Automating Information Flow in Hyperspectral Measurements for Soil Characterization in Tailings

by

Shaina Mahajan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Engineering Management

Department of Mechanical Engineering
University of Alberta

Abstract

In the first section of this thesis, two laboratory experiments conducted on tailing samples were evaluated using a structured qualitative analysis procedure. The process model of Suppliers, Inputs, Process, Output, and Customers (SIPOC) was used to study relationships amongst different parts of the experimental processes and to build a process model that fits the output requirements and covers all essential parts of the system. This model was chosen for this study as it allows for different aspects of system description from supply of samples to the delivery of reports to the customers. As part of the model, a process flow diagram was developed to establish how inputs are used to produce outputs identified in the SIPOC model and every activity of the process flow was analysed using an Integrated Definition - 0 (IDEF0) model. A process value analysis was finally performed to identify non-value added activities, if any, in the procedure. The results indicated the absence of non-value added activities in both procedures and emphasized the importance of execution of every element of the process. It, however, suggested automation of repetitive tasks and active database management.

In the second section of this thesis, two procedures were created to describe bitumen profiles of total bitumen content (TBC) in drill cores scanned using a hyperspectral imaging system, so that information flow could be streamlined and low value-added processes might be automated. The profiles represent variation in TBC as a function of depth along the drill core and convey statistical parameters computed at a range of drill core downhole resolutions. Two procedures were created in the Environment for Visualizing Images (ENVI) and written in Interactive Data Language (IDL) to create these profiles and were compared on the basis of time and accuracy. The first method, also called "The Basic Approach", presented higher accuracy while the second method, also called "The Improved Method", reduced the processing time from hours to a few minutes. The profile generation tools would allow facilitation of multi data analysis by providing comparison to other forms of profiles, such as geophysical. They, however, don't consider missing drill cores and ignore cracks in the core samples.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisors Dr. Michael Lispett and Dr. Benoit Rivard for providing me the opportunity of being a part of their research team. They have continuously guided me, motivated me and provided me with immense knowledge. They have been great mentors in the last two years and created a wonderful learning experience. I am also grateful to them for spending time to discuss the research whenever it was required, it would have been impossible to complete my research without their cooperation.

I would also like to thank Dr. Jilu Feng for helping me in learning the basics of IDL and in determining the right approach at times of confusion. He never hesitated to discuss even the smallest parts of research and provided invaluable suggestions to help build the IDL tools that were created as a part of this thesis. I am also very grateful to Iman Entezari for taking time out of his busy schedule to explain his experimental procedures and to review process flow diagrams, without which it would have been impossible to complete the first section of this thesis.

I would like to acknowledge the love, inspiration, and encouragement of my parents and my brother, and the many years of support that helped me pursue Masters at such an esteemed university. They helped me realize my own potential and provided great advice when most needed. Without their love and guidance, I may have never gotten where I am today.

I am also grateful to my friends at Mece 6-29 for creating a warm and friendly environment through the course of this research. I would also like to thank my friends Bharti Wadhwa, Priya Mahajan, Saurabh Bharti and all others who supported me through this entire process. Finally, I express my heartfelt gratitude to my boyfriend, Dibyo Sarkar, who has constantly encouraged and helped me from the very beginning. His endless support, guidance, understanding, and love made the completion of this thesis possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Oil Sands

Oil sand is a naturally occurring mixture of sand, clay or other minerals, water and bitumen. Bitumen is a heavy and extremely viscous petroleum that must be treated before it can be used by refineries to produce usable fuels such as gasoline and diesel [1]. Canada is home to vast reserves of oil sands with three main deposits located in the Athabasca, Peace River and Cold Lake regions (Figure 1-1). These have combined proven reserves of about 168 billion barrels-the third-largest proven crude oil reserve in the world, after Saudi Arabia and Venezuela [2].

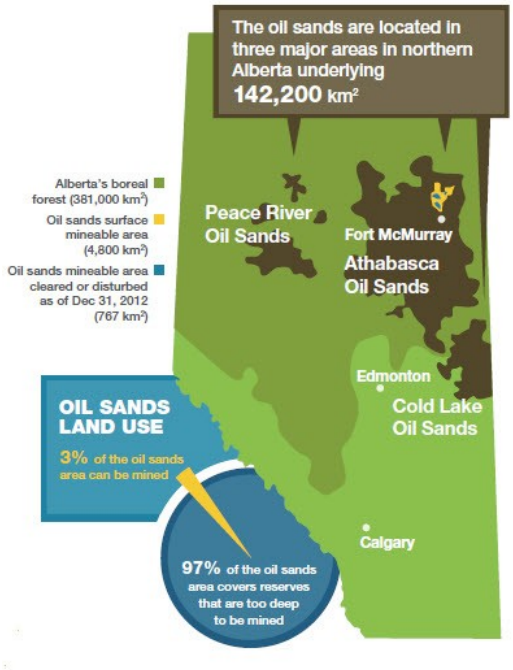

Figure 1-1: Three main oil sands located in Northern Alberta [2]

1.2 Extraction methods

The two current commercial methods of oil sands extraction are in-situ recovery and surface mining. The selected method depends on how deep the reserves were deposited. Surface mining is used for deposits closer to the surface and the in-situ recovery method is used for deposits buried at depths of more than 100 metres, which applies to 80% of Alberta's oil sands [3]**.** Drill core sampling is the principal method for investigating the subsurface geology in both in-situ and mining operations in the Athabasca oil sands [4]. Drill holes, also known as wellbores, are drilled in order to remove a cylindrical core sample from the earth [5] which is then split longitudinally. Cores are manually logged by geologists for facies classification [4].

Surface mining methods employ trucks and shovels, similar to those methods used in coal, copper and diamond mine operations. The operators of the Athabasca McMurray region use a water-based bitumen separation method proposed by Dr. Karl A Clark at the Alberta Research Council [6]. The Clark Hot Water Process (CHWP) achieves a bitumen recovery of over 90% at a water temperature of 85°C or less [7]**.** Ore mined at the surface is processed via hot water extraction, which involves forming a slurry of crushed ore to which a reagent is added to promote liberation of the bitumen from the oil sand matrix, coalescence of bitumen droplets, and aeration, thereby creating a bitumen-rich froth that can be skimmed [8]. The process waste from this method is called tailings. Because of a zero-discharge policy for companies operating in the Alberta oil sands, tailings are held in large surface impoundments [9]. The current practice for the bitumen production process consists of these steps: Ore Preparation, Slurry Preparation and Hydro-Transport, Primary Extraction, Froth Treatment (Secondary Extraction), Slurry Transport of Tailings [10]. In the ore preparation process, ore brought from oil sands are crushed in a primary crusher. Crushed ore is discharged to a conveyer belt designed to transport crushed ore

from crusher to a surge facility and from the surge to a rotary breaker or mixing box, where crushed ore is mixed with hot water and caustic soda and forms ore slurry which is then transferred to hydro-transport pipelines [11]. The hydro-transport slurry pipeline performs the combined functions of "conditioning" the oil sands for bitumen separation from the solids matrix, while at the same time delivering the ore to the bitumen recovery section of the extraction process [10]**.** At the extraction plant, two steps of bitumen recovery are performed: primary recovery and secondary recovery. In primary recovery, slurry is transferred to a large diameter vessel. Bitumen floats to the top and froth at the top of the column is skimmed and collected as a product. This froth contains about 60% bitumen, 30% water and 10% mineral solids. Dense tailing are withdrawn from the bottom of the vessel and a middling stream is drawn from the side. The middlings stream undergoes second recovery. In this step, air addition and agitation produces further aerated bitumen. The middlings are recovered as froth in a range of devices, including deep-cone settlers and mechanical flotation cells [12]. Both recovery steps produce waste streams that are blended and transferred to tailing ponds. The bitumen froth recovered from these steps is then de-aerated, and diluted (mixed) with solvents to provide sufficient density difference between water and bitumen and to reduce the bitumen viscosity. The dilution by a solvent facilitates the removal of the solids and water from the bitumen froth using inclined plate settlers, cyclones, and/or centrifuges [8]. The diluted bitumen is sent to an upgrading plant afterwards and the froth treatment tailings stream also goes into the tailings pond.

As specified, during the Clark Hot Water Process of bitumen extraction, large volumes of water are used to float bitumen. Approximately 12 barrels of water are required to produce each barrel of bitumen although much of the water is recycled [14]. This process produces a significant by-

product of tailings stream, which contains water, mineral components, and unrecovered bitumen. The minerals consist of a coarse (>22 μm) sand component and a fines (<22 μm) fraction made up of silts and clays (predominantly kaolinite and illite) [15]. When tailings are discharged into the ponds, the sand particle quickly settle and up to 30% of the remaining fines settle after a few years. The 30wt% solids content fines are called Mature Fine Tailings (MFT) [8]. Excess water in tailings should be removed from the ponds so that the area can be reclaimed and revegetated. However, the process of water separation is a challenge and natural consolidation of these fine tailings would require hundreds of years and remain poor candidates for terrestrial reclamation [15]. The industry continues to develop better technologies and approaches to tailings management in order to reduce the environmental impact [16]. Multiple techniques have been proposed to accelerate water removal and the consolidation process. Some of them being MFT Evaporation Dry and Freeze-Thaw, In-Situ Densification, Thickened Tailings, Accelerated Dewatering, Centrifiguted MFT, MFT Water Capped Lake and Composite Tailings [17]. It is important to estimate the moisture content of tailings to evaluate the efficiency of these techniques and to monitor the state of the tailings surface [18].

The second extraction method of in-situ recovery uses technologies to inject steam, combustion or other sources of heat into the reservoir to warm the bitumen – oil that is too thick to flow - so it can make its way to the surface through recovery wells [19]. The steam assisted gravity drainage (SAGD) method, initially proposed by Butler and Stephens [20], is the most commonly used in-situ extraction method, with oil recovery of up to 55% [21]. This method involves pumping steam underground through a horizontal well to reduce the viscosity of the bitumen, which flows to a second recovery well and is then pumped to the surface along with produced water [22] (figure 1-2). These wells, steam injection well and production well, are paired at a

distance of 5 m. The method is very effective because the injected steam continuously rises and heats the bitumen in the formation, both flowing to the surface [23]. The bitumen is sent to a process facility where water, gas, and impurities are removed.

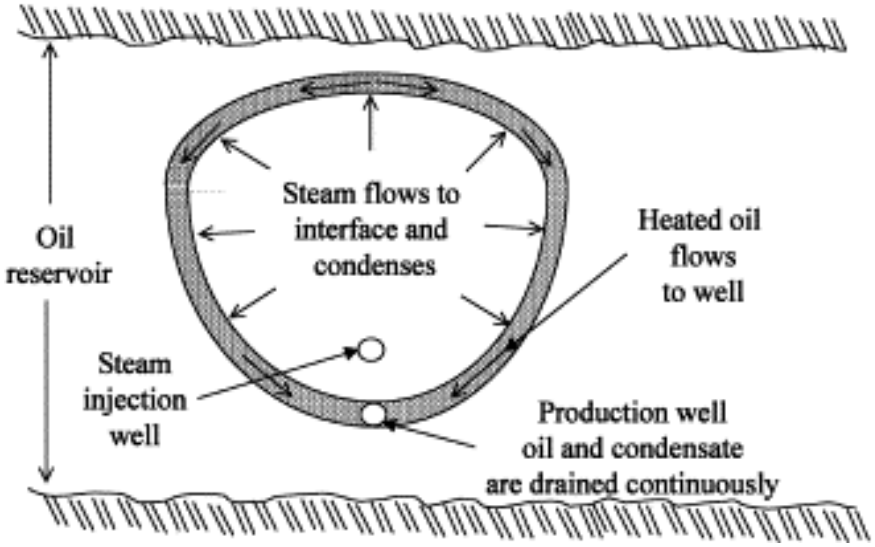

Figure 1-2: Steam-assisted gravity drainage concept Reprinted from "A laboratory study of single-well steam-assisted gravity drainage process" [24] , Copyright 2001 with permission from Elsevier

This research examines applications of hyperspectral imaging that relate to both tailings and drill core.

1.3 Hyperspectral Imaging

Hyperspectral imaging is a fast growing area in remote sensing. It expands and improves capability of multispectral image analysis [25]. Hyperspectral imaging, known also as chemical or spectroscopic imaging, is an emerging technique that integrates conventional imaging and spectroscopy [26] to attain both spatial and spectral information from the surface of an object. The data is often described as forming a hyperspectral cube which contains two spatial dimensions and one spectral dimension (Figure 1-3). Using high-resolution digital imaging technology based on diffraction optics that disperse different wavelengths of light, hyperspectral

sensors offer the user a wealth of important information and provide significantly more analytical power than currently available with single-point spectroscopy [27].

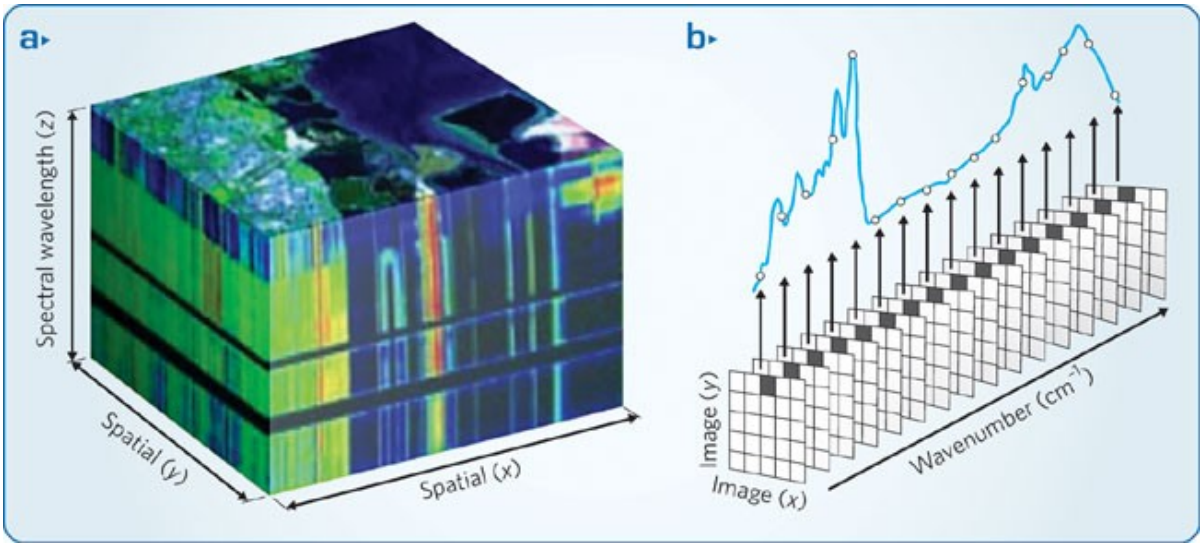

Figure 1-3: a) Hyperspectral cube formed as product passes the sensor b) Hyperspectral cube is a three dimensional image comprised of spatial (x and y coordinates) and spectral data (wavelength). Reprinted by permission from Macmillan Publishers Ltd: Hyperspectral imaging: Cubes and slices [27], copyright 2009

Hyperspectral images as seen in figure 1-3 are made up of hundreds of contiguous wavebands for each spatial position of a target studied. Consequently, each pixel in a hyperspectral image contains the spectrum of that specific position [26]. Pixels along the x-axis of the image are referred to as "samples", along the y-axis are referred to as "lines" and wavelengths are also called spectral bands.

So far, significant hyperspectral research applied to oil sands has included the estimation of soil moisture. Entezari et al., [28, 29], investigated laboratory based processes to estimate the moisture content and normalised evaporation of soft tailings. They proposed a model to accurately estimate the water content of tailings using hyperspectral observations. As water evaporates from the surface of tailings, a crust forms which eventually slows the evaporation process. According to Entezari et al., [29] there are a number of factors that affect the rate of

evaporation including climatic conditions and precipitation. In their study, they investigate the potential of using remote sensing methods to develop techniques for the estimation of water content and evaporative fluxes from the surface of tailings using laboratory observations. A need to perform a qualitative analysis of these laboratory experiments was identified. Thus the research in this thesis aims to perform an analysis of these processes in order to search for information gaps while documenting them.

At another front, Speta et al., [4] used hyperspectral imaging to scan drill cores collected from Athabasca oil sands. The process of core logging can be very challenging and samples need to be sent to commercial laboratories for Total Bitumen Content (TBC) estimation using Dean Stark Analysis [30, 31]. This analysis method is a laborious and expensive procedure used in the oil industry for determining oil saturation in cores plugs typically sampled every foot of the core [32]. TBC is determined by weight difference of extracted water and dry extracted solids from the fresh sample. Reflectance spectroscopy has been used in the past by Shaw and Kratochvil [33] to determine TBC which was later optimized by Donkor et al., [34]. Based on Lyder et al.,'s [35] results of using Gaussian fitting and wavelet analysis to identify useful bitumen features, Rivard et al., [36] developed a broad band and wavelet predictive models that improve the accuracy of the estimate of TBC. Speta et al., [4] illustrated the use of a SisuROCK imager to estimate TBC in oil sands drill cores. SisuROCK (Figure 1-4) is a complete, automated, computer-controlled, spectral imaging workstation designed for easy, high-speed scanning of drill cores and other geological samples. It is capable of imaging a single drill core in a few seconds [37]. They used a SisuRock core imaging system manufactured by Spectral Imaging (Specim) Ltd Finland to collect drill core sample images in 256 spectral bands scanning them at a scale of 1 mm per pixel.

Figure 1-4: Specim's SisuROCK Imaging System, courtesy of Dr. Benoit Rivard, Professor and Associate Chair, Department of Earth & Atmospheric Sciences at the University of Alberta

However, how the estimated TBC information needs to be conveyed remains to be determined. Typically in mining operations, the largest datasets come from geophysical logs. There is a need to convey bitumen maps into profiles that can be easily compared to geophysical logs. Thus, in the present work examines how to construct such representations with consideration of time, scale, and methodologies to generate the profiles using drill cores scanned by Speta et al., [4].

1.4 Research Objectives

Following are the key objectives of this study:

- To perform a structured qualitative analysis on two laboratory experimental methods aimed to estimate moisture content and evaporation fluxes in tailings using hyperspectral measurements.
- To create procedures to describe bitumen profiles of total bitumen content estimated in drill cores scanned using a hyperspectral imaging system, so that information flow can be streamlined and low value-added processes might be automated.

1.5 Structure of Thesis

In this consideration, chapter 2 provides procedure of qualitative analysis performed on laboratory experiments aimed to study soft tailings and discusses the results of this analysis.

Chapter 3 provides methods to create profiles of TBC estimates, compares the methods, and discusses results obtained from these methods. Finally, Chapter 4 provides conclusions of the research and recommendations for future work.

Chapter 2

A Qualitative Analysis relevant to Mature Fine Tailings

2.1 Introduction

Business activities and processes generally benefit from analysis to improve the process. This disciplined approach is referred to as continuous improvement, which involves identifying, analyzing, and taking actions to improve existing processes. Improvements can be elimination of steps that slow down the process without adding true value and/or automation of activities to increase the process speed and efficiency. However, the time, effort, and cost involved should bring right values to the system and should not exceed the profit made through these changes. Therefore, an analysis should be done before introducing process improvement strategies in a system. Experimental methods are processes, and so they can be evaluated using structured analysis and design to improve performance, according to some pre-defined metrics.

This study focuses on such a qualitative analysis conducted on two experimental methods performed on tailing samples at the Earth and Atmospheric Department of the University of Alberta. The experiments use spectral measurements to estimate the water content in tailing samples. The array of spectra obtained from these experiments is used to quantify moisture content using measurements of spectral features and their absolute reflectance or absorption depth, and to derive the normalised soil moisture index (NSMI [38]). This chapter discusses the methodology used to conduct this study and the results of the analyses are shared at the end of the study.

2.2 Research Methodology

A key part of improving a process is to understand the process in a way that can be communicated to everyone without the risk of ambiguity [39]. Some of the most common techniques used in process improvement are:

DRIVE [40]: This technique to problem solving and analysis can be used as a part of process improvement. It starts by "D"efining the criteria by which success will be measured, followed by "R"eviewing the current situation, understanding the background, and identifying problem areas. Thereafter, improvements or solutions to the problem are "I"dentified alongwith changes to enable and sustain the improvements. The next step is to "V"erify that improvements will bring changes that meet the defined success criteria. Finally, the plan to implement improvements is "E"xecuted.

Process Mapping [40]: Process maps are useful tools that help a team to understand a problem and identify opportunities of improvement. It requires constructing a dynamic model/picture of the activities that take place in a process therefore allowing a systematic way of working. To construct a process map, all activities are brainstormed, grouped into sub-processes, and sequence of events and links between sub-processes is identified.

SIPOC: SIPOC is short for Suppliers, Inputs, Process, Outputs, and Customers. The system analysis requires the analyst to consider the suppliers, the inputs to the process, the process being studied, the outputs of the process and the customers that receive the process outputs [40]. There is also flexibility to append requirements to the end of the SIPOC model for further details. This framework is used to identify all performance elements relevant for process improvement [38].

Cause and Effect Diagram: Also known as the fishbone diagram, it is useful for opening up thinking in problem solving. As shown is Figure 2-1, the problem being investigated is shown at the right end of a horizontal arrow and potential causes or concerns are shown as labelled arrows entering the main cause arrow.

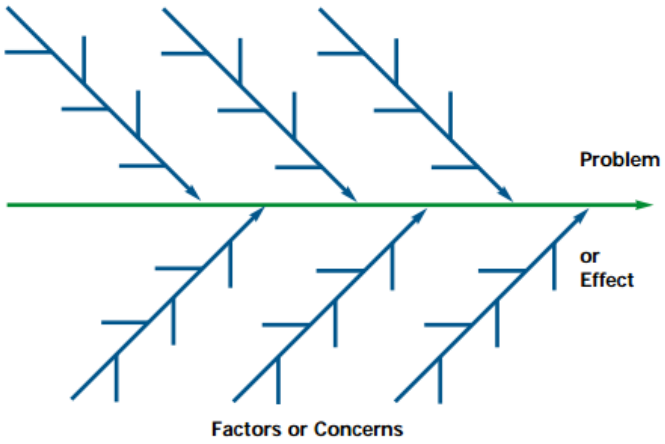

Figure 2-1: Fishbone diagram [39]

For this research, the process model of Suppliers, Inputs, Process, Output, and Customers (SIPOC) was used to study relationships amongst different parts of the processes and build a process model that fits the output requirements and covers all essential parts of the system. This model is chosen for this study as it allows for different aspects of system description from supply of samples to the delivery of reports to the customers. It also allows the user to fill information gaps, to complete connections and to contemplate process improvement. The model is then used to perform a qualitative analysis of the processes to find the value and necessity of executing every step of the process. As part of the model, a process flow diagram was developed to establish how inputs are used to produce outputs identified in the SIPOC model. Each element of the flow diagram was then analysed to study all the inputs needed, outputs created, controls used, and resources required for a successful execution.

