] Girouard, G., Bannari, A., El Harti, A., & Desrochers, A. (2004, July). Validated spectral angle mapper algorithm for geological mapping: comparative study between QuickBird and Landsat-TM. In XXth ISPRS Congress, Geo-Imagery Bridging Continents, Istanbul, Turkey (pp. 12-23).

[25] Gitas, I. Z., Mitri, G. H., & Ventura, G. (2004). Object-based image classification for burned area mapping of Creus Cape, Spain, using NOAA-AVHRR imagery. Remote Sensing of Environment, 92(3), 409-413.

[26] Goodenough, D. G., Dyk, A., Niemann, K. O., Pearlman, J. S., Chen, H., Han, T., & West, C. (2003). Processing Hyperion and ALI for forest classification. Geoscience and Remote Sensing, IEEE Transactions on,41(6), 1321-1331.

[27] Homer, C., Huang, C., Yang, L., Wylie, B., & Coan, M. (2004). Development of a 2001 national land-cover database for the United States.Photogrammetric Engineering & Remote Sensing, 70(7), 829-840.

[28] Jakubauskas, M. E., Lulla, K. P., &Mausel, P. W. (1990). Assessment of vegetation change in a fire-altered forest landscape. PE&RS, Photogrammetric Engineering & Remote Sensing, 56(3), 371-377.

[29] Jones, T. G., Coops, N. C., & Sharma, T. (2010). Assessing the utility of airborne hyperspectral and LiDAR data for species distribution mapping in the coastal Pacific Northwest, Canada. Remote Sensing of Environment,114(12), 2841-2852.

[30] Kefauver, S. C., Peñuelas, J., &Ustin, S. L. (2012, July). Applications of hyperspectral remote sensing and GIS for assessing forest health and air pollution. In Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International (pp. 3379-3382). IEEE.

[31] Kokaly, R. F., & Clark, R. N. (1999). Spectroscopic determination of leaf biochemistry using band-depth analysis of absorption features and stepwise multiple linear regression. Remote sensing of environment, 67(3), 267-287.

[32] Kruse, F. A., Boardman, J. W., & Huntington, J. F. (1999, February). Fifteen years of hyperspectral data: northern Grapevine Mountains, Nevada. InProceedings of the 8th JPL Airborne Earth Science Workshop: Jet Propulsion Laboratory Publication, JPL Publication (pp. 99-17).

[33] Laurin, G. V., Chan, J. C. W., Chen, Q., Lindsell, J. A., Coomes, D. A., Guerriero, L., ... &Valentini, R. (2014). Biodiversity mapping in a tropical West African forest with airborne hyperspectral data. PloS one, 9(6), e97910.

[34] Laurin, G. V., Chen, Q., Lindsell, J. A., Coomes, D. A., Del Frate, F., Guerriero, L., ... &Valentini, R. (2014). Above ground biomass estimation in an African tropical forest with lidar and hyperspectral data. ISPRS Journal of Photogrammetry and Remote Sensing, 89, 49-58.

[35] Leutner, B. F., Reineking, B., Müller, J., Bachmann, M., Beierkuhnlein, C., Dech, S., &Wegmann, M. (2012). Modelling forest α-diversity and floristic composition—On the added value of LiDAR plus hyperspectral remote sensing. Remote Sensing, 4(9), 2818-2845.

[36] Lu, D., Li, G., Moran, E., Dutra, L., &Batistella, M. (2014). The roles of textural images in improving land-cover classification in the Brazilian Amazon. International Journal of Remote Sensing, 35(24), 8188-8207.

[37] Lu, S., Lu, X., Zhao, W., Liu, Y., Wang, Z., &Omasa, K. (2015). Comparing vegetation indices for remote chlorophyll measurement of white poplar and Chinese elm leaves with different adaxial and abaxial surfaces. Journal of experimental botany, erv270.

[38] Martin, M. E., Newman, S. D., Aber, J. D., & Congalton, R. G. (1998). Determining forest species composition using high spectral resolution remote sensing data. Remote Sensing of Environment, 65(3), 249-254.