In this research, two laboratory experimental cases were studied. The first case studied is a dehydration experiment that observes the behaviour of various spectral features with respect to moisture content in the tailings samples received from an oil sands company partner and aims to create a model to predict moisture content that could potentially be applied in the field (section 2.4). The second case study is an evaporation experiment that uses the normalised evaporation of water and tailings samples to observe the behaviour of different spectral features and creates a predictive model (section 2.5) of normalised evaporation.

2.3 Development of a SIPOC Model

A SIPOC model is built as it is spelled out S>I>P>O>C. However, it becomes difficult to follow this sequence due to the difficulty of identifying suppliers without knowledge of inputs, the complication of defining inputs without knowing the process and so on. Therefore, COPIS (reverse of SIPOC) is a popular approach used in completing a SIPOC model. The first step is to identify primary customers, anyone who receives a product or a service. The next step is to identify the outputs, any products or services provided to these customers as desired by them. The heart of the model is the steps of processes carried out to create the products or services for the customers. After this, the inputs required for these processes are identified. It is important to compare the inputs to the processes to ensure a complete list. Once inputs are listed, the last component of the SIPOC model are identified, namely the suppliers. The selection and management of suppliers is a key success factor for the model. Figure 2-2 shows a simple representation of a complete SIPOC model.

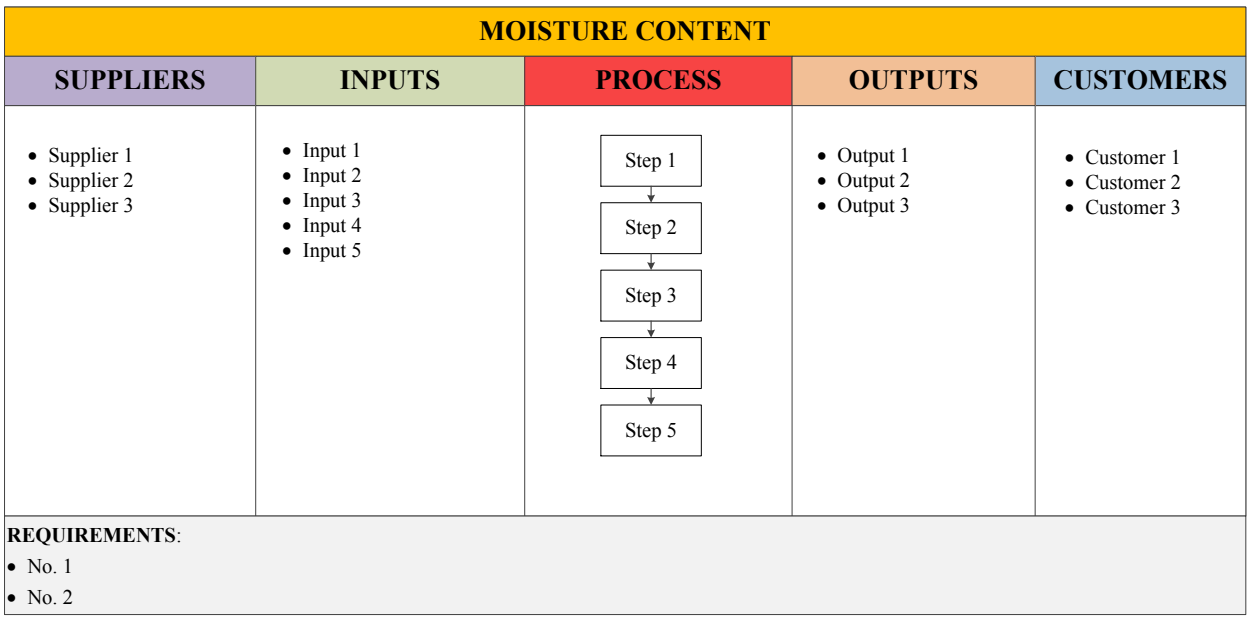

Figure 2-2: The SIPOC model

The next stage involves creating a flowchart depicting all the process steps from the beginning to the end. The purpose of the flowchart is to explicitly describe how the current process operates and to conduct an objective analysis that will identify problems, weaknesses, unnecessary steps, duplication, and confirm the objectives of the improvement effort [38]. It also helps in carefully examining connections between the process steps and in identifying opportunities of improvement. All inputs identified in the SIPOC model feed into the process maps and the outputs are produced at various stages of the process. The activities are grouped into approximately three to four subcategories; and the sequence of events and links between the sub-processes are identified. Accurate description of the sequence of the execution of steps is very important in order to produce a customer desired output.

At the next level, each step of the process is analysed using an Integrated Definition - 0 (IDEF0) model, as shown in Figure 2-3. IDEF0 is an activity modelling method used to provide a structured representation of the functions performed by an organization. It is composed by a

hierarchical series of diagrams that display increasing levels of detail describing functions and their interfaces within the context of a system [41], [42]. The purpose of IDEF0 models is to enable process analysis and identification of business process improvement opportunities. These models are used to perform analysis and must therefore be in a structured form and decomposed to the lowest level needed for analysis [43]. The IDEF0 model is widely used in the research community due to its flexibility and clarity for modelling activities and the information flows between them [41].

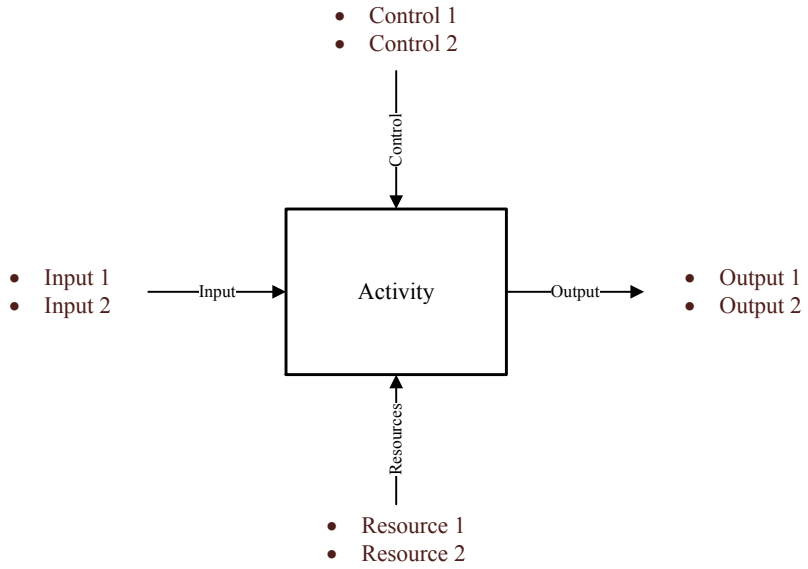

Figure 2-3: IDEF0 model

The model in figure 2-3 represents an activity from a process. Items listed on the left are the inputs needed to trigger the activity, the right list represents results of performing the activity, the list on the top are controls necessary to guide or regulate the activity such as plans, budget, and standards and at the bottom are the resources such as machines, tools and people required to perform the activity. Often times there is uncertainty whether an item is a control or an input, the distinguishing factor is that the controls influence the activity of execution but are not consumed or transformed. Each activity of the process is carefully analysed with the use of this model to

create a consistent flow, to find possible redundancies, and to evaluate its qualitative and/or quantitative value. For this research, we focused on assessing the qualitative value of the process using a "Process value analysis" approach described in the next section.

2.4 Process value analysis

Process value analysis is a qualitative analysis procedure that can be used for process improvement with the goal of saving time and resources. The analysis can be used to streamline or eliminate activities to decrease operational costs while meeting customer requirements without sacrificing quality [44]. The process steps are divided into three different categories of activities [45]:

Customer Value Added - These activities directly serve the customers and impact their satisfaction. These are necessary to meet the timelines and expectations of the customers.

Operational Value Added - These activities support the ability to deliver services to the people served [45]. These activities are reviewed to check if there are any unnecessary repetitions allowing improved efficiencies.

Non-value Added - These activities do not add value for either the customers or for operations. They warrant further analysis to determine whether they should stay the same [45]. Any type of rework is usually an example of a non-value added activity.

Once the activities are categorized, a flowchart of the process is created. Non-value added are highlighted and analysed to identify actions to reduce or eliminate these activities. Value added steps are also analysed to identify improvement opportunities. Proposed changes are then

simulated and compared to the base-case scenario and the target progress. Gaps for further improvement are thus identified on an ongoing basis until the target is achieved.

The following sections discuss the application of the methodology on the two laboratory experiments performed on tailing samples at the Earth and Atmospheric Sciences Department of University of Alberta.

2.5 Case Study 1: Dehydration Experiment

The aim of this experiment is to derive a model that can be used for the prediction of moisture content in tailings using hyperspectral measurements. It evaluates the relationship between the moisture content of tailings samples and spectral features and their absolute reflectance and absorption depth for water bands centered at 1450 nm and 1925 nm, and to derive the normalised soil index values (NSMI). The hyperspectral measurements were obtained using a spectrometer while tailings sample dried during exposure to artificial lighting. As already mentioned, the framework for describing this experiment is developed in three stages.

2.5.1 Step 1: SIPOC Framework

The SIPOC model shown in figure 2-4 represents the dehydration experiment. An oil sand company partner provided samples for the experiment accompanied by a sheet with information on each sample's number, date, % solids content, Methylene Blue Index (MBI) and yield stress. Panalytical is contacted to provide an ASD Terraspec spectrometer for collecting hyperspectral measurements. In this case the experimental equipment is provided by the department of Earth and Atmospheric Sciences at the University of Alberta. This includes spectrometer, computers, lamps, scale, and laboratory space.

MOISTURE CONTENT				
SUPPLIERS	**INPUTS**	**PROCESS**	**OUTPUTS**	**CUSTOMERS**
• An oil sand company • Panalytical • Earth and Atmospheric Sciences at the University of Alberta	• Samples • Characteristic Sheet from oil sands company - Sample No. - Date - % solids - MBI - Yield Stress • ASD Spectrometer • Equipment	Dehydration experiment	• Behaviour of different spectral metrics with respect to moisture content • A robust predictive model • R square values	• IOSI Team
REQUIREMENTS: • Samples should have a wide range of MBI characteristics to get accurate results • Predictive model produced should be as accurate as possible, ensuring minimum errors				

Figure 2-4: SIPOC model of dehydration Experiment

A sketch demonstrating relation of spectral metrics with moisture content forms one of the outputs (shown in figure 2-5). Each graph has an R square value of error, forming the basis on which the most robust model is selected as a moisture content estimator [46]. The end results are shared with the Institute of Oil Sands Innovation (IOSI) team and the oil sands company partner.

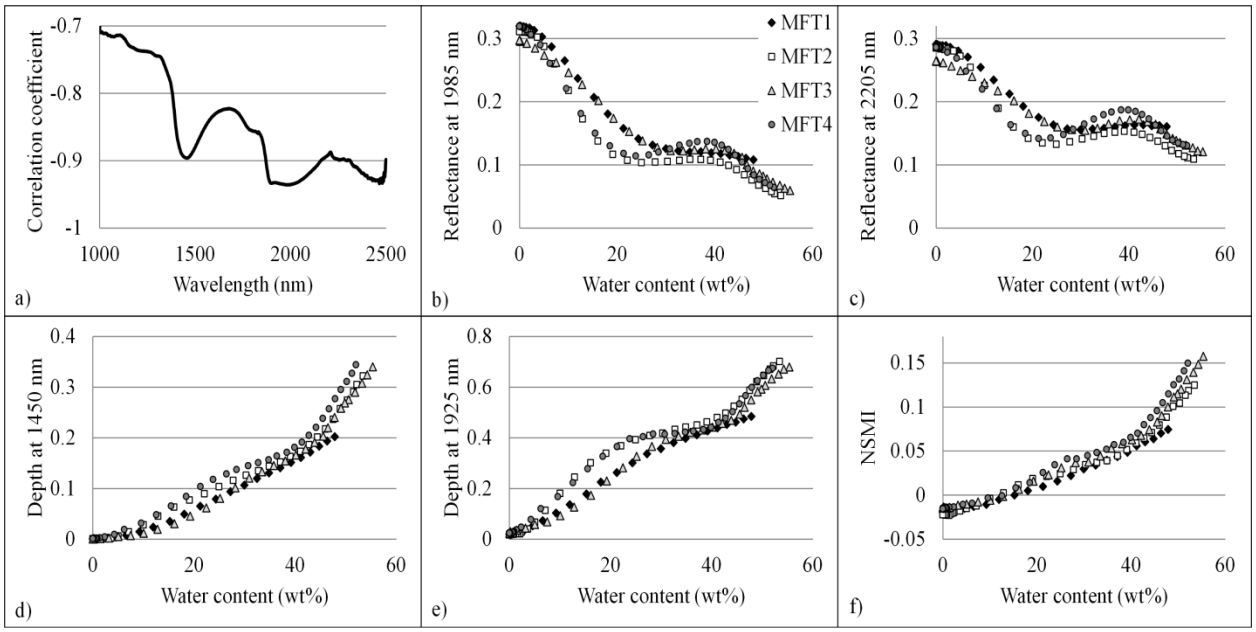

Figure 2-5: Spectral metrics vs. Moisture content (wt%) NSMI stands for normalised soil moisture index. [46]

The model also includes some requirements for the execution of this experiment. The tailings samples provided by the oil sands company were selected to show a wide range of MBI values to assess the sensitivity of the spectral measurements to the composition of samples. [46] This also helps in building a robust predictive model. The customer also requires the results to be as accurate as possible, showing only a minor error in the prediction. This can be achieved by selecting a model that has the largest R square and smallest RMSE values.

2.5.2 Step 2: Process Flowchart

The Process ('P') part of the SIPOC for the dehydration experiment is presented in the form of a system flowchart showing the input-output relationship between the activities of the process (figure 2-6). It also shows how different outputs feed into different steps of the framework. The system flowchart divides the experiment into three sub-processes: Experiment, File compilation, and Spectral analysis.

The first category focuses on the laboratory setup and performance of the experiment on the tailings samples at the laboratory. All spectral files created as a result of the first category are compiled in the file compilation stage. These compiled files are analysed in the last category using statistical tools in Microsoft Excel and the final outputs are thus obtained. The number in the activity boxes depicts the sequence of the activity execution. The information is limited to inputs and outputs at this level of the flowchart. Further details on the activities are explored in the next section.

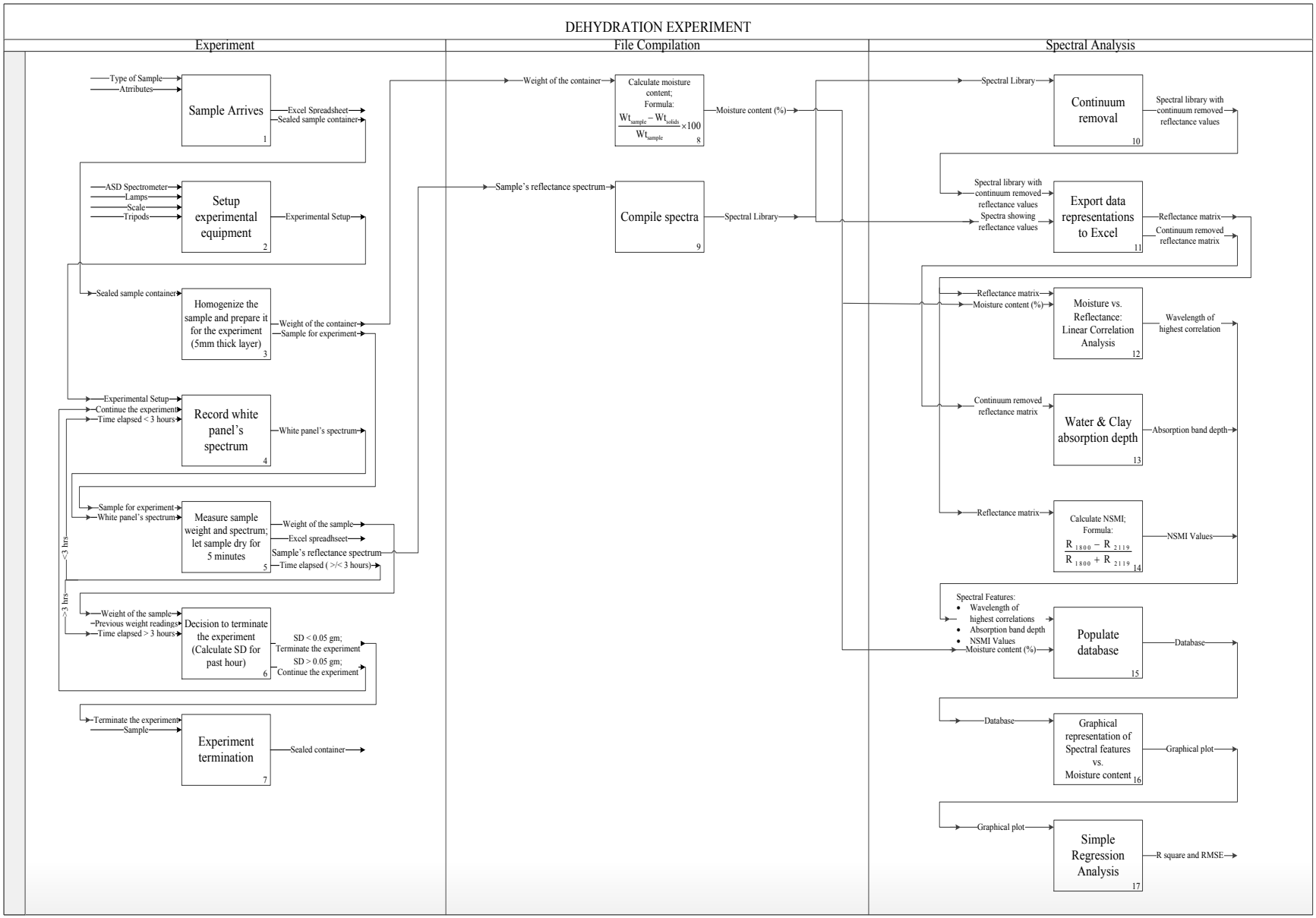

Figure 2-6: Process flowchart of dehydration experiment

2.5.3 Step 3: IDEF0 Model

After creating the process flowchart for the experiment, each activity/step is analysed using the IDEF0 model. The components of this model were discussed in section 2.3. Each activity is also supported by a description of the event and the procedure. Figure 2-7 to Figure 2-25 take the reader through each element of the process flowchart in the sequence of its occurrence.

Fig 2-7 represents the first element of the process.

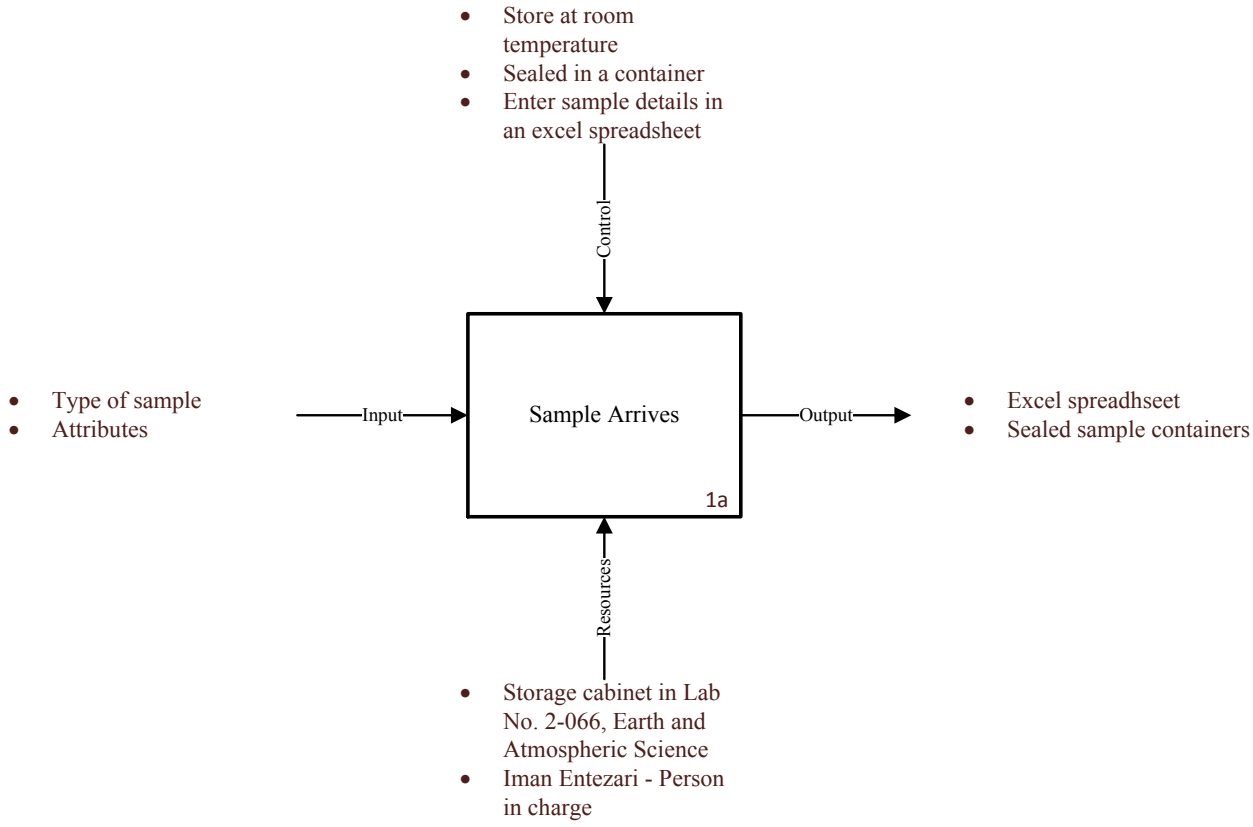

Figure 2-7: IDEFO model for element 1a

Samples of MFT and Tailings Solvent Recovery Unit (TSRU) underflow slurry are brought to the laboratory in marked containers. (The paraffinic froth treatment process produces TSRU tailings that contain a high fraction of asphaltenes. TSRU tailings are thus a very dark material [27].) The samples and their attributes including sample type, sample number, date, % solids,

MBI, and yield stress are logged into an excel spreadsheet. These samples are ultimately sealed in a container and stored at room temperature in an assigned cabinet.

The experiment is setup in the next step (Fig 2-8)

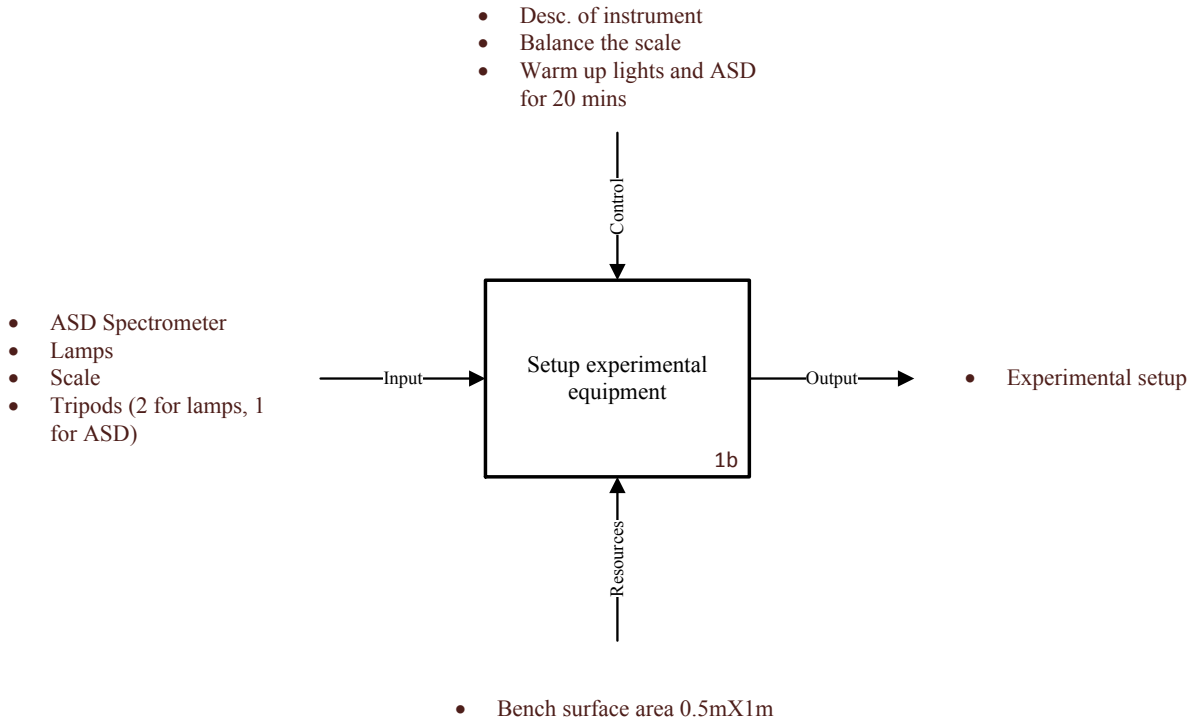

Figure 2-8: IDEF0 model for element 1b

A Panalytic ASD Terraspec Spectrometer positioned at a normal viewing angle to the sample surface [46], two lamps, a balanced scale, three tripods and a bench with the surface area of 0.5m by 1m are needed to start the experiment. Two 50W lamps illuminated the sample at an incidence angle of 30^{o}. Two light sources provided a more uniform illumination on the sample and accelerated the drying process. The sample diameter footprint sensed was approximately 4 cm. [27]. The arranged experimental setup is shown in Figure 2-9.

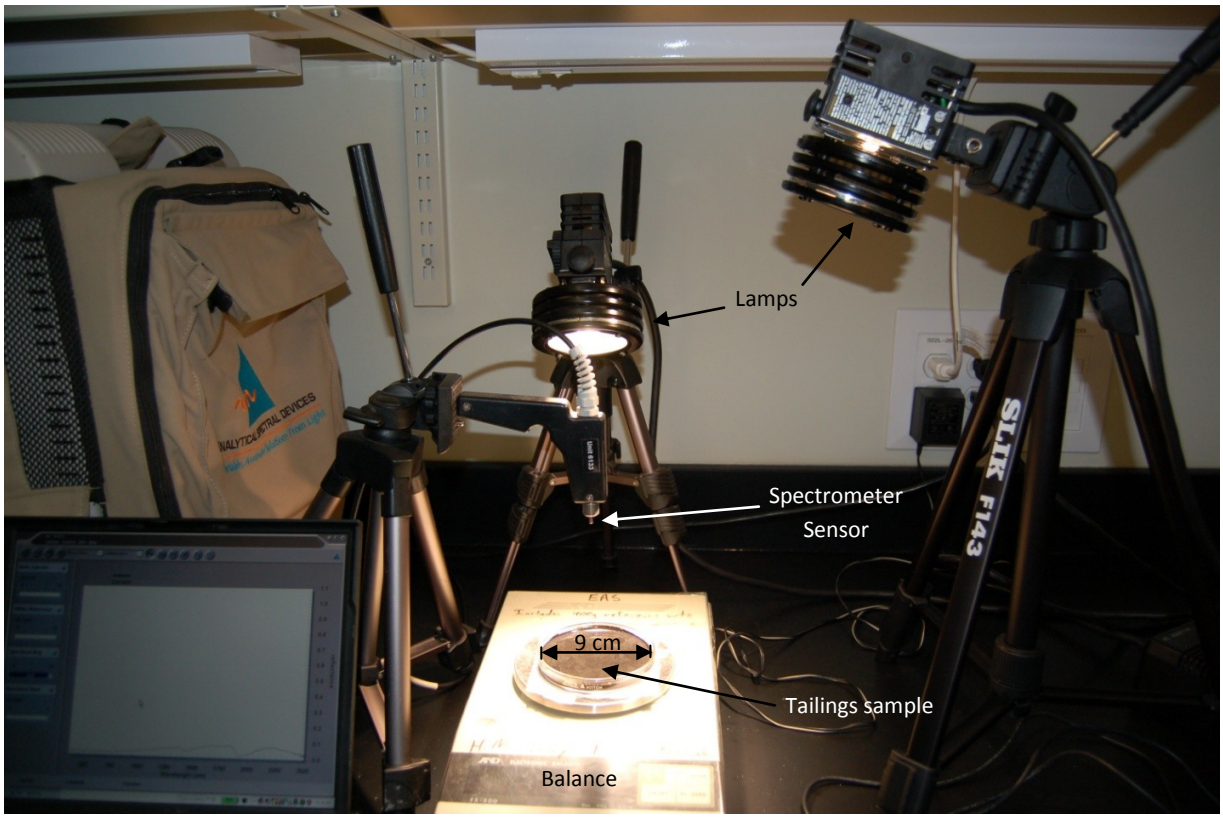

Figure 2-9: Experimental setup for the dehydration experiment, courtesy of Iman Entezari, a PhD candidate at the University of Alberta

Figure 2-10 shows a closer view of TSRU and MFT samples layered on the petri dish.

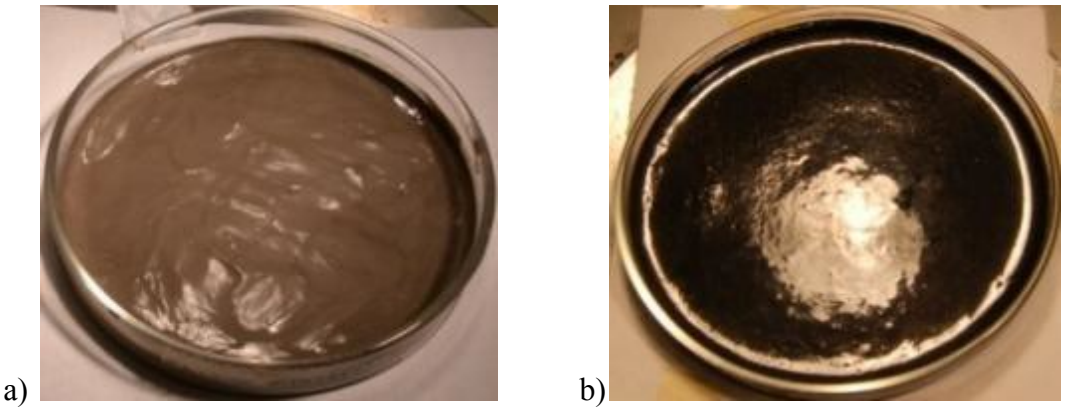

Figure 2-10: a) MFT and b) TSRU sample [27]

Figure 2-11 shows the process of sample preparation.

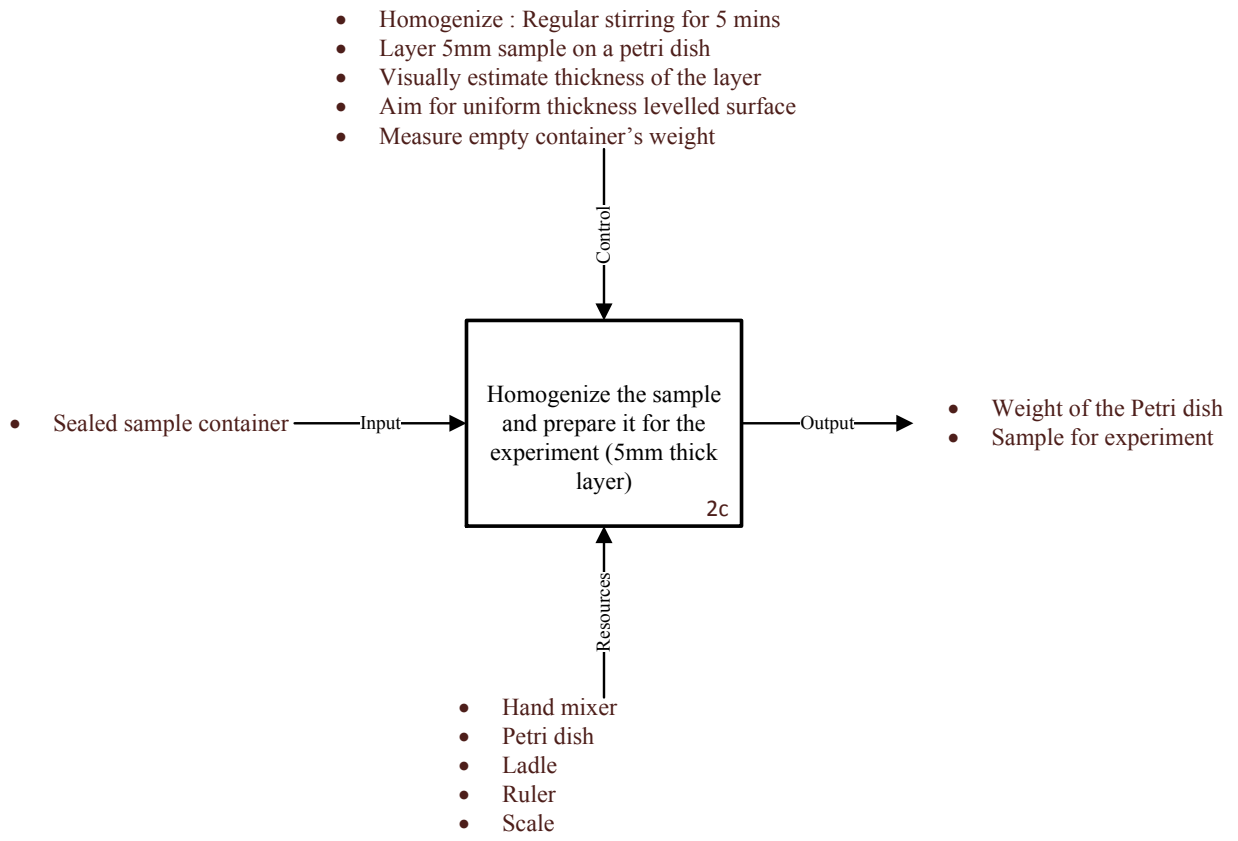

Figure 2-11: IDEF0 model for element 2c

The sample to be observed is stirred in its container for 5 minutes to create a homogenous mixture. It is then carefully scooped using a ladle onto a Petri dish to form a uniform layer of 5 mm thickness (Figure 2-10). The uniformity of this layer is important to target uniform dehydration. The thickness of this layer can be measured using a ruler. Before layering the sample, the weight of the empty Petri dish is measured and recorded on the excel spreadsheet for future reference.

Figure 2-12 represents the fourth element of this process.

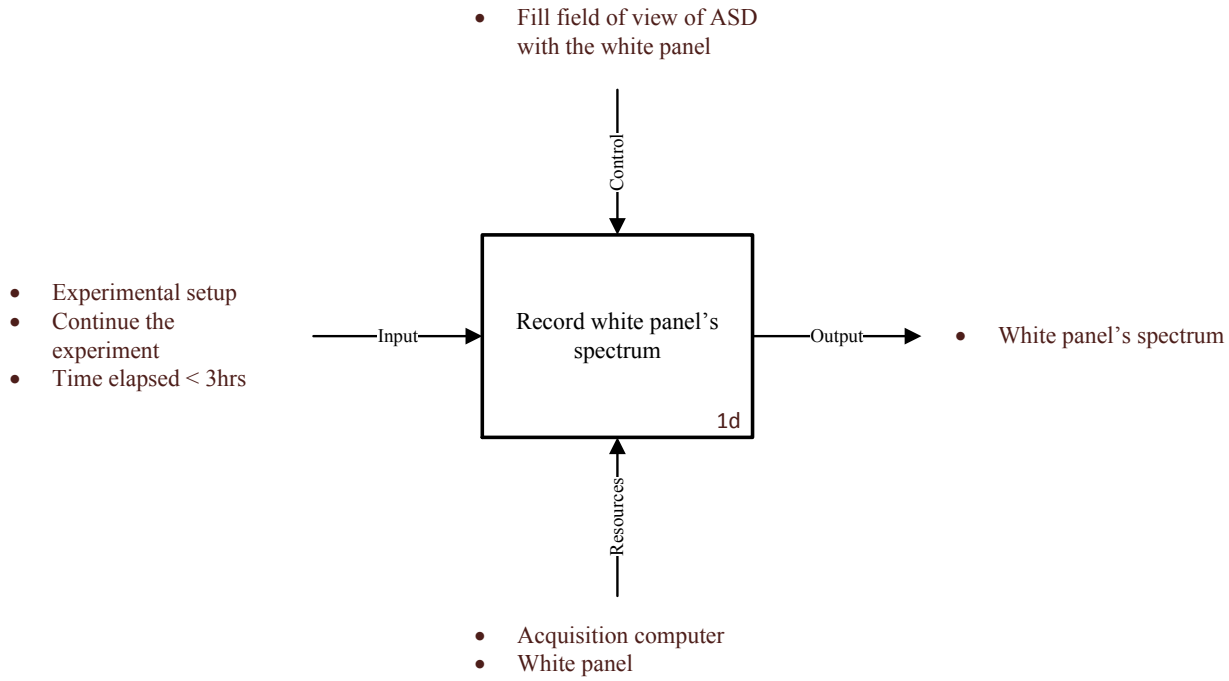

Figure 2-12: IDEF0 model for element 1d

A white panel is used to produce a 99% reflectance spectrum used for normalizing the sample spectrum. Before measuring the sample spectrum at each interval, a white panel spectrum is recorded. The same experimental equipment is used to collect the white panel measurement. The data-acquisition computer connected to the ASD spectrometer records this measurement and uses it to normalise the sample spectrum.

Figure 2-13 expands on the process of taking sample's readings.

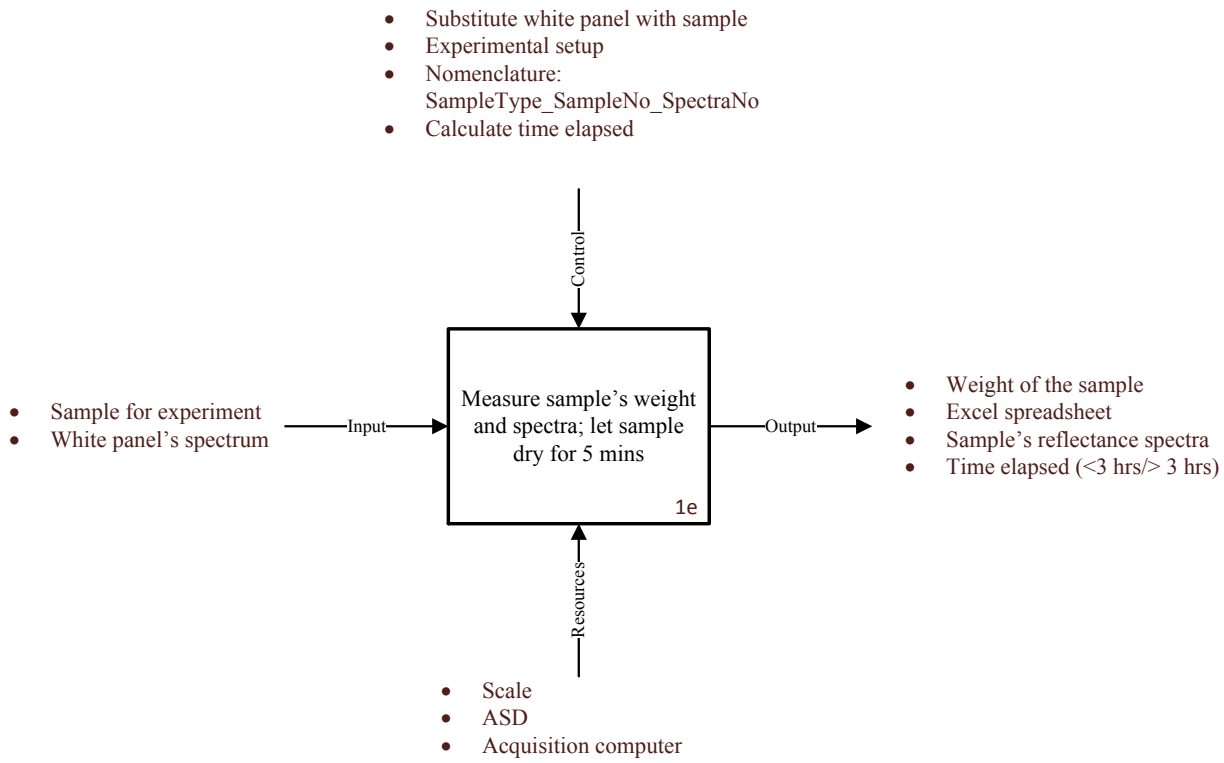

Figure 2-13: IDEF0 model for element 1e

The sample weight and spectrum are collected every 5 minutes until the experiment is terminated. The weight reading is measured using the scale and is recorded in the excel spreadsheet. The sample spectrum is normalised using the white panel spectrum measured a moment ago to provide a sample reflectance spectrum and saved on the data-acquisition computer. The procedure is repeated for 3 hours after which the next step is executed to check whether the experiment needs to be terminated.

It is now time to decide if the experiment needs to be terminated. Figure 2-14 elaborates on the criterion.

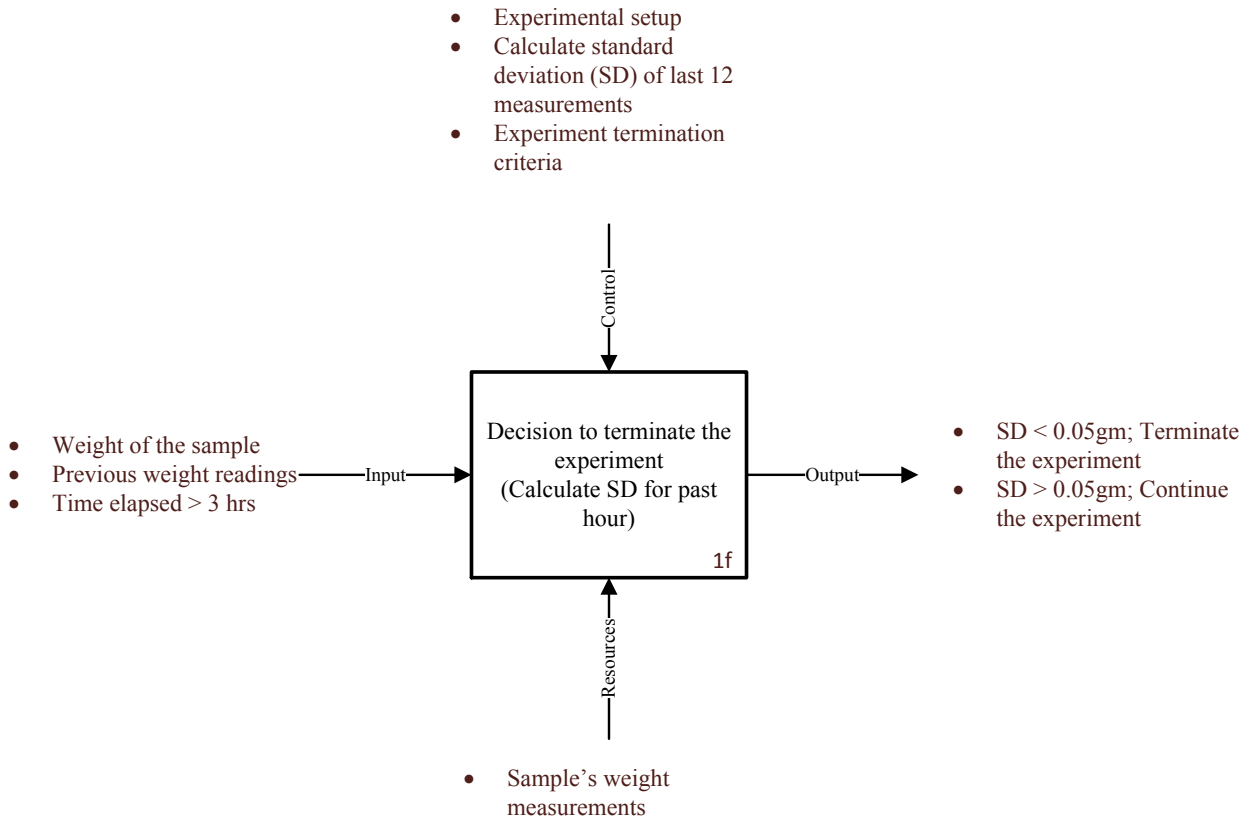

Figure 2-14: IDEF0 model for element 1f

After three hours from the commencement of the experiment or after collecting thirty-six readings, it is time to decide whether the experiment needs to be terminated. For this purpose, the previous twelve sample's weight readings are used to calculate the standard deviation. If this value of standard deviation is less than or equal to 0.05 gm then the experiment is terminated for analysis otherwise the process continues to be repeated every 5 minutes until the criterion for termination is met.

Next, follow the steps outlined in Fig 2-15 after the experiment is terminated.

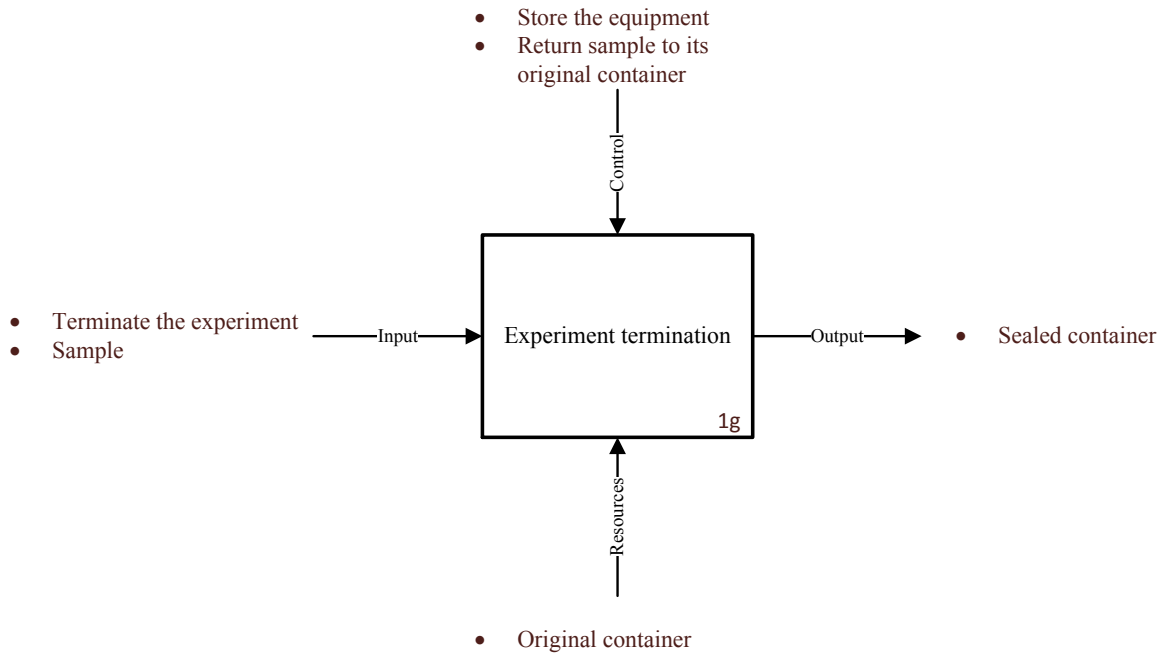

Figure 2-15: IDEF0 model for element 1g

After terminating the experiment, the dried sample is returned to its original container. The sample is stirred before sealing the container. The experimental equipment is cleaned and stored in the cabinet. All the data recorded is carefully saved on the acquisition computer before it can be compiled for analysis.