[39] Matthew, M. W., Adler-Golden, S. M., Berk, A., Felde, G., Anderson, G. P., Gorodetzky, D., ... & Shippert, M. (2002, October). Atmospheric correction of spectral imagery: evaluation of the FLAASH algorithm with AVIRIS data. In Applied Imagery Pattern Recognition Workshop, 2002. Proceedings. 31st(pp. 157-163). IEEE.

[40] McGwire, K., Minor, T., & Fenstermaker, L. (2000). Hyperspectral mixture modeling for quantifying sparse vegetation cover in arid environments. Remote Sensing of Environment, 72(3), 360-374.

[41] Mitri, G. H., &Gitas, I. Z. (2010). Mapping postfire vegetation recovery using EO-1 Hyperion imagery. Geoscience and Remote Sensing, IEEE Transactions on, 48(3), 1613-1618.

[42] Mobasheri, M. R., Rezaei, Y., &ValadanZoej, M. J. (2007). A method in extracting vegetation quality parameters using hyperion images, with application to precision farming. World Appl. Sci. J, 2(5), 476-483.

[43] Mora, A., & Nelson, P. (2014). Hyperspectral remote sensing for detecting vegetation affected by hidrocarbons in the amazon forest.

[44] Mundt, J. T., Glenn, N. F., Weber, K. T., Prather, T. S., Lass, L. W., & Pettingill, J. (2005). Discrimination of hoary cress and determination of its detection limits via hyperspectral image processing and accuracy assessment techniques. Remote Sensing of Environment, 96(3), 509-517.

[45] Mutanga, O., & Skidmore, A. K. (2004). Narrow band vegetation indices overcome the saturation problem in biomass estimation. International Journal of Remote Sensing, 25(19), 3999-4014.

[46] Nagendra, H., &Rocchini, D. (2008). High resolution satellite imagery for tropical biodiversity studies: the devil is in the detail. Biodiversity and Conservation, 17(14), 3431-3442.

[47] Pirotti, F., Laurin, G. V., Vettore, A., Masiero, A., &Valentini, R. (2014). Small footprint full-waveform metrics contribution to the prediction of biomass in tropical forests. Remote Sensing, 6(10), 9576-9599.

[48] Pu, R., Kelly, M., Anderson, G. L., & Gong, P. (2008). Using CASI hyperspectral imagery to detect mortality and vegetation stress associated with a new hardwood forest disease. Photogrammetric Engineering & Remote Sensing, 74(1), 65-75.

[49] Rasel, Sikdar MM, et al. "Endmember identification from EO-1 Hyperion L1_R Hyperspectral data to build saltmarsh spectral library in Hunter Wetland, NSW, Australia." SPIE Remote Sensing. International Society for Optics and Photonics, 2015.

[50] Sahoo, R. N., Pargal, S., Pradhan, S., Krishna, G., & Gupta, V. K. Processing of Hyperspectral Remote Sensing Data.

[51] San, B. T., & Süzen, M. L. (2011). Evaluation of cross-track illumination in EO-1 Hyperion imagery for lithological mapping. International journal of remote sensing, 32(22), 7873-7889.

[52] Steyaert, L. T., Hall, F. G., & Loveland, T. R. (1997). Land cover mapping, fire regeneration, and scaling studies in the Canadian boreal forest with 1 km AVHRR and Landsat TM data. Journal of Geophysical Research: Atmospheres, 102(D24), 29581-29598.

[53] van der Meer, F. (2006). The effectiveness of spectral similarity measures for the analysis of hyperspectral imagery. International journal of applied earth observation and geoinformation, 8(1), 3-17.

[54] Vauhkonen, J., Ene, L., Gupta, S., Heinzel, J., Holmgren, J., Pitkänen, J.& Lien, V. (2011). Comparative testing of single-tree detection algorithms under different types of forest. Forestry, cpr051.

[55] Yuan, J., & Niu, Z. (2008, June). Evaluation of atmospheric correction using FLAASH. In Earth Observation and Remote Sensing Applications, 2008. EORSA 2008. International Workshop on (pp. 1-6). IEEE.

[56] Zarco-Tejada, P. J., Berni, J. A., Suárez, L., Sepulcre-Cantó, G., Morales, F., & Miller, J. R. (2009). Imaging chlorophyll fluorescence with an airborne narrow-band multispectral camera for vegetation stress detection. Remote Sensing of Environment, 113(6), 1262-1275.

Visual and coded geoprocessing workflows based on ILWIS and Python

Rob Lemmens, Jamshid Farifteh, Claudio Piccinini, Bas Retsios,
Martin Schouwenburg, João Bonina
Department of Geoinformation Processing
Faculty of Geo-Information Science and Earth Observation (ITC),
University of Twente
Enschede, The Netherlands
r.l.g.lemmens@utwente.nl

***Abstract*— Matching software development with user requirements is not an easy task, neither being flexible after the first code release. We believe that representing workflows visually and deploying them in rapid application development using a programming language like Python can play a big role in easing this process. The work described aims at allowing people at different skill levels to create and communicate geoprocessing workflows and deploy them on different platforms. We provide a use case in the MaMaSe project which we run with several partners in the setting of water management in Kenya.**

Keywords—workflow; geoprocessing; Python; GIS; web processing

XV. VISUAL GEOPROCESSING WORKFLOWS

A workflow represents a combination of process steps designed by humans and handled by computers. Geoprocessing workflows consist of geodata (satellite images, in-situ sensor data, human sensor data) and the operations required for their analysis, storage and presentation. In this research we develop methods for creating and sharing geo-workflows between humans and computers to support knowledge sharing and system interoperability. A workflow suitable for human interaction needs to be simple and visual, basically includes boxes representing data and operations and arrows indicating the flow between them. Visual workflows should come in machine-readable representations for execution, storage and inter-machine exchange. We distinguish between (1) abstract workflows, in which humans express their process logic, but not necessarily know or want to express the actual data and software operations and (2) concrete workflows, which contain process steps executable by a specific software.

In our efforts we aim at bridging abstract and concrete workflow representations for the sake of easing the creation and sharing of simple geoprocessing logic within projects. Several generic process languages such as BPMN [1] allow standardized sharing of workflows, but they are not practical for filling the gap between abstract and concrete workflows. We are developing a solution in the following steps:

1. Visual workflow builder tightly-coupled in GIS software (ILWIS).
2. Visual workflow built with web editor.
3. Roundtrip workflow conversion to Python.
4. Ontology-based workflow processing.

These steps are briefly described as follows: step 1 and 2 in Section 3 (current implementation), step 3 and 4 in Section 5 (planned work).

XVI. INTRODUCING ILWIS

ILWIS (The Integrated Land and Water Information System) (http://52north.org/communities/ilwis) [2] is a Windows-based, integrated GIS and Remote Sensing (RS) open source software supporting:

- Visualization and manipulation of raster, vector and tabular data
- Interactive retrieval of attribute information associated with spatial data
- Implementing/developing algorithms for air/space borne image data processing
- GIS analytical tools
- Modeling dynamic spatial systems behaviors using modeler tools

- 3D analysis of spatial temporal events using space-time-cube' tools
- Computations on raster and tabular data using Python command lines
- Direct access to spatial and tabular data of any format (via connectors)
- Using remote server via WPS-ILWIS Bridge
- Multitask operations
- Educational platform for teaching GIS and RS
- Python script language for automation of repetitive procedures.

Anchored in many research projects and training activities, ILWIS has a rich development history and wide user base amongst geo-information professionals. The new ILWIS 4 is a modular GIS and RS software platform, emphasizing ease of use, customization and integration with third-party software. Researchers, trainers and students can now easily implement, store and share their methods via software, in addition to their written reports.

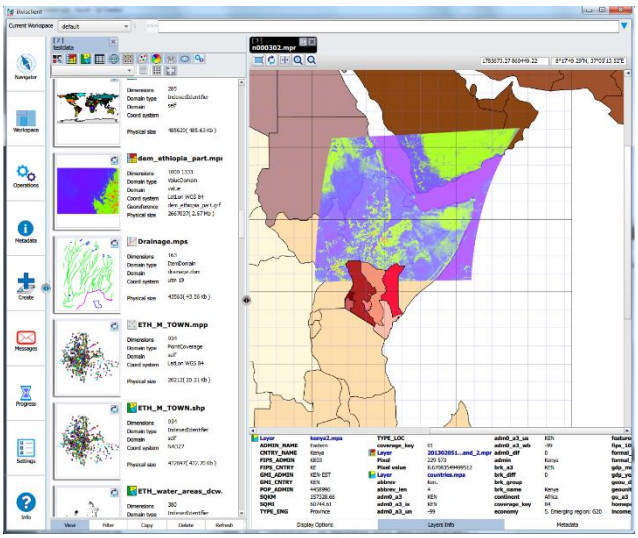


Fig. 1. ILWIS 4 Graphical User Interface

ILWIS 4 contains a highly interactive drag-and-drop and touch-enabled desktop application, based on map thumbnails and linked views: changes in the data properties panel are interactively visible in the map window.