Figure 2-16 shows the moisture content calculation procedure.

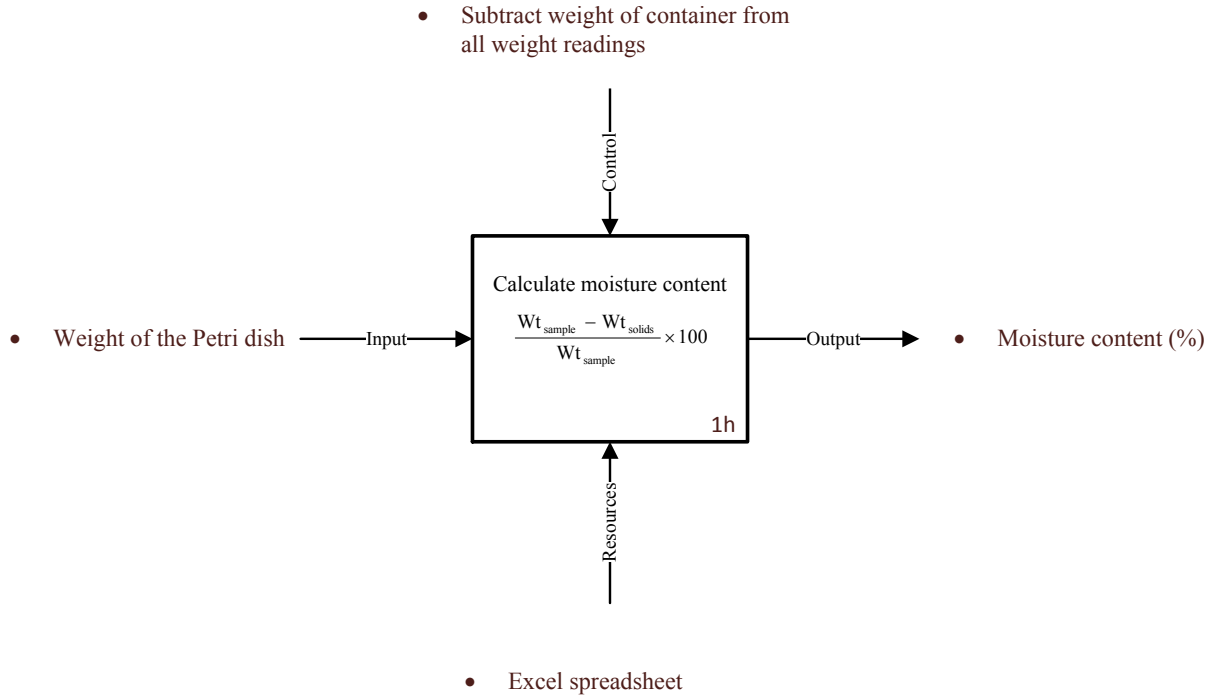

Figure 2-16 : IDEF0 model for element 1h

The moisture content percentage is calculated for each interval of the experiment, using the formula provided in the block. Wt_{sample} is the weight of sample at each interval as recorded on the scale less the weight of the empty Petri dish. Wt_{solids} refers to the weight of the sample as measured on the scale at zero percent moisture content less the weight of the empty Petri dish. The moisture content (% by weight) is saved in an excel spreadsheet.

Figure 2-17 shows the spectra compilaton process.

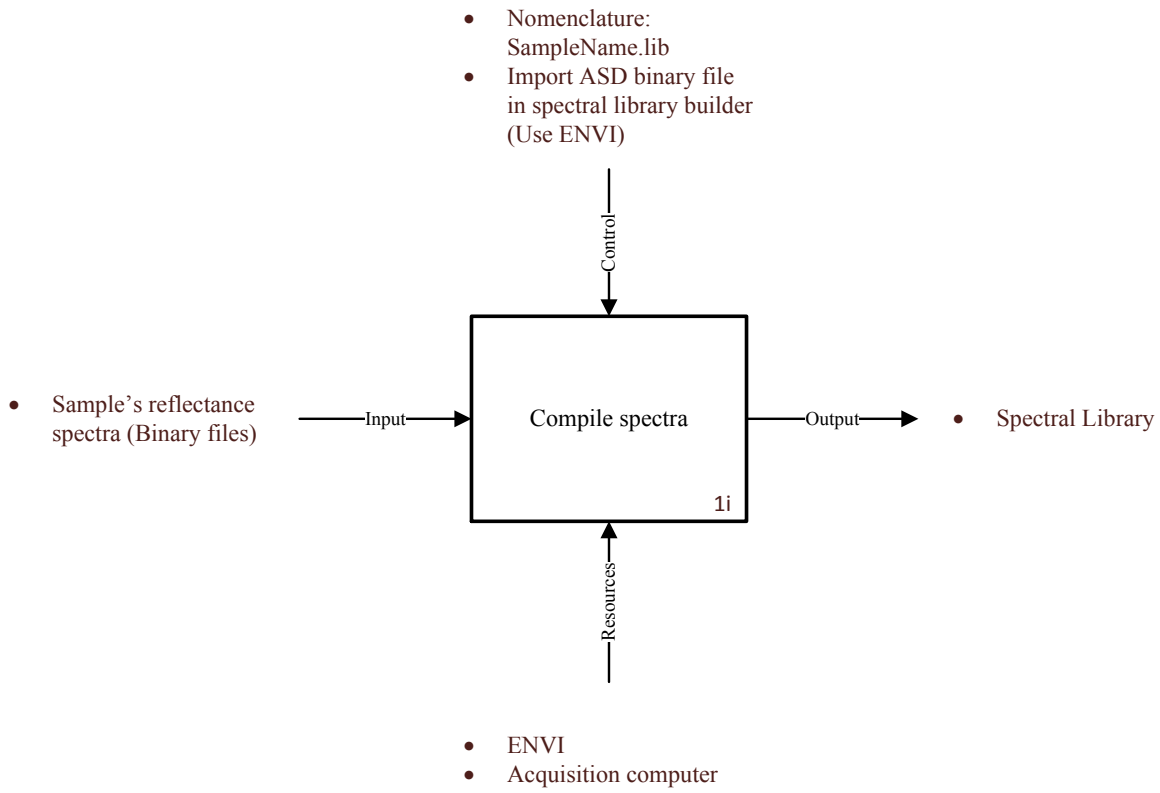

Figure 2-17: IDEF0 model for element 1i

All binary spectra files stored in the acquisition computer during the experiment are compiled using import function in ENVI to create a spectral library of the sample. A specific nomenclature "samplename.lib" is used for this spectral library.

The next step is to remove spectral continuum from each spectrum in the spectral library by using the continuum removal function in ENVI.

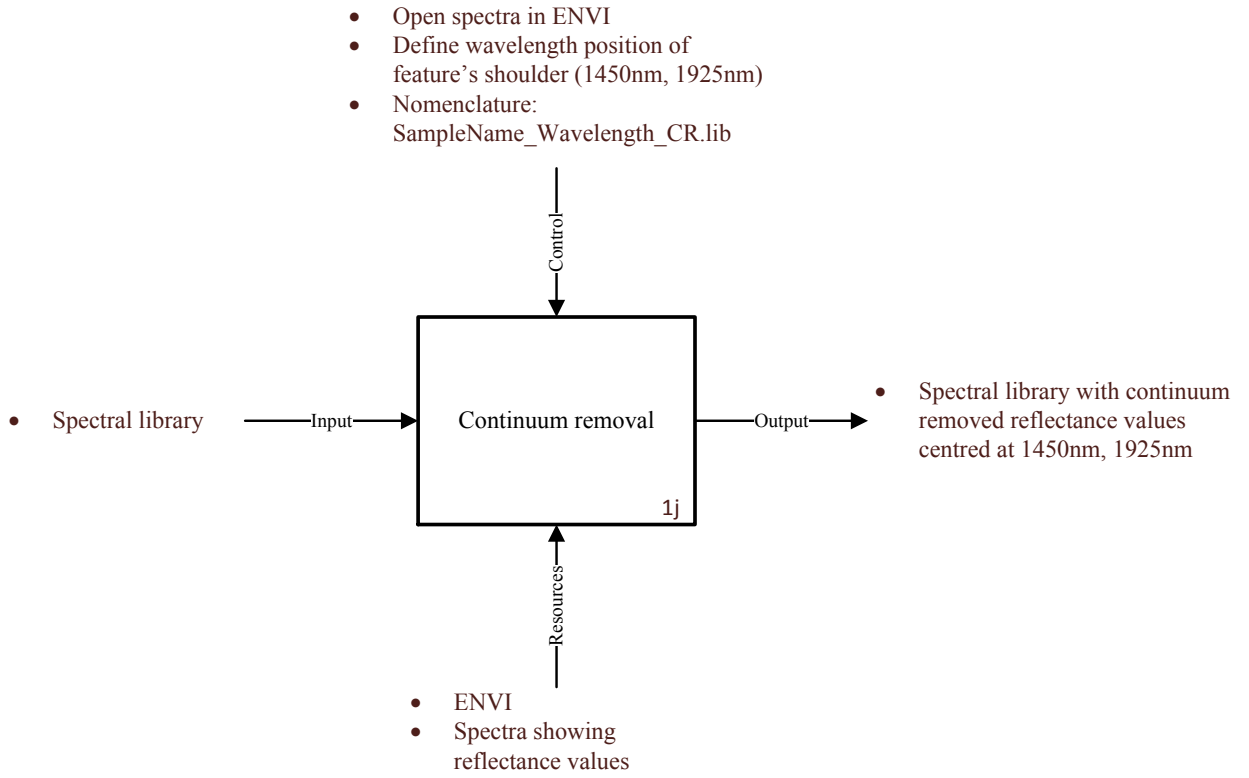

Figure 2-18: IDEF0 model for element 1j

The first step is to carefully look for the nearest left and right shoulders to the selected feature wavelength range, 1450nm and 1925nm in this case. The output is a spectral library (Samplename_wavelength_CR.lib) containing spectra with continuum removed reflectance values centred at 1450nm and 1925 nm.

Spectral data collected so far is exported to Excel in the next step. (Figure 2-19).

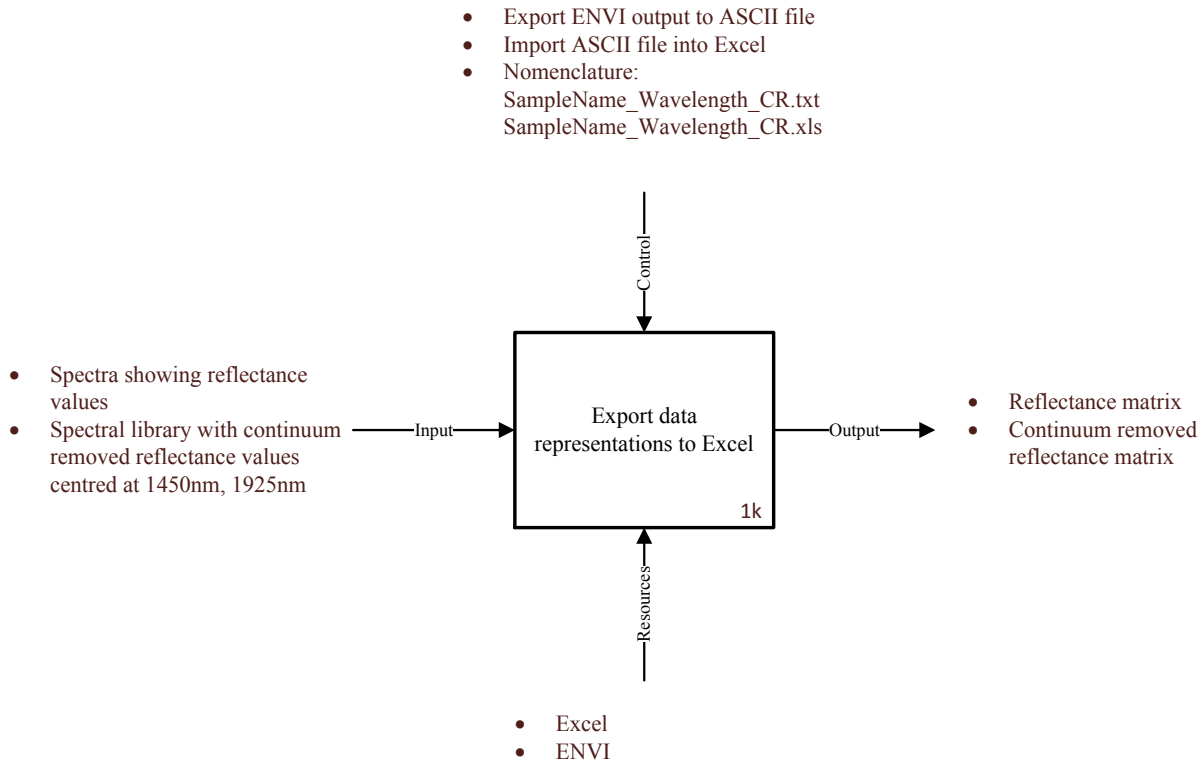

Figure 2-19: IDEF0 model for element 1k

Open the original spectral library and continuum removed spectral library in ENVI. Export the reflectance values to an ASCII file (SampleName_wavelenght_CR.txt) and then import them to an Excel file (SampleName_Wavelength_CR.xls). The output is a matrix that shows reflectance values for selected wavelength range and for all levels of moisture content.

Linear correlation analysis is performed in the next step using Figure 2-20's guided process.

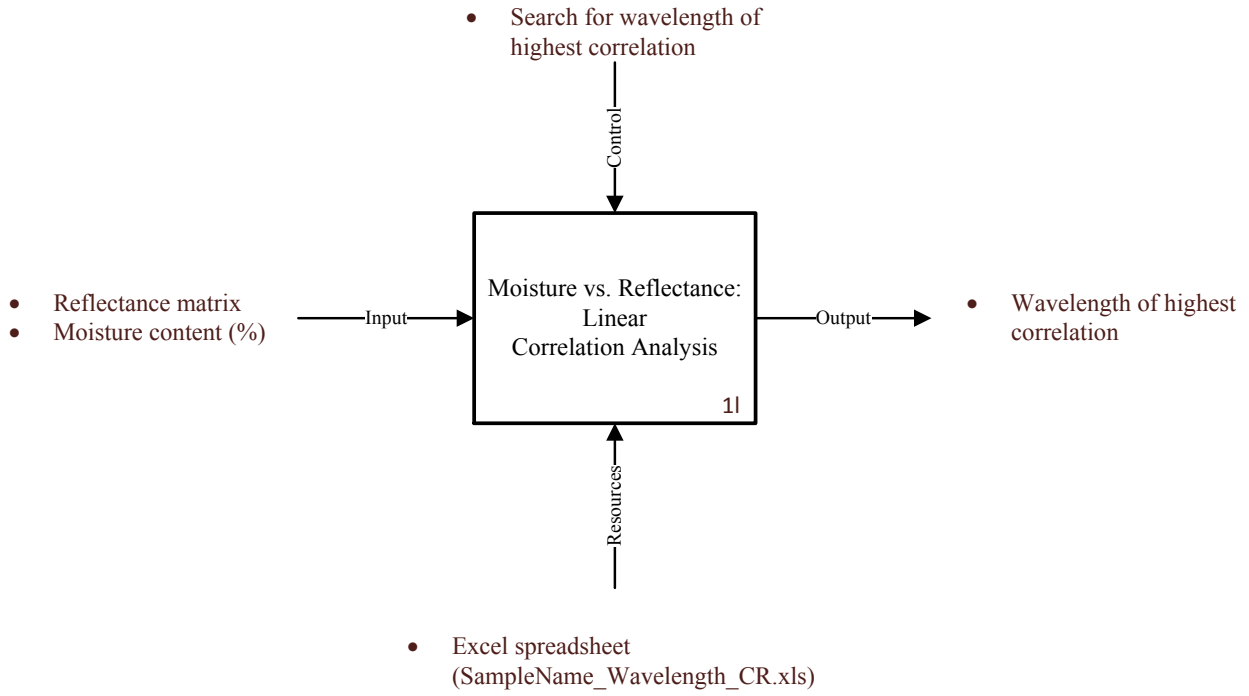

Figure 2-20: IDEF0 model for element 1l

To compare reflectance with moisture content level, a wavelength of highest correlation needs to be determined. This can be achieved by using a linear correlation analysis approach in excel. The output is saved in the same excel spreadsheet (SampleName_Wavelength_CR.xls).

Figure 2-21 shows the next step of this process.

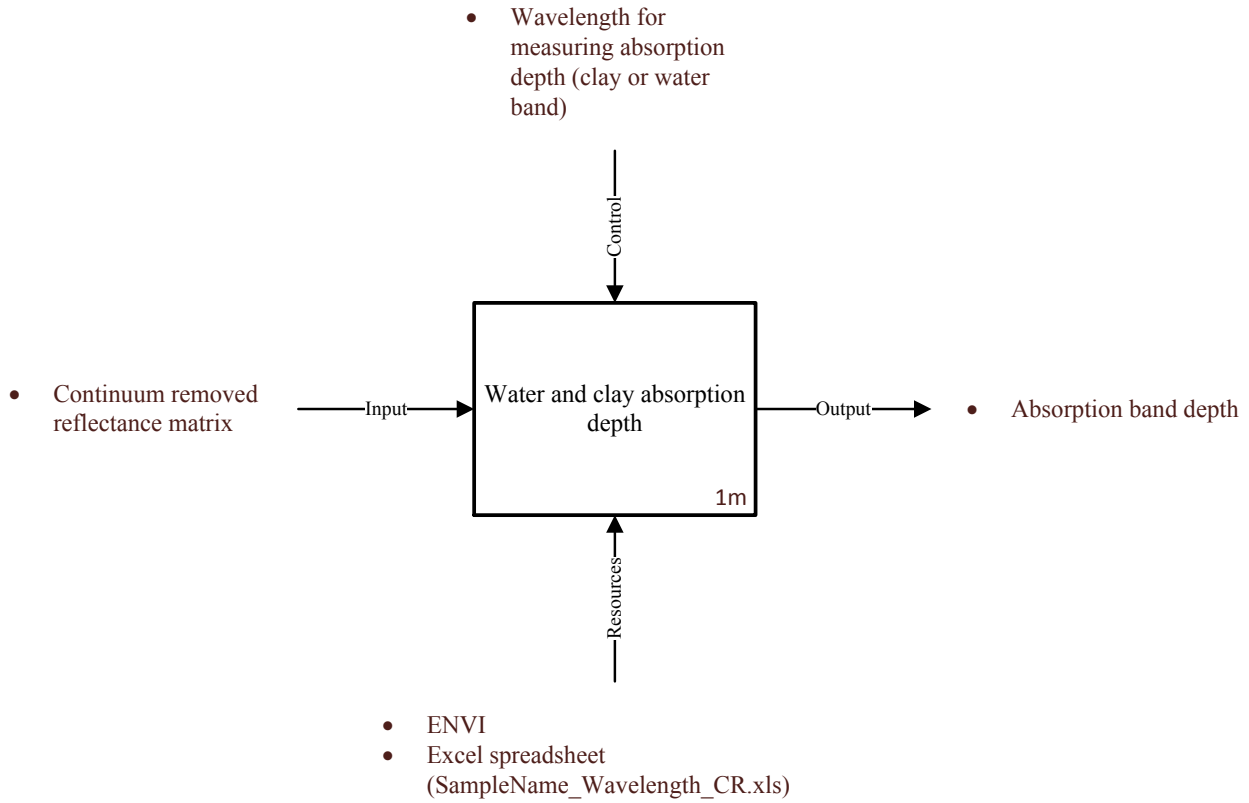

Figure 2-21: IDEF0 model for element 1m

Another important spectral feature is the absorption band depth for the water and clay features. Wavelength bands are used to the measure absorption band depth on continuum removed reflectance matrix.

The normalised soil moisture index (NSMI) values are also calculated and compared with the moisture content level of the sample as shown in figure 2-22.

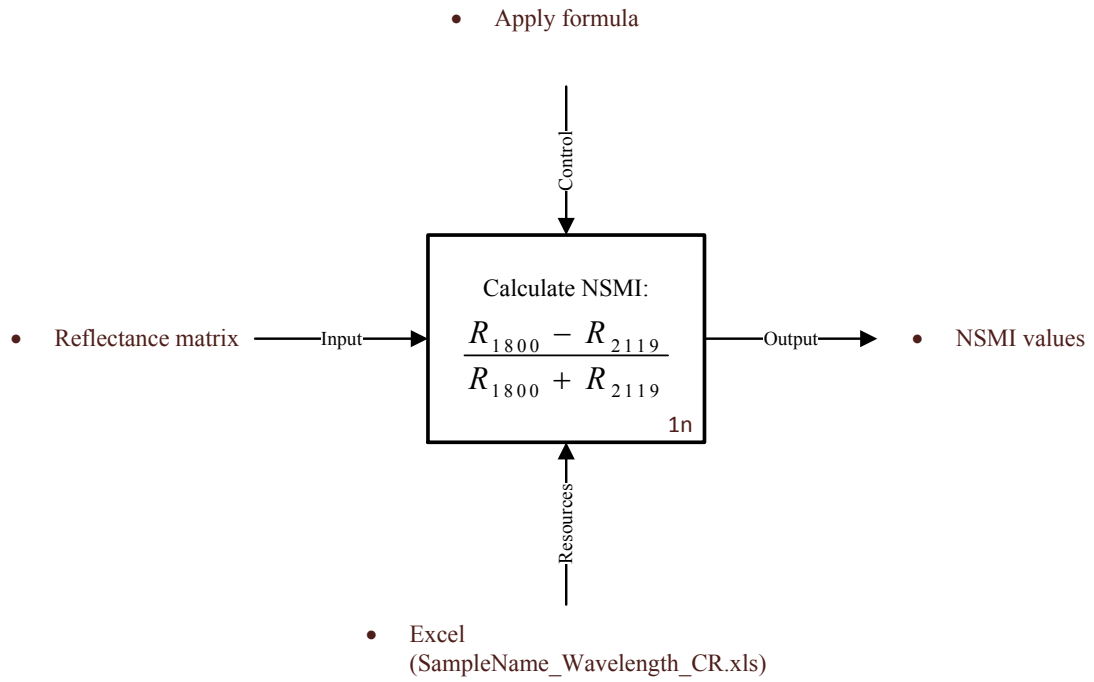

Figure 2-22: IDEF0 model for element 1n

The simple formula shown in the block diagram (Figure 2-22) is used to calculate these NSMI values from the reflectance matrix. The results are saved in the same excel spreadsheet (SampleName_Wavelength_CR.xls).

A spectral matrix is populated next as described by figure 2-23.

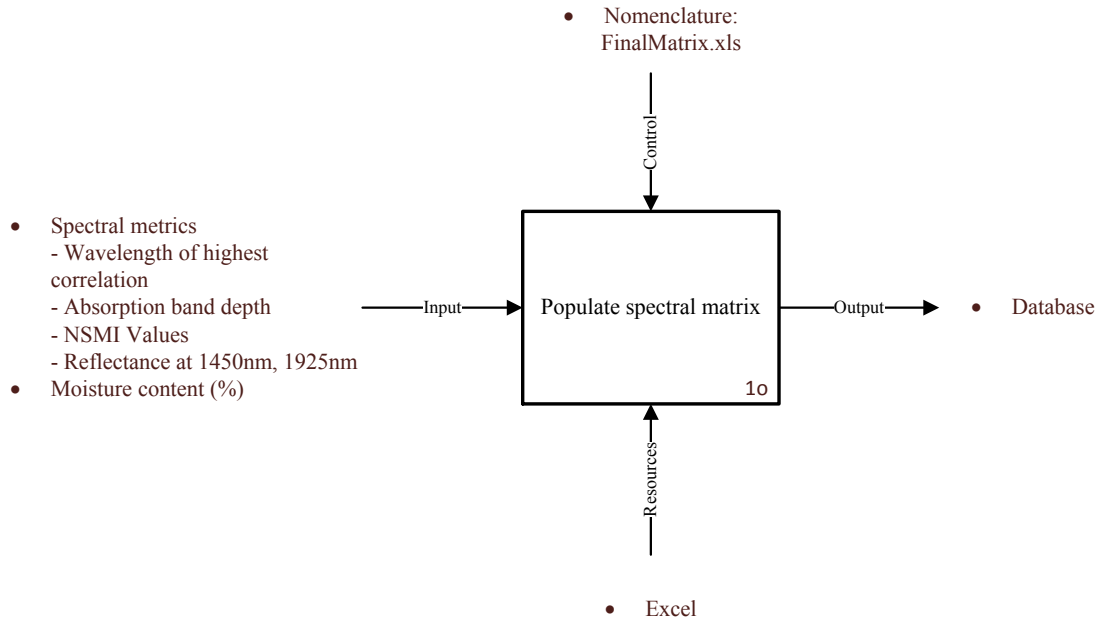

Figure 2-23: IDEF0 model for element 1o

All spectral features: wavelength of highest correlation, absorption band depth, NSMI values and reflectance at 1450nm and 1925nm are tabulated against moisture content percentage to create a database in excel (FinalMatrix.xls).

Graphical representations are constructed in the next step (Figure 2-24).

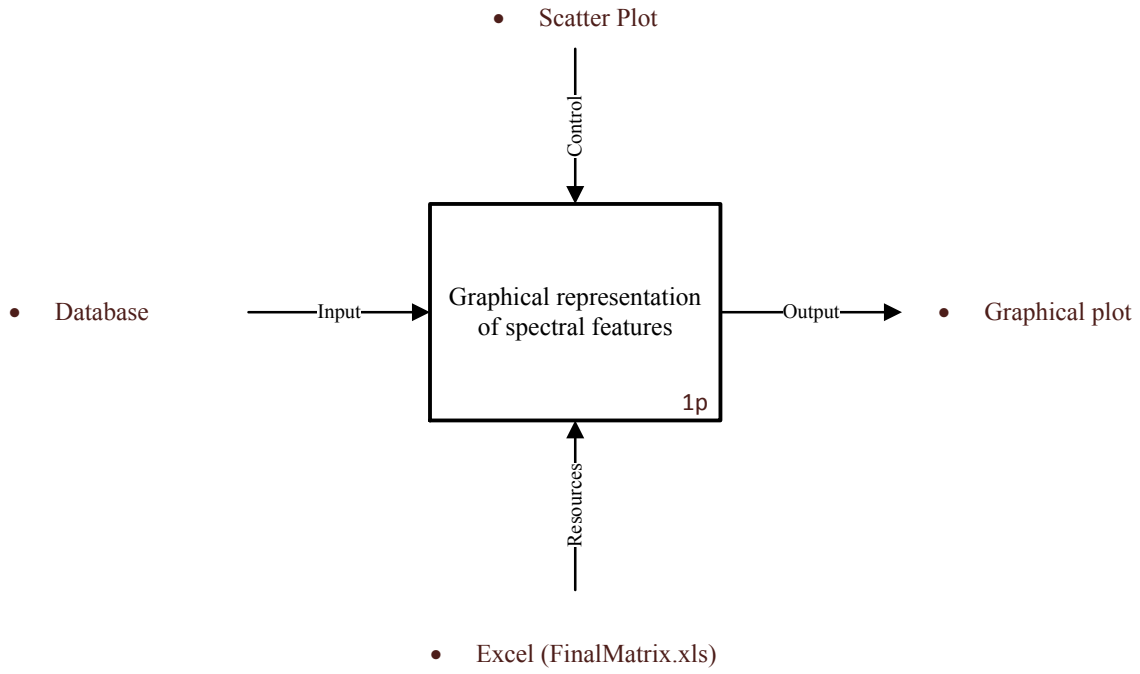

Figure 2-24: IDEF0 model for element 1p

The database created in the last step is now used to create different scatter plots for reflectance vs. moisture content, absorption depth vs. moisture content, and NSMI vs. moisture content.

Lastly, a regression analysis is performed in Excel (Figure 2-25).

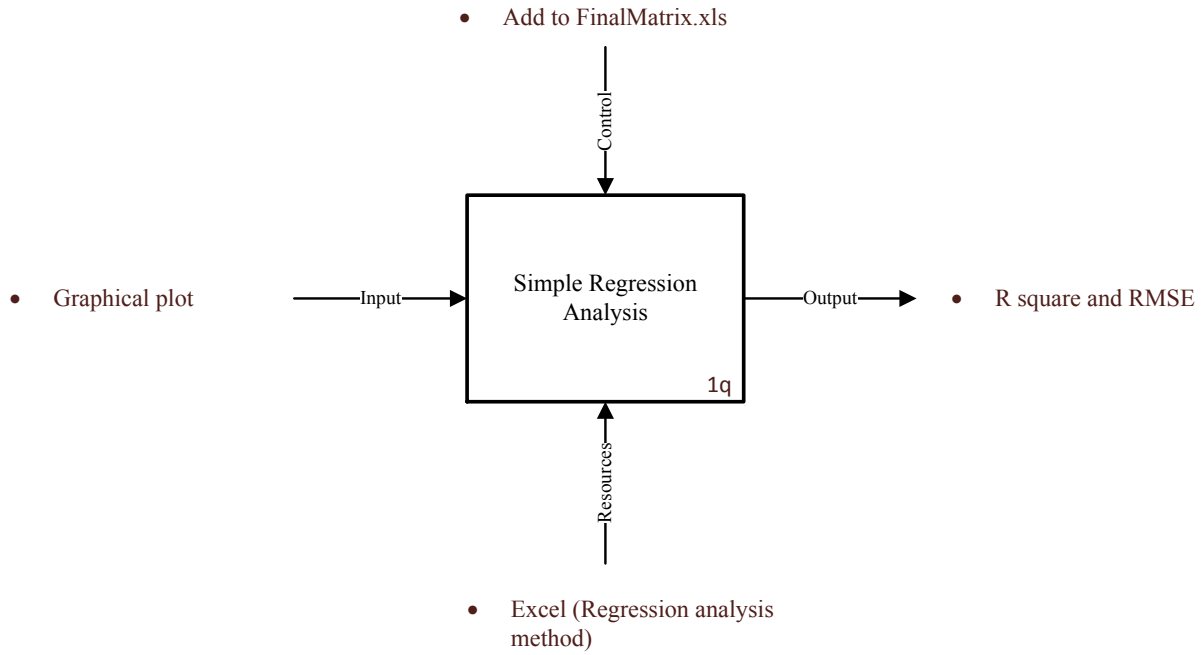

Figure 2-25: IDEF0 model for element 1q

Regression analysis is used to calculate statistical parameters such as the coefficient of determination (R^2) and the root mean square error (RMSE) for all the graphical plots. The R^2 and RMSE values are hence used to determine the most robust model.

2.5.4 Step 4: Process Value Analysis

A qualitative analysis of the dehydration process was done using Table 2-1. It shows the absence of non-value added activities in the process and represents where the value is to the customer. Based on this analysis, it was realised that each activity that goes towards the completion of this experiment holds a value where the absence of even a single one would interfere with the success of the experiment, that is, there is no redundant step. The input-output relationship flowchart also supports the conclusion as it shows that the steps are sequential and the outputs from different activities move to the different steps in the framework.

Activity	Customer Value Added (Y/N)	Operational Value Added (Y/N)	Non-Value Added (Y/N)
Sample Arrives	N	Y	N
Setup experimental equipment	N	Y	N
Homogenize the sample and prepare it for the experiment	N	Y	N
Record white panel's spectrum	N	Y	N
Measure sample weight and spectrum; let the sample dry	N	Y	N
Decision to terminate the experiment	N	Y	N
Experiment termination	N	Y	N
Calculate moisture content	N	Y	N
Compile spectra	N	Y	N
Continuum removal	N	Y	N
Export data representations to Excel	N	Y	N
Moisture vs. Reflectance: Linear correlation analysis	Y	N	N
Water and Clay absorption depth	Y	N	N
Calculate NSMI	Y	N	N
Populate database	Y	N	N
Graphical representations of spectral features vs. moisture content	Y	N	N
Simple regression analysis	Y	N	N

Table 2-1: Process value analysis of the dehydration experiment

2.6 Case Study 2: Evaporation Experiment

Similar to the dehydration experiment, the evaporation experiment is used to derive a predictive model to estimate moisture content. The evaporation experiment also uses evaporation fluxes from tailing surfaces to calculate the model result. Hyperspectral time-series laboratory observations are collected from MFT samples that are allowed to evaporate from an initial state of water saturation to an air-dried state. From these data, several spectral features are evaluated to predict water content and the normalised evaporation rate [28]. The same methodology as case study 1 is followed to study this experiment using the three-step procedure.

2.6.1 Step 1 : SIPOC Model

The analysis starts by framing a SIPOC model for the experiment as shown in figure 2-26. The oil sands company supplies the MFT samples for the experiment with its number, date, % solids,

MBI and yield stress. The ASD Spectrometer needed for the experiment is supplied by Panalytic. The experimental equipments required is obtained from the department of Earth and Atmospheric Sciences at the University of Alberta. The process of the evaporation experiment is detailed using the flowchart shown in Step 2. Behavioral graphs of spectral metrics with respect to the normalised evaporation and moisture content are produced to create a robust predictive model. The statistical parameters R^2 and RMSE are calculated to measure error in the model. The results are shared with the oil sands company partner and the IOSI team.

There are also a couple requirements that need to be met. The customer requirement is that the final model should show minimal errors in prediction. Secondly, the MFT samples used for the experiment should span a range of MBI to assess the sensitivity in behaviour. The next step is to map the experimental process to identify need for improvement.

EVAPORATION EXPERIMENT				
SUPPLIERS	**INPUTS**	**PROCESS**	**OUTPUTS**	**CUSTOMERS**
• An oil sands company • Panalytic • Department of Earth and Atmospheric Sciences at the University of Alberta	• Samples • Characteristic Sheet from oil sands company - Sample No. - Date - % solids - MBI - Yield Stress • ASD Spectrometer • Equipment	Evaporation experiment	• Behaviour of different spectral metrics with respect to moisture content and normalised evaporation • A robust predictive model • R square and RMSE values	• IOSI Team
REQUIREMENTS: • Samples should have a wide range of MBI characteristics to get accurate results • Predictive model produced should be as accurate as possible, ensuring minimum errors				

Figure 2-26: SIPOC model of evaporation experiment

2.6.2 Step 2: Process Flowchart

At this stage, the evaporation experiment procedure is depicted in the form of a flowchart that shows the input-output relationship between different activities of the process. The system

flowchart is shown in figure 2-27. The three sections represent three stages of the procedure: experimental stage, file compilation stage and the file analysis stage. At the first stage, samples are brought to the laboratory, prepared for the experiment, and observed to collect data. At the file compilation stage, the collected data is compiled to perform the analyses. The file analysis stage, as the name suggests, mainly comprises statistical analysis on the compiled files to obtain desired outputs. The analyses are done in Microsoft Excel using some of its popular features such as Linear correlation, and Regression Analysis. The process map guides the reader through the sequence of activities and how the element output flows through the flowchart.

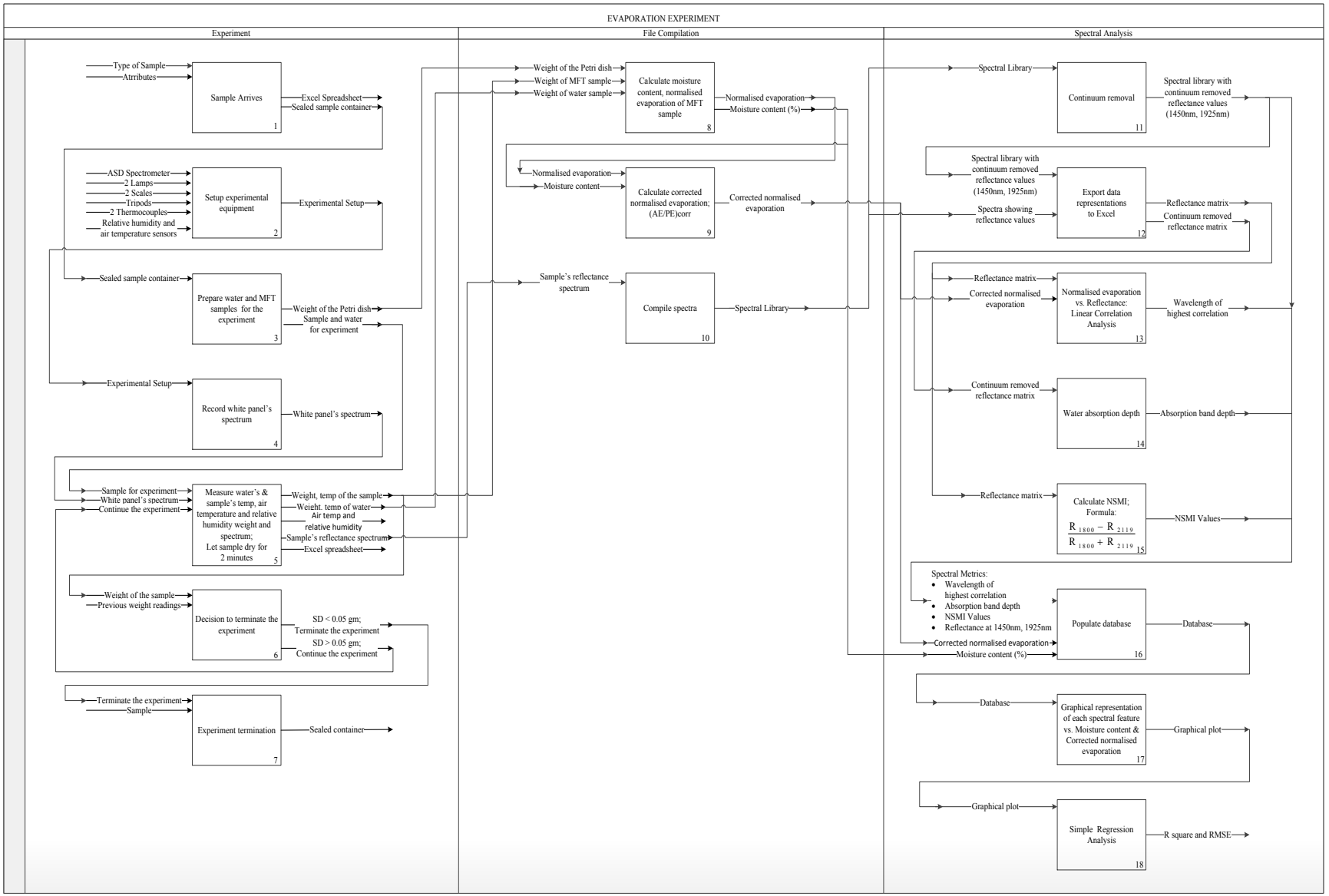

Figure 2-27: Process flowchart of evaporation experiment

2.6.3 Stage 3: IDEF0 Model

The system flowchart for the evaporation experiment is now broken down using the IDEF0 model and each activity is supported by a description of the actions involved. Figure 2-28 to figure 2-47 carry the reader through the sequence of the evaporation experiment activities.

Figure 2-28 represents the first step of this process.

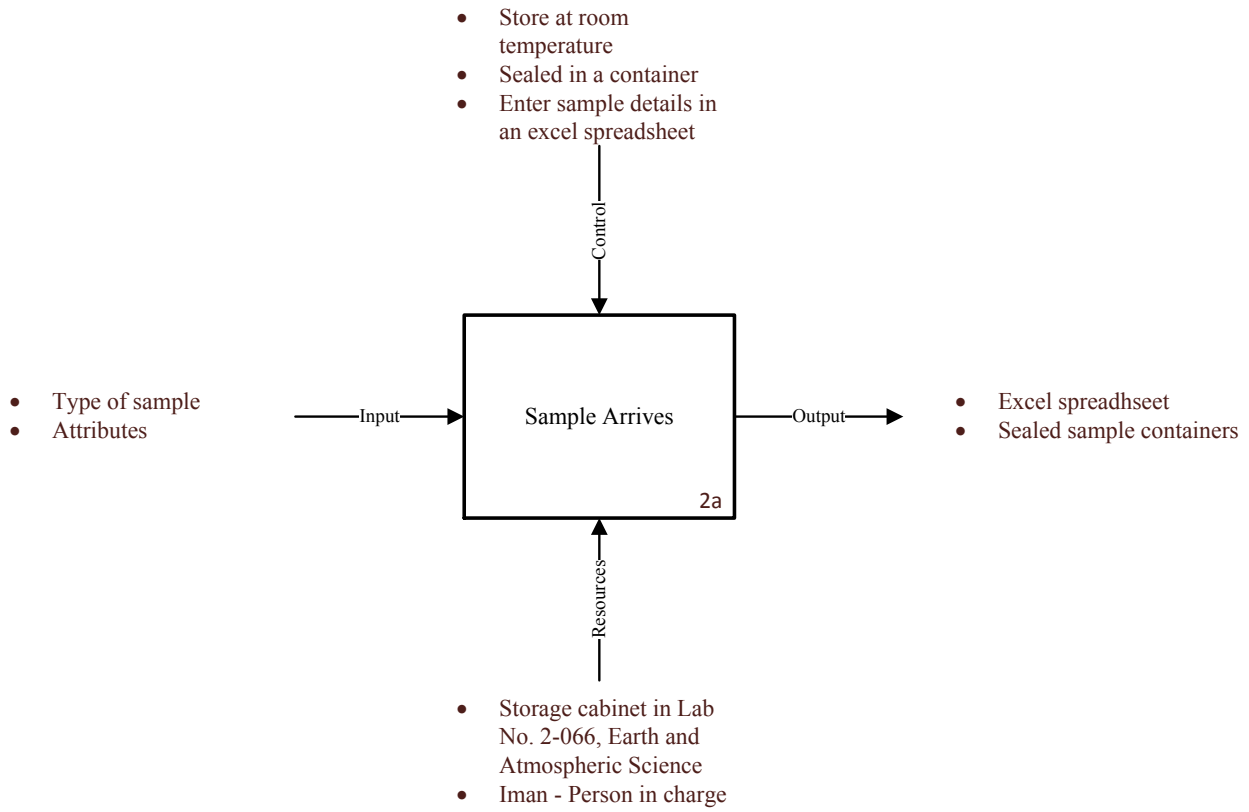

Figure 2-28: IDEF0 model for element 2a

When the sample(s) arrives at the lab, its attributes are recorded in an excel sheet. The sample is sealed in a container and stored at room temperature in the designated cabinet at the Earth and Atmospheric Science Lab of the University of Alberta.

Figure 2-29 describes experiment setup procedure.

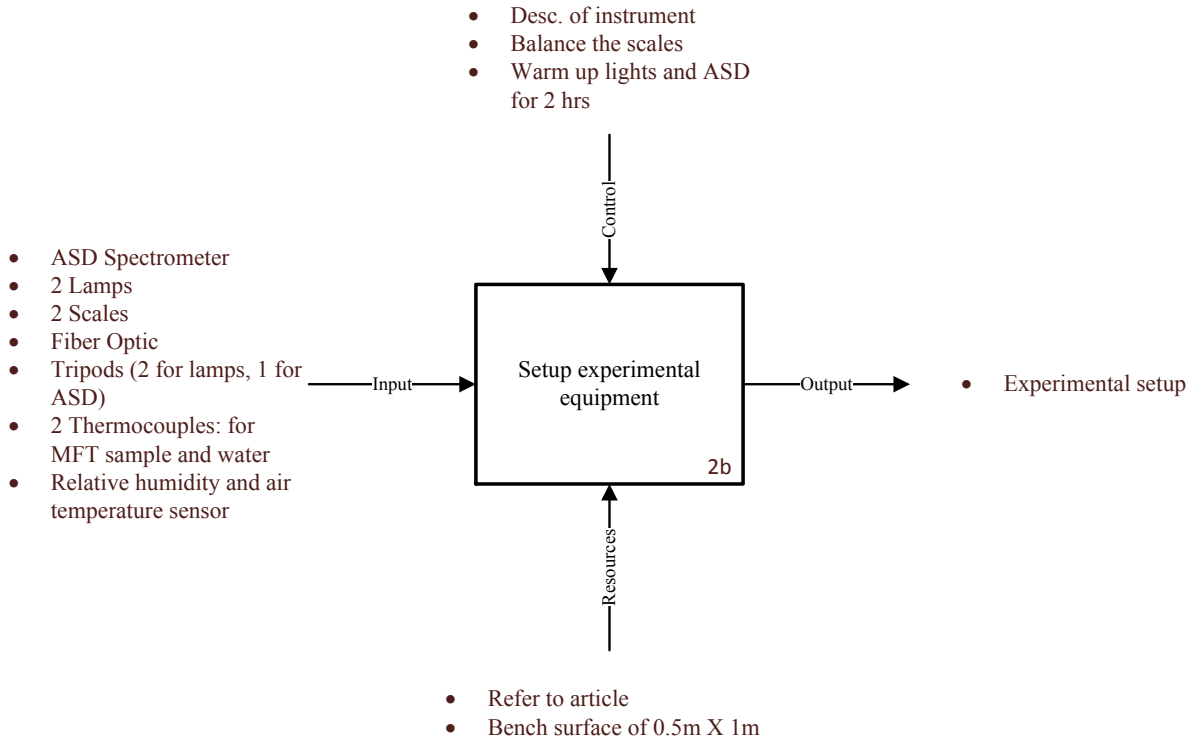

Figure 2-29: IDEF0 model for element 2b

Similar to the moisture content experiment, the equipment is set up but this time two polystyrene Petri dish each measuring a 150 mm diameter are used, one hosting the MFT sample and the other one hosting water. Two identical 50W quartz halogen lamps position at an incidence angle of approximately 20^{o} are used to illuminate each container, and two thermocouples are embedded in the perimeter of each Petri dish to monitor the temperature of the sample and water. Humidity and temperature sensors are used to measure air temperature and relative humidity. The ASD spectrometer is pointing at a normal viewing angle to the sample surface and used to collect spectra. More details on the procedure are mentioned in Entezari et al.,[28]. The

equipment is set up and halogen lights and the ASD instrument are warmed up for two hours before the experiment can start. Figure 2-30 shows the complete experimental setup.