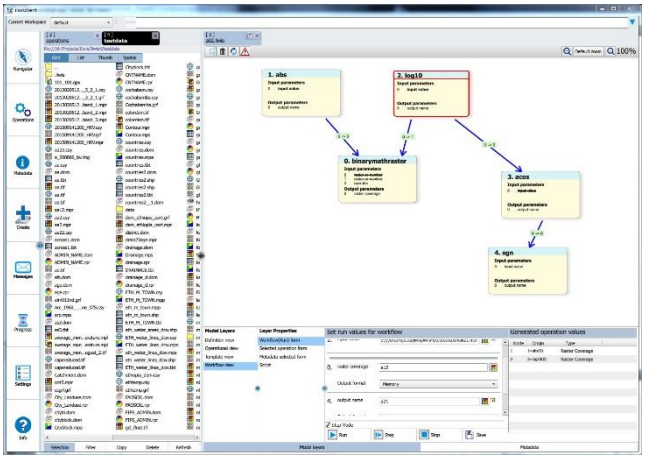


Fig. 2. ILWIS Workflow Builder.

The underlying framework, ILWIS-Objects, supports the extension of functionality through Python scripts and the creation of new connectors to data formats, programming languages and software libraries, such as R and to other user environments, such as web and mobile.

XVII. WORKFLOW BUILDER

The ILWIS workflow builder (see Fig. 2) is a tool which facilitates the creation of flexible workflows, allowing any geoprofessional to simply apply raster and vector operations to geodata by drag-and-drop on a canvas. The workflow builder assists the user by visual validation of each part of the workflow.

The ILWIS workflow builder stands out in the handling of aggregated workflows and conditional statements in order to support the larger GI workflows. For the purpose of sharing the workflows, we currently export them in a JSON format (see Listing 1). This can be used in a web environment (see Fig. 3).

```
"subworkflows": [
    {
        "id": 0,
        "metadata": {
            "longName": "PRODUCTION",
            "description": "The workflow for the MaMaSe project",
            "syntax": "MaMaSeWorkflow(raster,raster,raster,raster)",
            "resource": "Ilwis",
            "keywords": "workflow, MaMaSe, drainage",
            "inputParameterCount": 4,
            "outputParameterCount": 2
        },
        "operations": [
            {
                "id": 0,
                "metadata": {
                    "longName": "Anoperation",
                    "description": "AnOperationDescription ",
                    "syntax": "thefirstoperation(inputrastermap)",
                    "resource": "Ilwis",
                    "keywords": "operation, keyword,operation",
                    "inputParameterCount": 1,
                    "outputParameterCount": 1,
                    "final": false
                },
                "inputs": [
                    {
                        "id": 0,
                        "url": "veg.com",
                        "term": "",
                        "type": "map",
                        "value": "",
                        "units": "",
                        "min": "",
                        "max": "",
                        "name": "Vegetation structure",
                        "show": true,
                        "change": true,
                        "description": "",
                        "picture": ""
    }
```

Listing 1. JSON file for exchanging workflows.

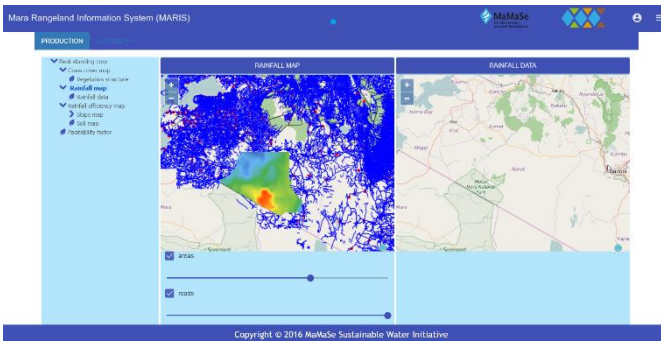


Fig. 3. Web interface for running workflows in the MaMaSe project (for a description of its use, see Section 5).

XVIII. PYTHON API

Besides the fact that Python is an open source programming language used for object-oriented programming as well as for scripting and rapid application development, it is a relatively easy to learn language and makes it suitable for developers without high-level programming skills. Therefor we opted to implement a Python API on top of ILWIS, which core is developed in C++. From the Python interpreter all ILWIS functionality can be imported. This allows users to write high level scripts for multi-source geoprocessing, in a standard scripting environment, running on different platforms. The Python interpreter and its extensive standard library are included in ILWIS, so there is no need for a separate installation. However, Python may be installed independently and then ILWIS objects may be imported which will support ILWIS raster and vector operations.

A key advantage of having the Python API is that we can combine Python-based libraries with ILWIS operations. The example shown in Listing 2 opens a file with ILWIS-Objects, accesses its attribute table, analyses the table data with Pandas, and uses matplotlib to plot the data. The result is shown in Fig. 4.

```
import ilwisobjects as ilw
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

#first we need to set the working catalog
ilw.Engine.setWorkingCatalog(r"file:///[folder]\data")
#we create a feature coverage
world = ilw.FeatureCoverage("countries2.shp")
#we access the attribute table
table=world.attributeTable()
#get some statistics: number of records, number of columns a
tuple with column names
recCount = table.recordCount()
colCount = table.columnCount()
columns = table.columns()

#create a dictionary; key is the field name, value is a tuple with
field values
t={}
for i in columns:
    t[ i ] = table.column(i)

#create a Pandas DataFrame; use country names for the row
index
df=pd.DataFrame(t,index=table.column('name'))

#initialize a matplotlib figure that will contain 2 subplots
fig=plt.figure(figsize=(10,10))

#first subplot; group data by continent and sum it to get the total
population per continent
plt.subplot(211)
gr=df.groupby('continent').aggregate(np.sum)
#plot the total population per continent
 gr['pop_est'].plot(kind='barh')
plt.subplot(212)

#second subplot; group data by type of income and sum it to get
the total population per type of income
gr=df.groupby('income_grp').aggregate(np.sum)
gr['pop_est'].plot(kind='pie')
plt.show()
```

Listing 2. Integration of ILWIS with other software libraries through the Python API.

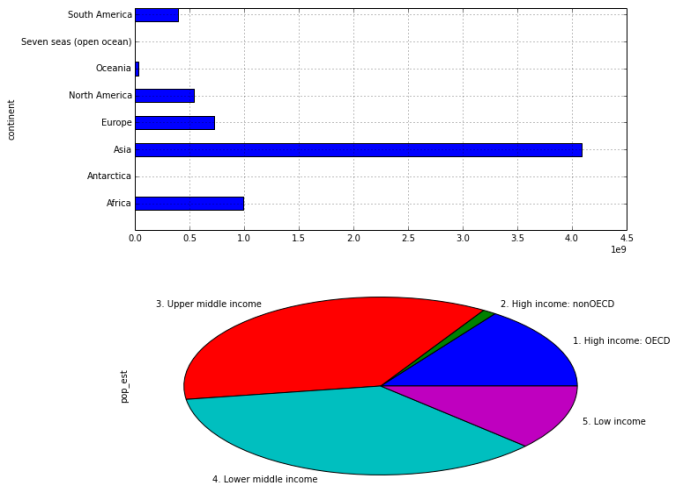


Fig. 4. Result of the Python code in Listing 2: World population by continent (top) and by income (bottom).