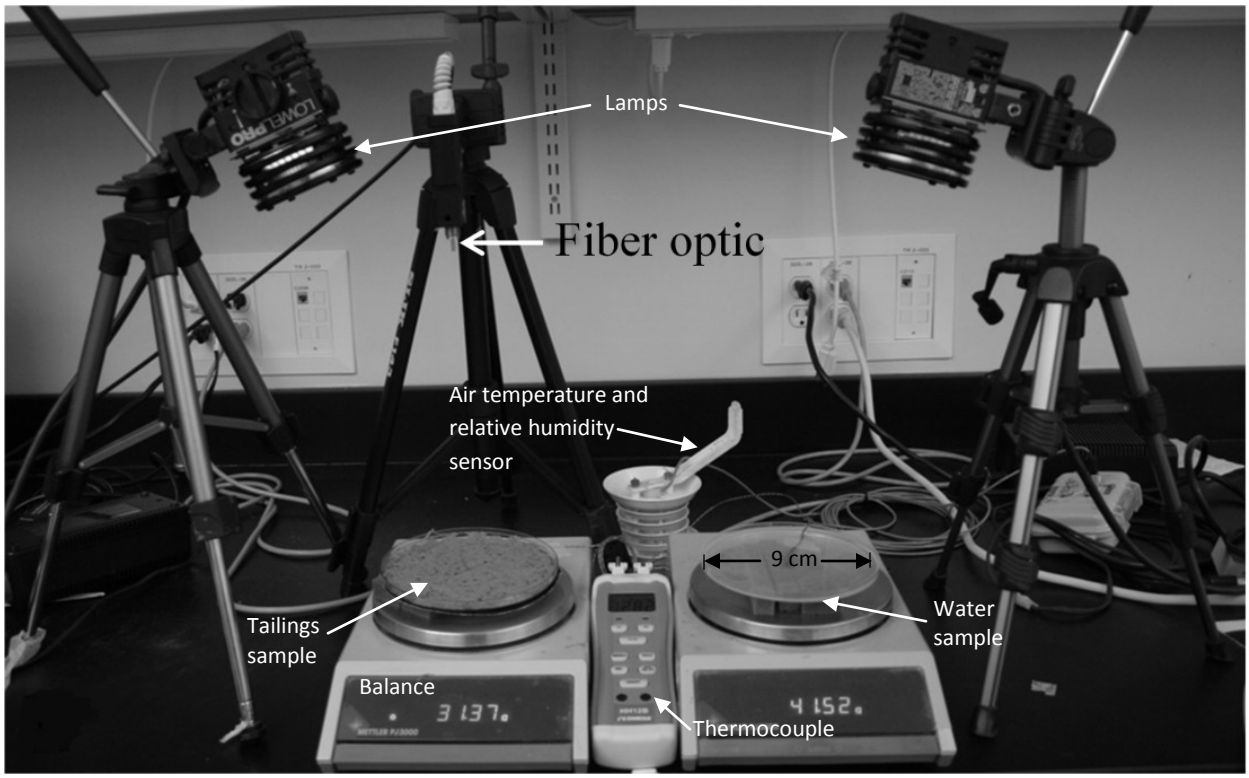

Figure 2-30: Experimental Setup [28]

Figure 2-31 describes the steps to prepare sample for the experiment.

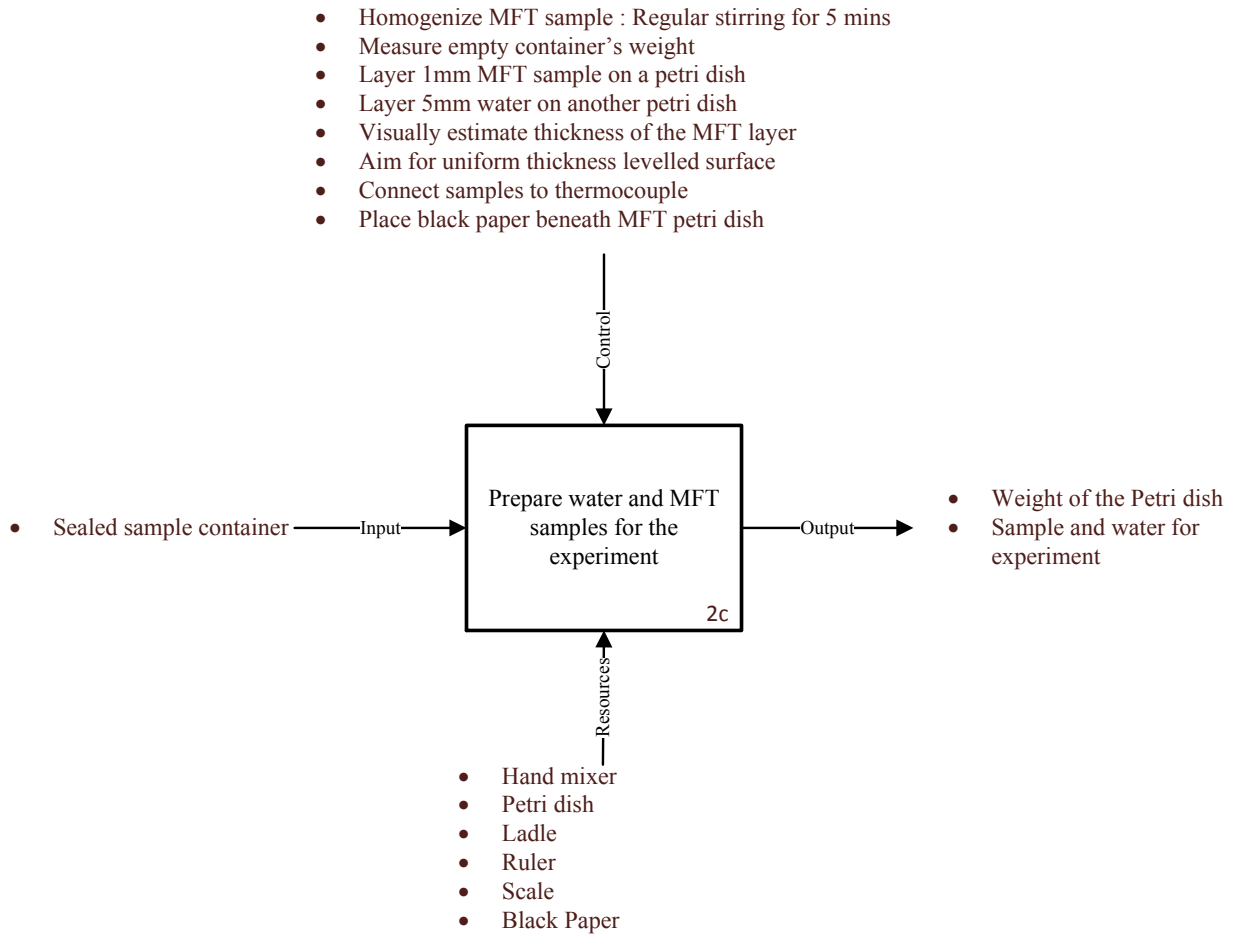

Figure 2-31: IDEF0 model for element 2c

Record the weight of the empty Petri dish for future calculations. Homogenize the sample by stirring it in the container for about 5 minutes and scoop a 1 mm uniform layer of the sample in the Petri dish. The MFT samples were made as thin as possible due to two reasons: 1) to minimize the effect of the water content below the evaporating surface as the soil properties at the soil surface controls the evaporation characteristics, and 2) to minimize decoupling between spectral measurements and the tailings bulk properties, as the reflectance is a surface phenomenon collected from the surface within a few hundred microns in depth [28]. In another

Petri dish pour a 5 mm layer of water. Place a black paper under the MFT sample, connect both samples to thermocouple and proceed to begin the experiment in the next step.

Measurements of the sample spectra need to be normalised with the measurements of the white panel as described in figure 2-32.

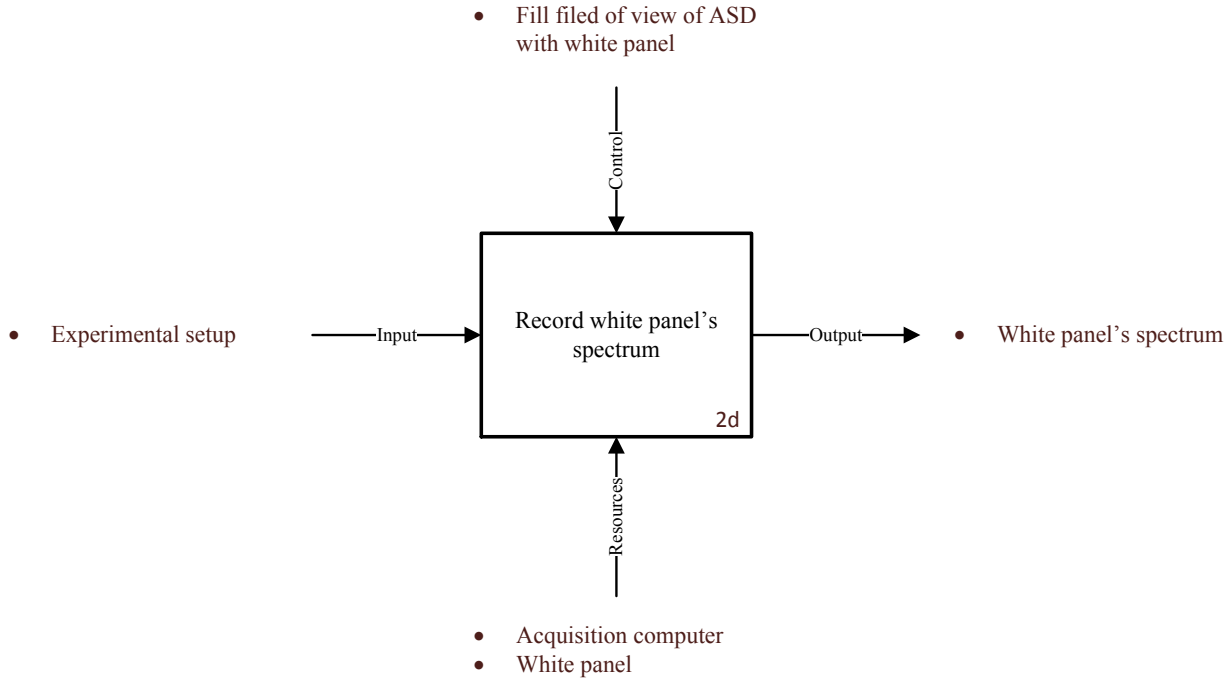

Figure 2-32: IDEF0 model for element 2d

First, a white panel spectrum is recorded. Use the same experimental setup and fill the field of view of the ASD with the white panel and record its spectrum on the acquisition computer.

At the next step, water's and MFT sample's readings are measured (Figure 2-33).

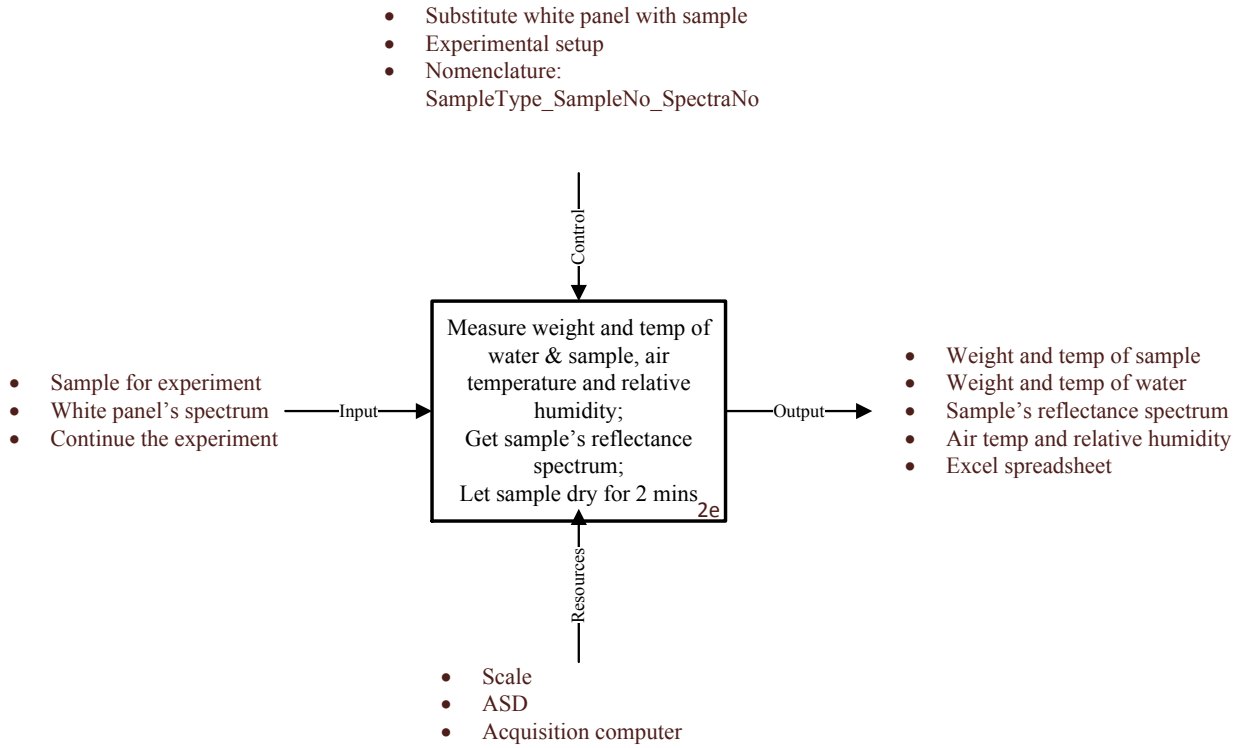

Figure 2-33: IDEF0 model for element 2e

Substitute the white panel with the sample and measure its spectrum. Simultaneously measure the water and sample weight and temperature, the air temperature and relative humidity and record on an excel spreadsheet. The sample reflectance spectrum after normalization to the spectrum of the white panel is saved on the data-acquisition computer. The sample is air dried for 2 minutes and the next round of readings starts.

Using the guidelines shown in figure 2-34, it is decided whether to terminate the experiment or to continue it.

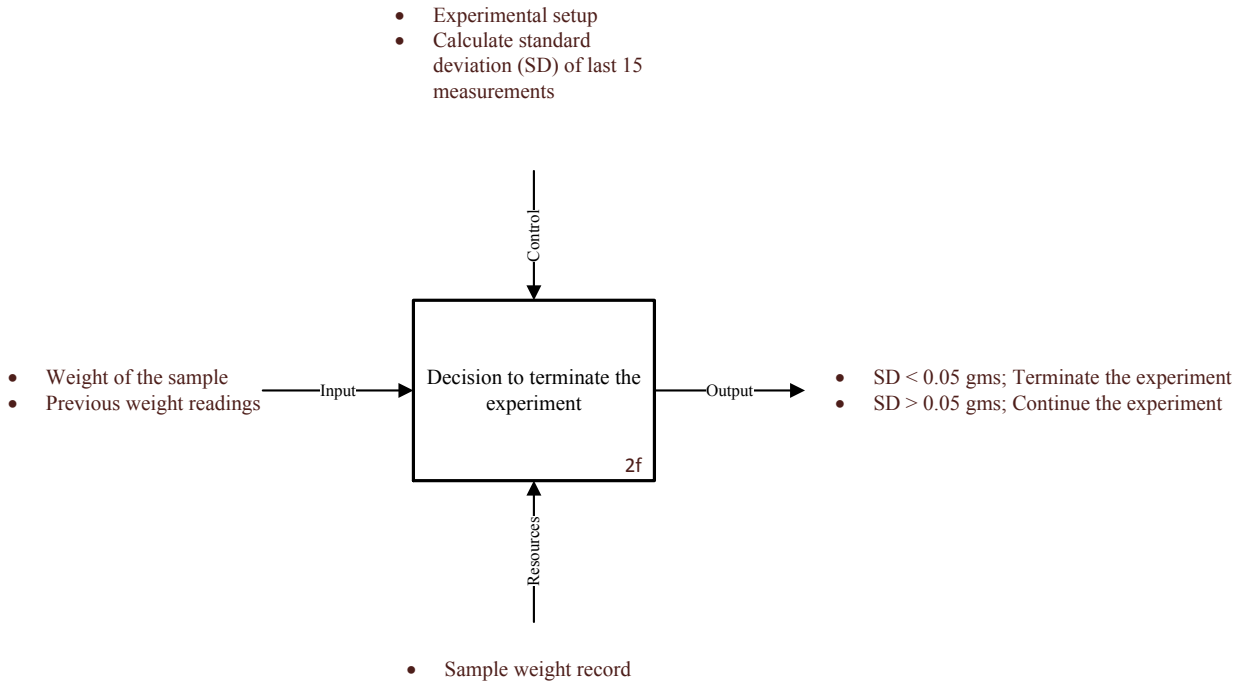

Figure 2-34: IDEF0 model for element 2f

To decide when to terminate the experiment, calculate the standard deviation of the mass measurements from the previous 15 samples. If the value of the standard deviation is less than 0.05 g, then the experiment is terminated; otherwise, it is continued and previous steps are repeated.

Figure 2-35 describes the steps following experiment termination.

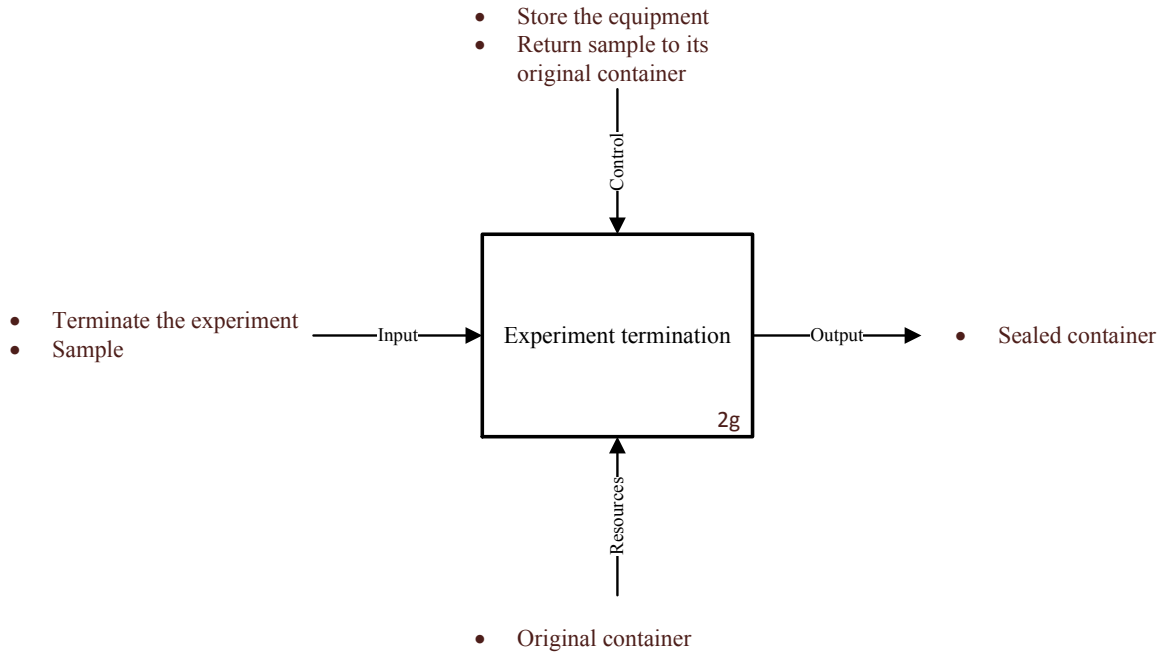

Figure 2-35: IDEF0 model for element 2g

Once the experiment finishes, the equipment is cleaned and stored. The sample is returned to its original container, which is sealed and stored in the designated cupboard of the lab. The MFT samples studied for the experiment are shown in Figure 2-36 as they look after the completion of the experiment.

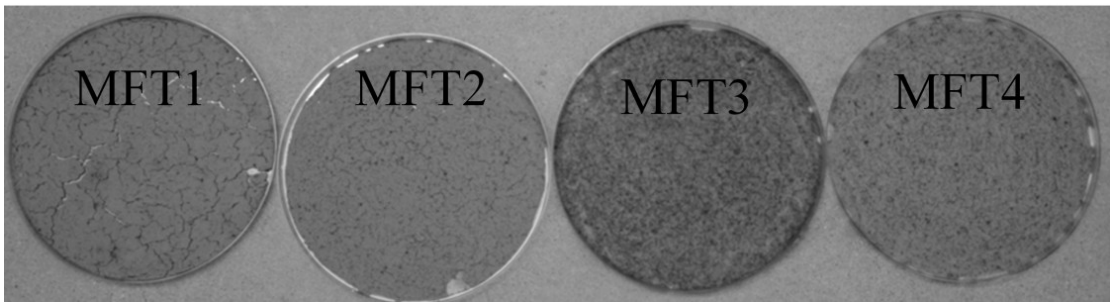

Figure 2-36**:** Dried MFT samples post experiment [46]

At this stage, the normalised evaporation (AE/PE) and moisture content (MC) are calculated using the formulae shown in the block of figure 2-37.