XIX. Application use case (MaMaSe, Kenya)

The Mau Mara Serengeti (MaMaSe) project (http://www.mamase.org) aims at the improvement of water safety and security in the Mara River Basin, Kenya.

```
ilwisdemo3.py - D:\ilwisdemo3.py (3.4.3)
File  Edit  Format  Run  Options  Window  Help
#save individual polygons into shapefiles:

import ilwisobjects as ilwis
units = ilwis.FeatureCoverage("file:///D:/ilwis4/Bou
rfe = ilwis.RasterCoverage("file:///D:/ilwis4/rfe201
processlist = ["Maji Moto", "Naikarra", "Enonkishu",
for unit in units:
        name = unit["Name"]
        if name in processlist:
                outname = "output_" + name.replace("
                geom = unit.geometry()
                rcSel = rfe.select(geom)
                stats = rcSel.statistics(ilwis.Prope
                rainfall = stats.prop(ilwis.Property
                print(name, rainfall)
                rcSel.store("file:///D:/ilwis4/" + c
                fc = ilwis.FeatureCoverage(outname)
```

Listing 3. Calculation of Average rainfall per conservancy in Mara region, Kenya.

MaMaSe embarks upon the strengthening of governmental institutions' management, residents' economic situation and protection of forest and savannah ecosystems. Geography-based analyses are key to addressing the abovementioned issues and the project approach is to develop a web-based system that can be operated by local stakeholders. An example Python script is run with ILWIS operations as in Listing 3. The developers have created this code, based on interactive discussions and iteratively refining the visual workflow as represented in Fig. 5. This workflow with inputs shown in Fig. 6 was then also used to create a JSON file (see Listing 1), which was used to populate the workflow template on the project website (see Fig, 3). In this web application, Kenya conservancy managers can run the rainfall analysis and vary its parameters. The first evaluations are promising, but the web application still needs fine tuning.

XX. Discussion and outlook

The work described aims at allowing people at different skill levels to create and communicate geoprocessing workflows. Workflows are currently created in ILWIS in a visual way and can be shared through a JSON file with other applications. The JSON file is converted to the representation in ILWIS and on the Web. Work is underway to smoothen the conversion of JSON workflows to Python code and vice-versa. The exchange of these JSON files appears to be suitable for the exchange with web applications from which users have interactive but restricted access to the ILWIS server application and which allows system engineers to update the workflows easily. In addition, we are connecting to geodata and geoprocessing ontologies (see [3]) in order to be able to translate concrete workflows into abstract workflows and – in the other direction- to support users to create workflows from abstract to concrete ones.

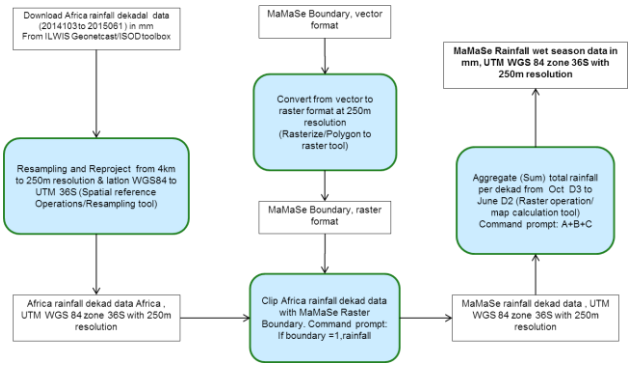


Fig. 5 MaMaSe calculation of average rainfall: concrete workflow representation.

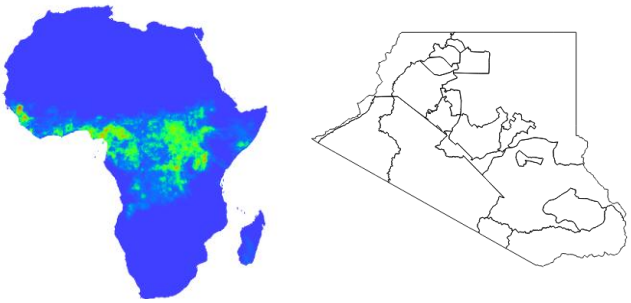


Fig. 6 MaMaSe calculation of average rainfall: Input rainfall map (left), input boundary map (right)

References

[1] Campagna, M. (2014). Orchestrating the spatial planning process : from Business Process Management to 2nd generation Planning Support Systems. http://www.agile-online.org/Conference_Paper/cds/agile_2014/agile2014_132.pdf

[2] ILWIS Open Source GIS/RS software. Video introduction (2016) https://vimeo.com/user29453510/review/153355429/1c1a97df84

[3] De Carvalho Diniz, F. (2016) Composition of semantically enabled geospatial web services. Enschede, University of Twente Faculty of Geo-Information and Earth Observation (ITC), 2016.