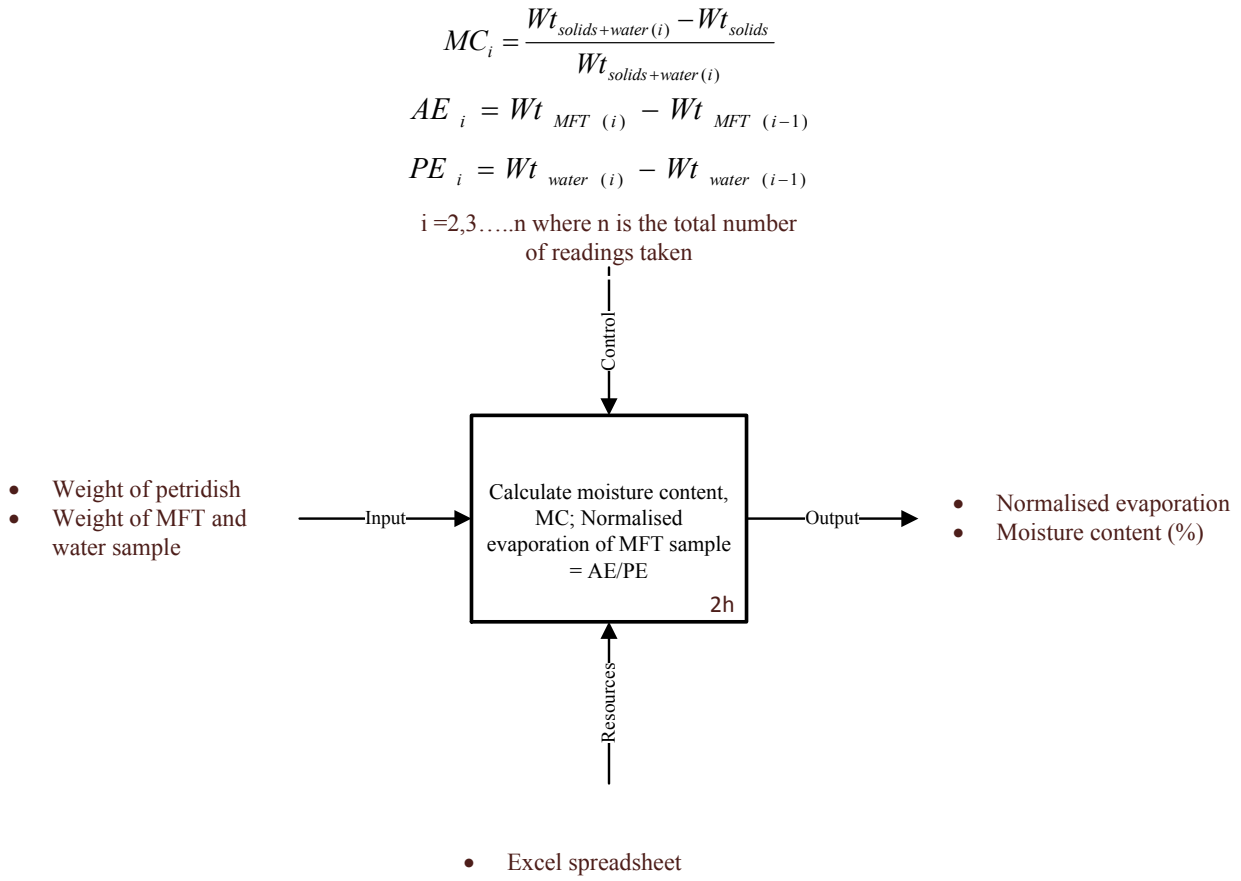

Figure 2-37: IDEF0 model for element 2h

$Wt_{solids+water}$ refers to the weight of sample without the weight of Petri dish and Wt_{solids} is the last weight reading of sample when it is all dried up minus the weight of Petri dish.

At this step, the normalised evaporation (AE/PE) is calculated using the formula shown in the diagram (Figure 2-38).

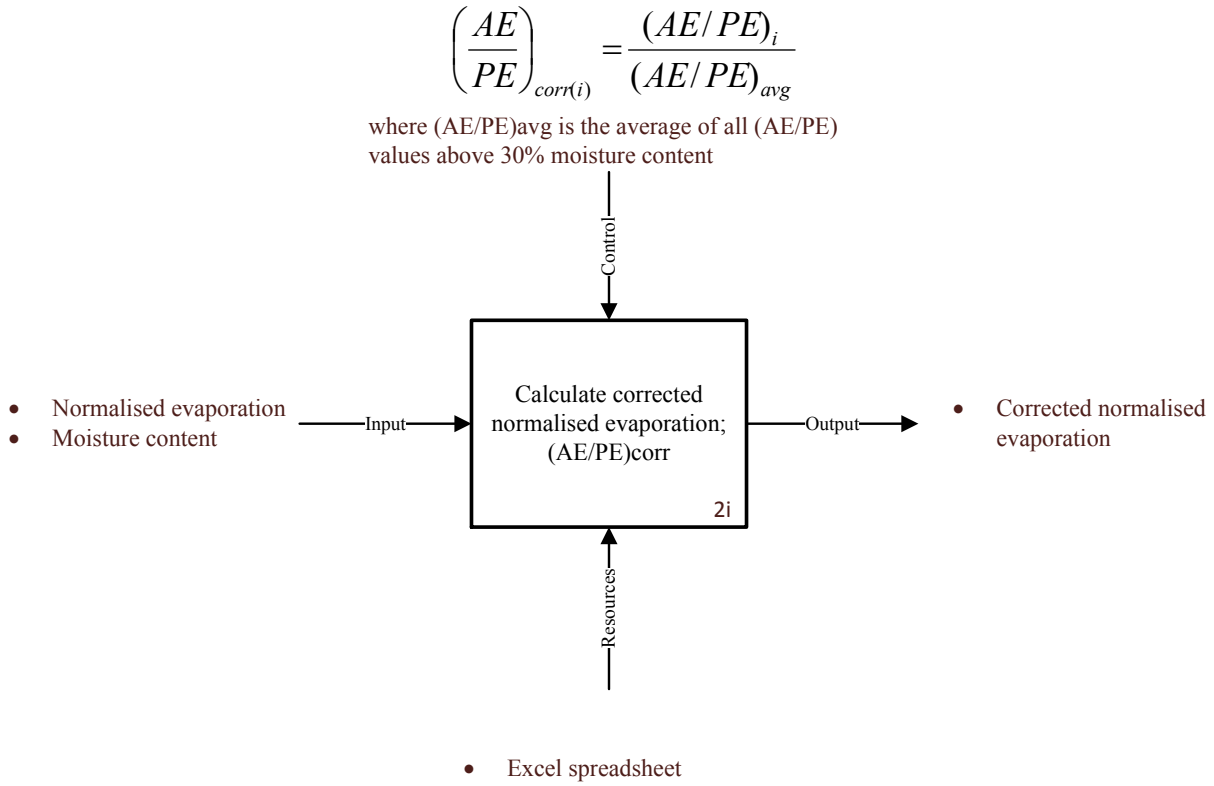

Figure 2-38**:** IDEF0 model for element 2i

The values are taken from the excel spreadsheet and corrected values are saved in the same sheet. These values of normalised evaporation are used further in the process of analysis.

At the next step, spectra obtained from the experiment is compiled (Figure 2-39).

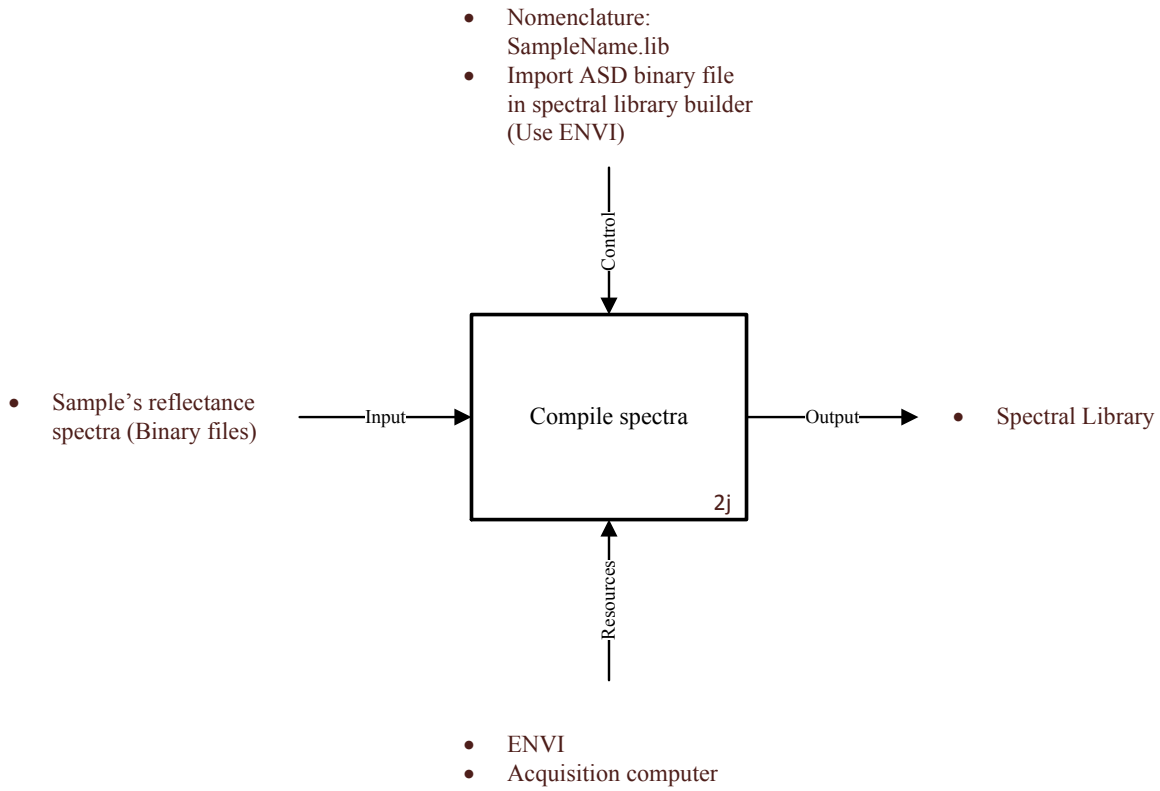

Figure 2-39: IDEF0 model for element 2j

Sample's reflectance spectra files (binary files) stored on the acquisition computer are imported to ENVI and compiled into a spectral library (SampleName.lib) using the spectral library builder function in ENVI.

Figure 2-40 focuses on the next step of continuum removal.

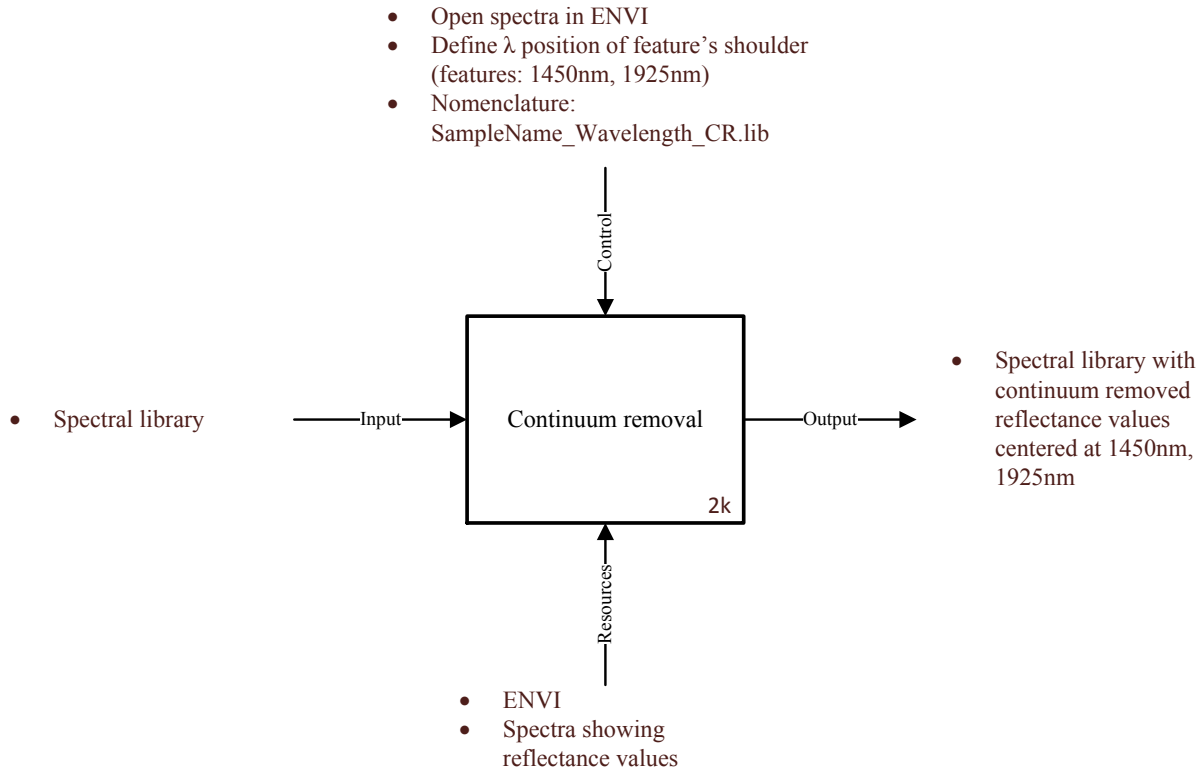

Figure 2-40: IDEF0 model for element 2k

For accurate analysis, the continuum is removed from the reflectance spectra for the spectral range of 1450 nm to 1925 nm. These wavelength positions are shoulders to the water absorption feature and are carefully defined. The positions are defined by looking for nearest left and right shoulders to the water absorption position on each spectrum displayed using ENVI. ENVI's continuum removal function is used to achieve the results.

The data representations obtained so far are exported to Excel for firther analysis. (Figure 2-41).

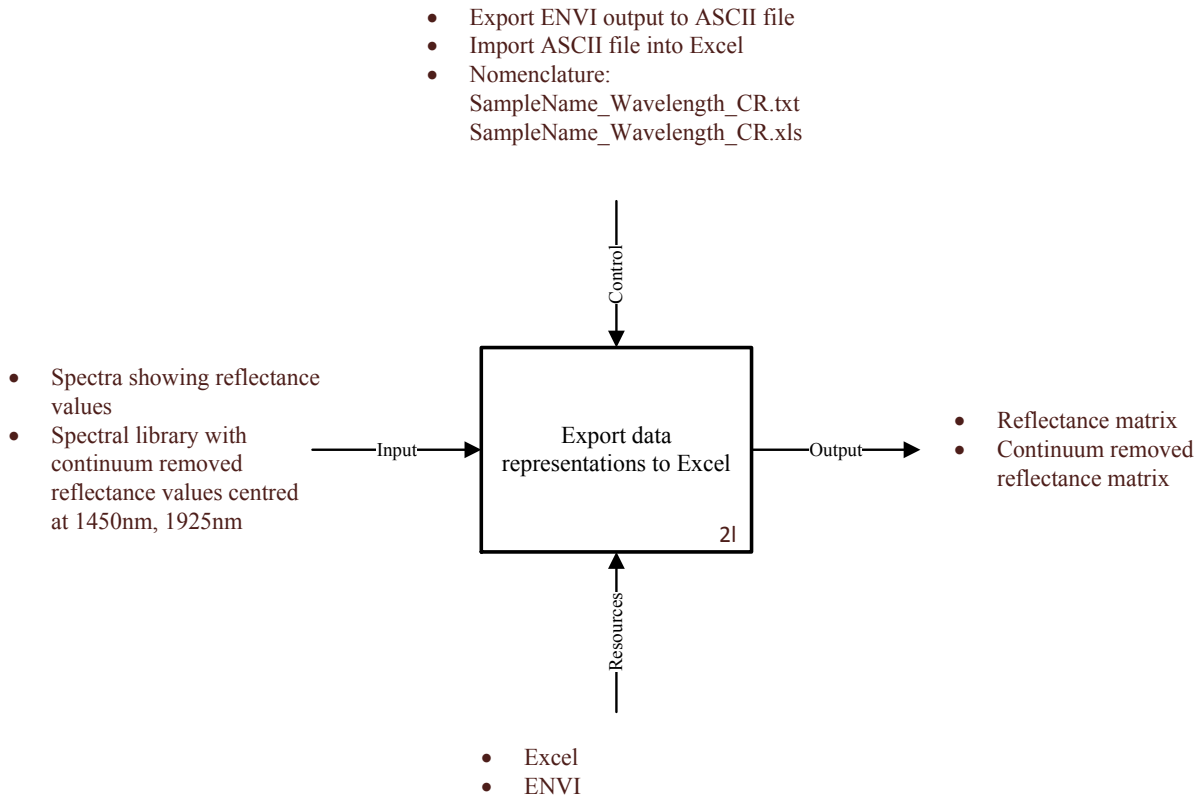

Figure 2-41: IDEF0 model for element 2l

Continuum removed reflectance spectra are exported to an ASCII file and then imported into an Excel file to create a matrix base for analysis. Two kinds of matrices are obtained from this step - Reflectance matrix and Continuum removed reflectance matrix. The results are saved in an Excel Workbook (SampleName_Wavelength_CR.xls)

Linear correlation analysis is performed in the next step (Figure 2-42).

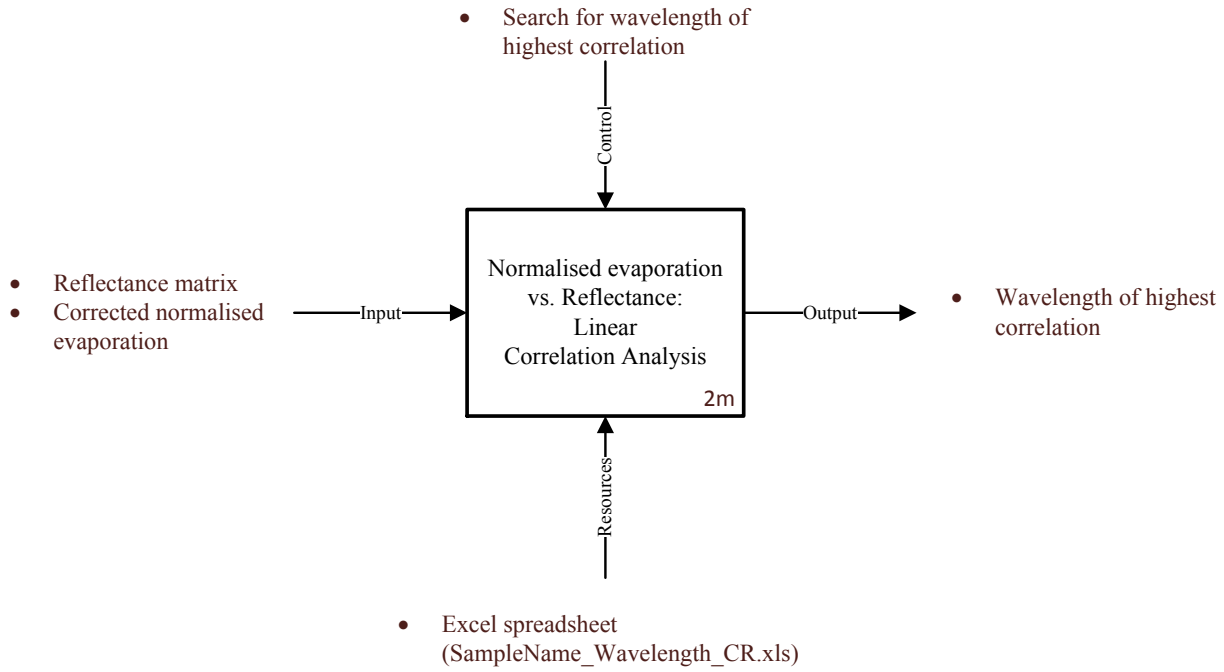

Figure 2-42: IDEF0 model for element 2m

A linear correlation analysis between normalised evaporation and reflectance is obtained using the reflectance matrix and corrected values of normalised evaporation from excel spreadsheet (SampleName_Wavelength_CR.xls). The correlation is used to determine the wavelength of highest correlation.

Figure 2-43 represents the next step of the process.

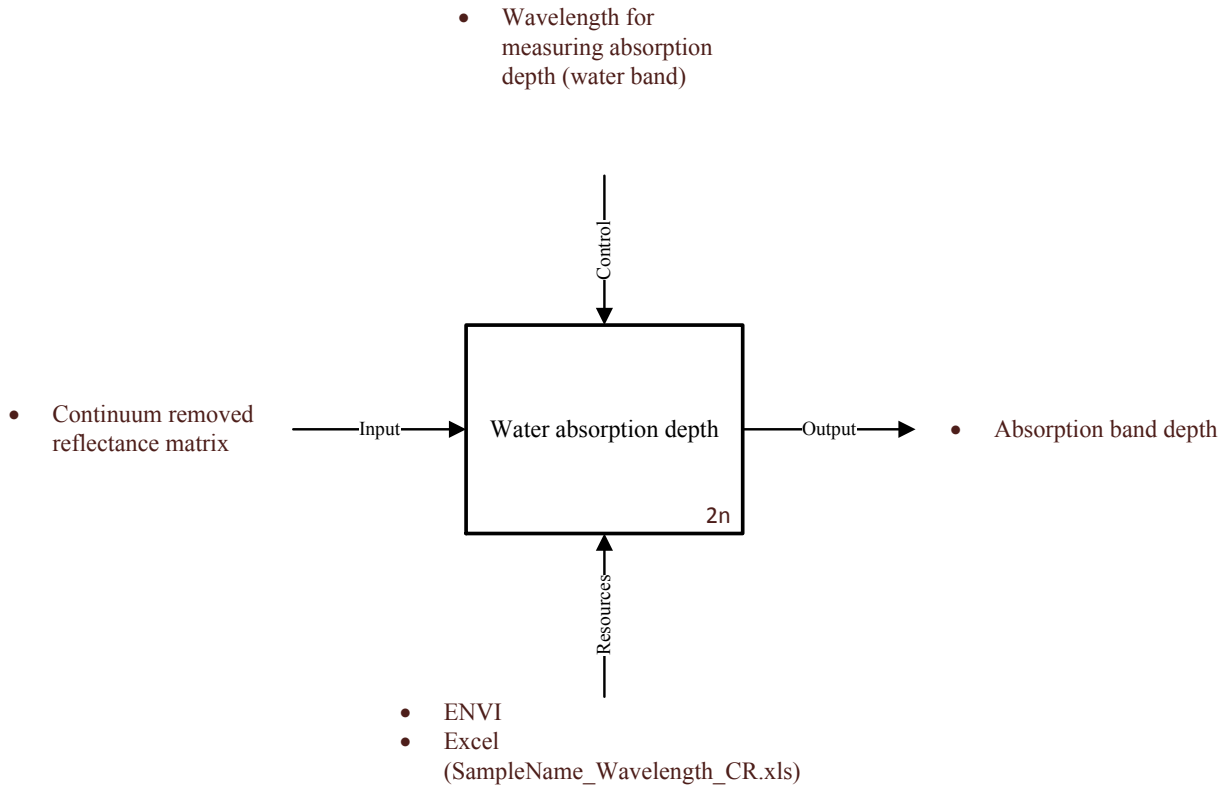

Figure 2-43: IDEF0 model for element 2n

The continuum removed reflectance matrix is now used to obtain absorption band depth using water band wavelength. The matrix is obtained from the excel spreadsheet and the results are stored in the same sheet.

NSMI values are also calculated in the process using guidelines given in Figure 2-44.

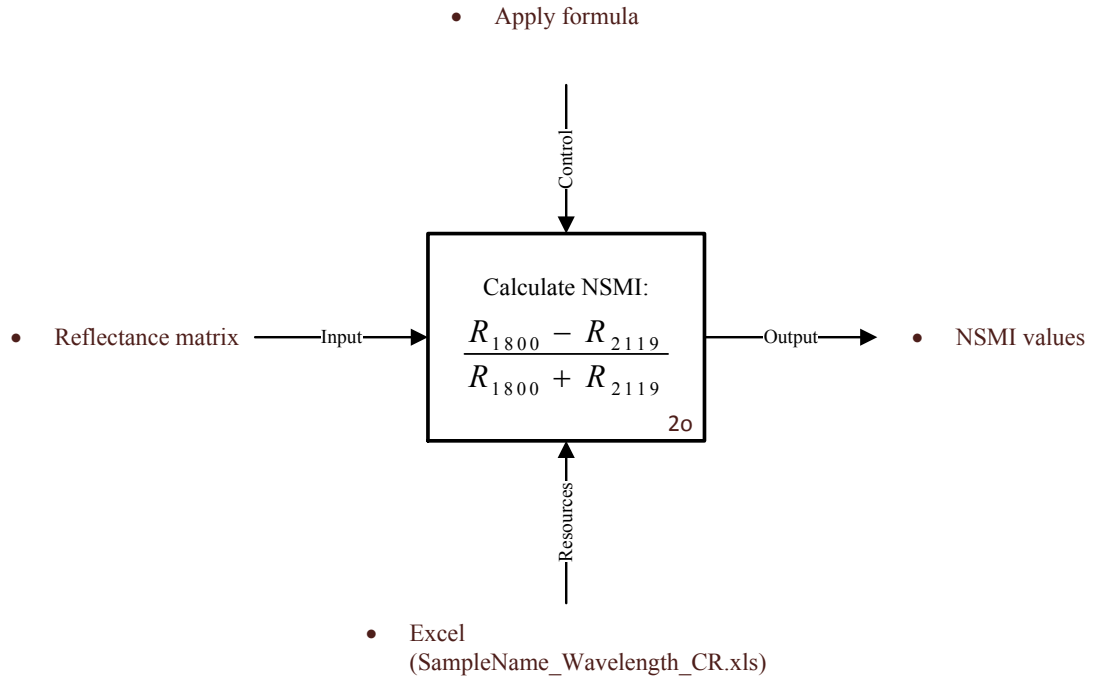

Figure 2-44: IDEF0 model for element 2o

Using the formula indicated in figure 2-44, NSMI vales are caluclated. Reflectance matrix stored in the excel spreadsheet is used to identify reflectance values at 1800 nm and 2119 nm, and the formula is applied to obtain NSMI values at each moisture content level. These values are also saved in the same excel file.

At the next step, a final database is popultaed with the data calculated so far (figure 2-45).

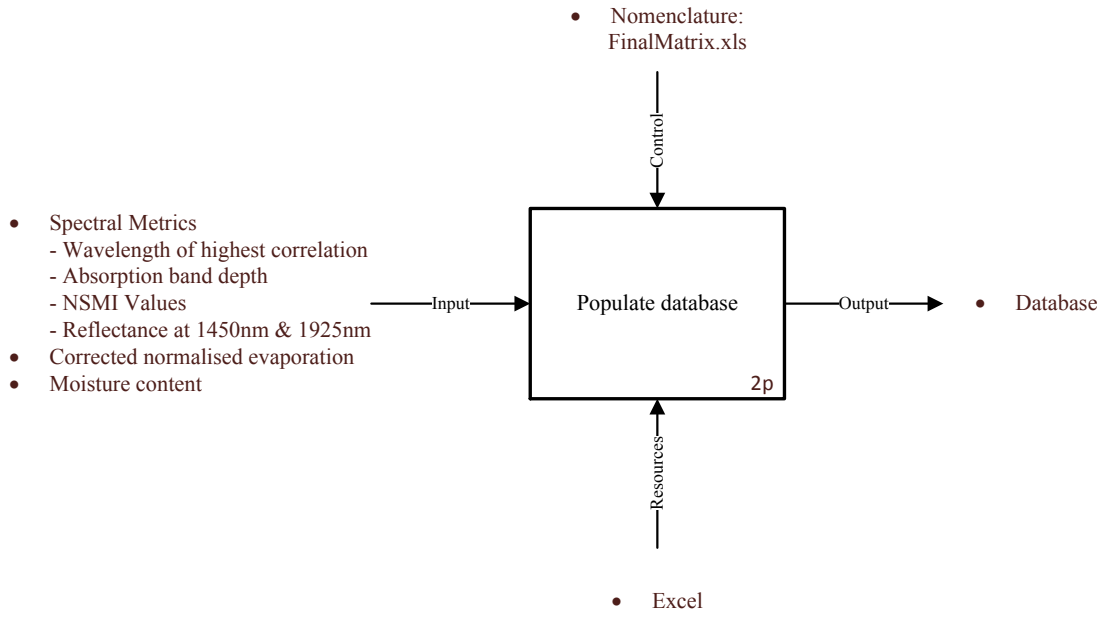

Figure 2-45: IDEF0 model for element 2p

A database using spectral metrics (wavelength of highest correlation, absorption band depth, NSMI values, reflectance at 1450 nm and 1925 nm), corrected normalised evaporation and moisture content is populated in excel (FinalMatrix.xls)

Next, the graphical representations are constructed as described in figure 2-46.

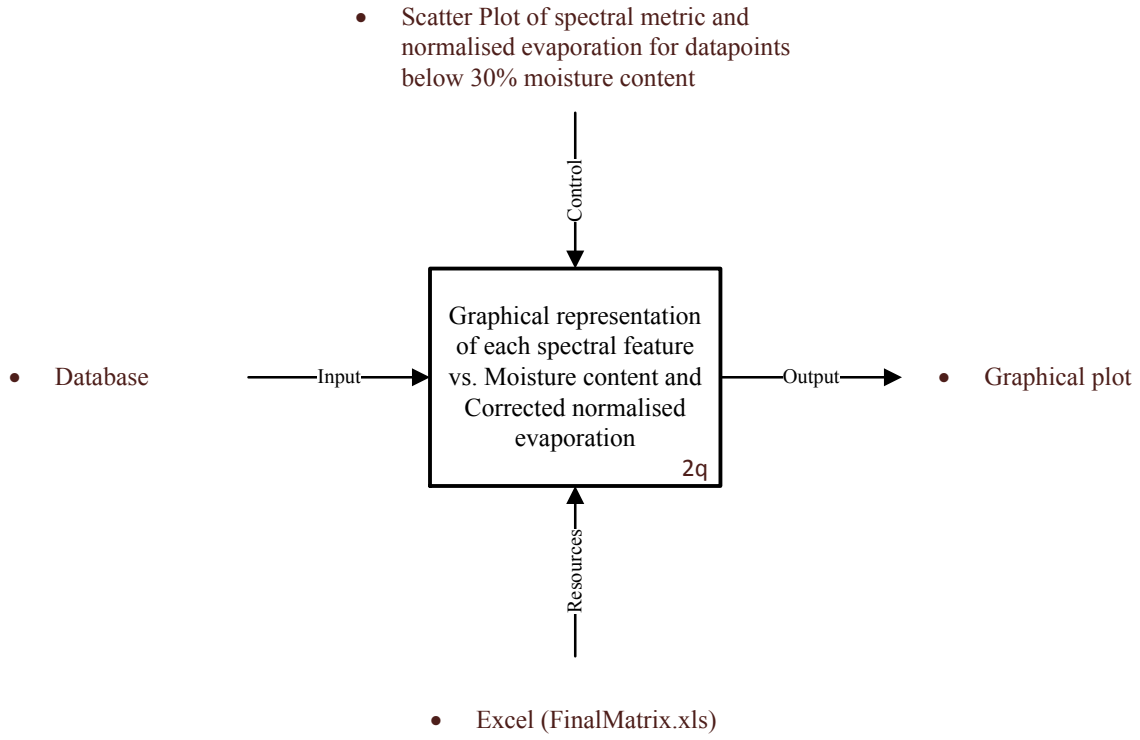

Figure 2-46: IDEF0 model for element 2q

A scatter plot of each spectral feature with normalised evaporation is created in excel using data values below 30% moisture content from the database populated in the last step.

Finally, simple regreession analysis methos is used to determine the best model (figure 2-47).

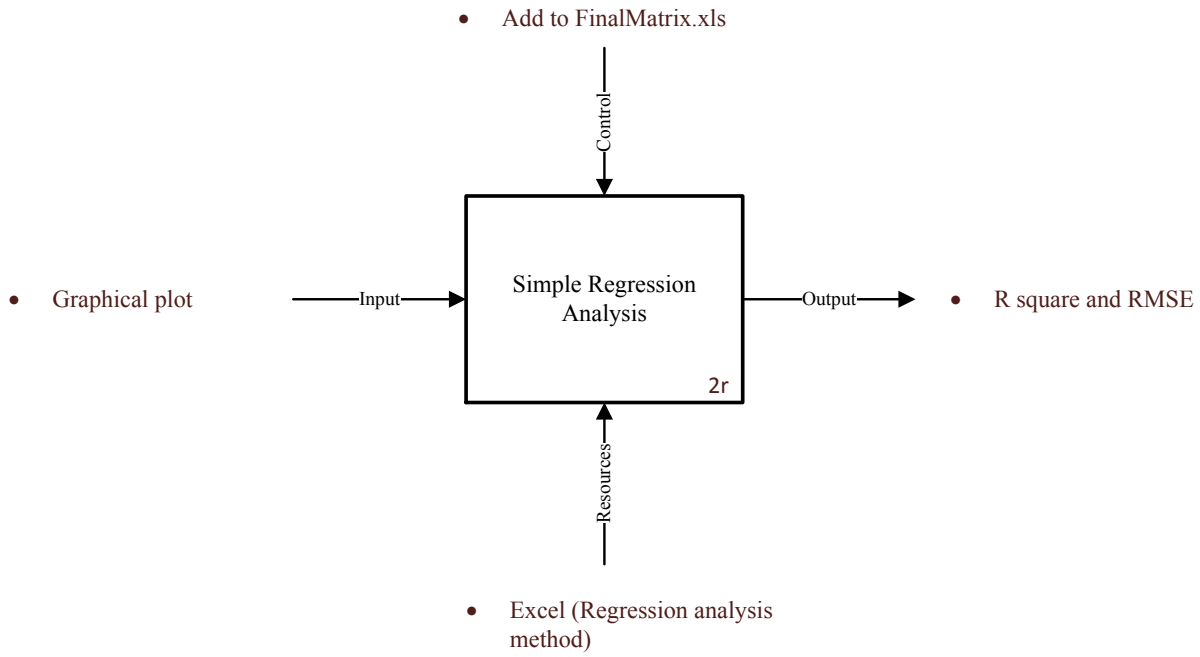

Figure 2-47: IDEF0 model for element 2r

A regression analysis of graphical plots is done to calculate R square and RMSE values using Regression analysis method in excel. The results show the NSMI model to be the most robust predictive model.

2.6.4 Step 4: Process Value Analysis

A qualitative analysis of the evaporation experiment is presented in the table 2-2. The analysis shows that all the activities involved in this experiment are necessary to be performed to achieve the outputs. Each activity contributes in converting the inputs listed in the SIPOC model to the outputs desired by the customers. The process flowchart also exhibits that the experimental procedure is sequential and hence the execution of each step is vital in the successful completion of the process.

Activity	Customer Value Added (Y/N)	Operational Value Added (Y/N)	Non-Value Added (Y/N)
Sample Arrives	N	Y	N
Setup experimental equipment	N	Y	N
Prepare water and MFT sample for the experiment	N	Y	N
Record white panel's spectrum	N	Y	N
Measure weight and temp of water & sample, air temperature and relative humidity; Get sample's reflectance spectrum; Let sample dry	N	Y	N
Decision to terminate the experiment	N	Y	N
Experiment termination	N	Y	N
Calculate moisture content and normalised evaporation	N	Y	N
Calculate corrected normalised evaporation	N	Y	N
Compile spectra	N	Y	N
Continuum removal	N	Y	N
Export data representations to Excel	N	Y	N
Normalised evaporation vs. Reflectance: Linear correlation analysis	Y	N	N
Water absorption depth	Y	N	N
Calculate NSMI	Y	N	N
Populate database	Y	N	N
Graphical representations of spectral features vs. moisture content & corrected normalised evaporation	Y	N	N
Simple regression analysis	Y	N	N

Table 2-2: Process value analysis of the evaporation experiment

2.7 Discussion

The research captured in this chapter studies the importance of qualitative analysis on a system before strategies of improvement can be applied to a system or process. It implemented a careful documentation of the processes and enabled any required corrections along the way. It served as a quality check on the processes which would have been absent otherwise. It could also help in quickly revising a framework and direction as the new information emerges or in transferring the documentation if the work is modified.

The results from this study indicate that because of the absence of non-value added activities in the experiments (table 2-1 and table 2-2) and because of their sequential nature, each activity performed contributes towards the end results in a way that its replacement or omission would

affect performance of these experiments. The outputs from every activity feed into other activities in the process as seen from the system flowcharts (figure 2-5 and figure 2-26). This further highlights the importance of execution of each element of the process. The activities, however, can be made more efficient by incorporating certain experimental requirements and managing the database more actively. As well, some repetitive tasks may be automated, especially if a process output is prone to error if there is process variability, and the task takes a long time (making it more expensive for a person to be doing the task manually).

Following this study, it would be interesting to perform a quantitative analysis to substantiate the results discussed earlier. Such an analysis would allow to meausure performance dimensions such as the time taken to perform an activity and the associated cost of performance. Constituents of time are "processing time" which is the time that resources spend on performing a process and "waiting time" due to the absence of resource to perform a process step. The time to perform an activity is accounted for in the "control" part of the IDEF0 model. Another parameter that should be considered is the cost of performing an activtiy. While measuring cost, several types of costs can be considered such as cost of production, cost of delivery, cost of physical elements and cost of human resources [47]. The cost should also be accounted for in the "control" part of the IDEF0 model. Once the cost and time values are added to each activtiy, they are summed up to calculate the total cost and the total time. The next step is to look for activities where waiting time can be eliminated or reduced, processing time can be decreased, and the associated cost can be minimized. The objective is to minimize the total time and the total cost of the process. The results obtained after the suggestions are applied are compared to the original results to ensure the quality of output is not compromised. The process is continued until time and cost are reduced to the minimum while still maintaing/improving the output quality.

Chapter 3

Total Bitumen Content Profile Creation

3.1 Introduction

Drill core sampling is the principal method for investigating the subsurface geology in industry operations in the Athabasca oil sands and data from drill core analyses are often the basis of reservoir models [4]. Speta et al., [31] illustrated the use of hyperspectral imaging, a remote sensing technique, to expedites the process of total bitumen content (TBC) estimation. This chapter builds on the research done to estimate TBC from hyperspectral images of drill core samples and examines the creation of profiles of bitumen content from hyperspectral images maps of TBC. The profiles represent variation in TBC as a function of depth along the drill core. and convey statistical parameters computed at a range of drill core downhole resolutions. A background on acquisition of data files used to create these profiles is provided in the following section.

3.2 Data Collection and Pre-Processing

Drill cores from Athabasca oilsands deposits were investigated at the Department of Earth and Atmospheric Sciences at the University of Alberta. Spectral imagery of these drill cores were collected with a SisuROCK core imaging system, developed and manufactured by Spectral Imaging (Specim) Ltd., Finland..

Cores are loaded on a platform covered by two high resolution spectral cameras and illuminated by an array of quartz halogen lamps. Up to four boxes of core were scanned at the same time, one of the scanned images is shown in figure 3-1. Further detail on the procedure is mentioned in

the work published by Speta et al. [31]. As seen, each scanned image contains four core boxes and is saved using nomenclature: SampleName_SampleNumber_StartDepth_EndDepth_Date-Time_refl.dat, for example the name of the image shown in figure 3-1 is "MRM13-CR0127_13_67m53_74m40_131218-143553_refl". The order of the core boxes is as marked in the picture.

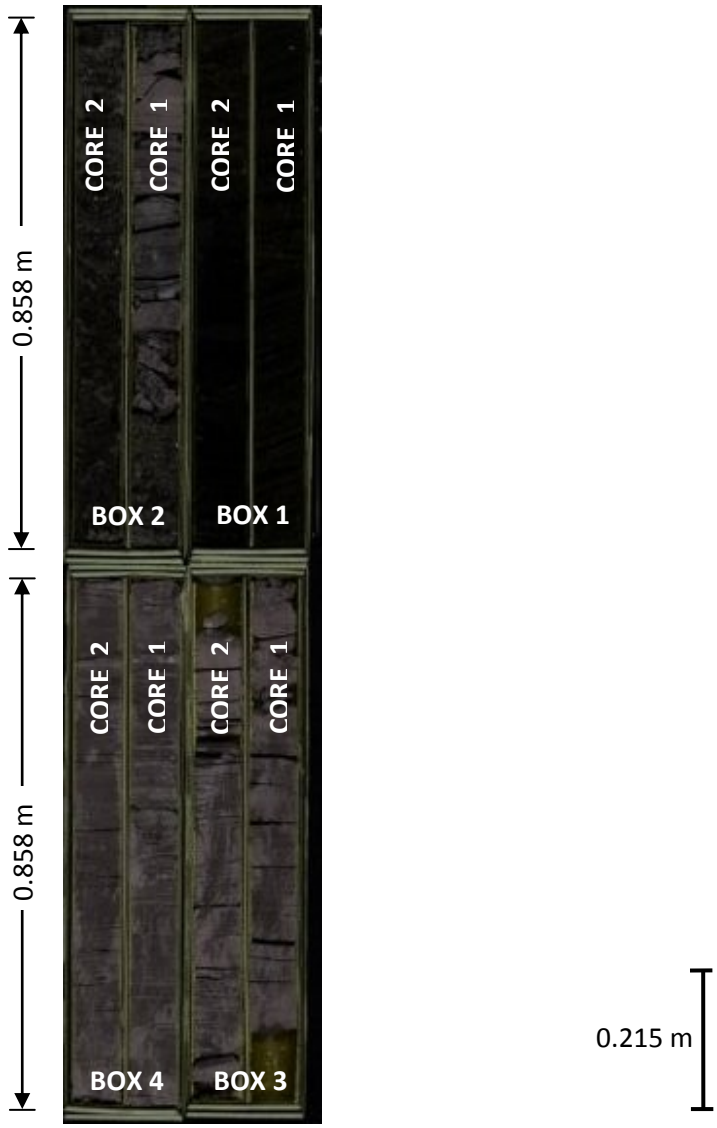

Figure 3-1:Image of four boxes of cores scanned at a time by SisuROCK imaging system. Order of boxes and cores is as marked in the image. Filename: MRM13-CR0127_13_67m53_74m40_131218-143553_refl

Using the "Box and Core Cropper" image-processing software tool designed and developed by Sen Cao at the Center for Earth Observation Sciences, University of Alberta, individual boxes and core samples were cropped out of these data files, hence generating a set of four boxes and eight cores from each file. This tool also mosaics all cores and boxes for a given drill hole, stitching them together horizontally and vertically, and thus generating two mosaicked data files (Nomenclature: SampleName_Core/Box_Horizontal/Vertical_mosaic.dat), an example of which is shown in figure 3-2.

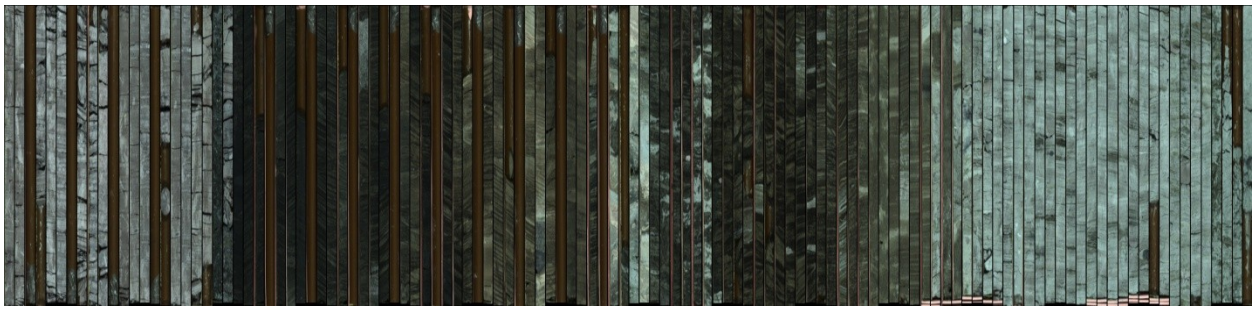

Figure 3-2: MRM13-CR0127-SWIR_Core_horizontal_mosaic, a horizontal mosaic of 120 core samples

After the mosaic images are created, TBC estimates are calculated using the tool created in ENVI/IDL by Dr. Jilu Feng at the University of Alberta based on the estimation models presented in [34], [35] and[47]. This tool uses three different types of models to calculate TBC estimates as detailed below. The output file generated from this tool is saved as a data file and named "SampleName_Core/Box_Horizontal_mosaic_TBC.dat" (the corresponding TBC estimated data file for above example shown in figure 3-2 would be named MRM13-CR0127-SWIR_Core_horizontal_mosaic_TBC.dat) contains three bands that include estimated values from each of the three models. The three models used to calculate TBC% estimates are a) Broad Band-Core b) Wavelet-Core c) Wavelet-Homogenized Sample and the bands are named after these models. These models produced TBC estimates within _+/- 1.1% to +/- 3% of Dean Stark values. This TBC prediction tool, however, only generates estimates for horizontally mosaicked

data files. The data files created using this tool were directly used in the process of creation of bitumen profiles and therefore these processes inherit the error of this prediction tool. The following section discusses the process of creation of these data profiles and the results hence generated.

3.3 Methodology and Results

This research captures TBC estimated data files created in section 3.2 to generate vertical profiles using IDL (Interactive Data Language). Vertical bitumen profiles, as already mentioned, represent variation in TBC values along the depth of a drill hole. Mean and standard deviation of TBC values are traced through the depth to create a profile. These profiles are useful in determining the areas along the core that are homogenous and rich in bitumen. The following steps describe the process of creation of vertical profiles for mosaic data files.

3.3.1 Step 1: Determine start and end depths of core samples

The first step in creating the profiles is to accurately determine the true depth of each core in the mosaic file. When drill core samples were scanned, four boxes of core or eight core samples were run through the system together. And therefore the start depth and the end depth on the file name correspond to the start and end depth of the eight samples collectively or in other words, start depth of the first core and end depth of the eighth core. To determine starting and ending depth of each core sample in the boxes, the total depth of four boxes or eight cores is linearly interpolated using an algorithm created in IDL (Appendix A-1). The program runs on individual core files and box files created using "Box and Core Cropper" tool. It allows the user to select the folder holding data files, and then uses the file names and the method of linear interpolation to calculate start and end depth values for each file in the folder. The results are written on two comma-separated values (csv) files by the name

"FilesDatabase_NumberofFiles_Core/BoxFiles.csv" and saved in the original folder. One of the files holds information about all data files with core samples and the other one about data files with boxes. An example of the output file created is shown in table 3-1. "S.No." on the data sheet is the combination of sample number, box number, and core number and is used to arrange the data in an ascending order while using it later in the study. In this study lost core that may result in inappropriate depth calculation within a box was not taken into consideration.

S.No.	File Name	Total Depth (m)	Start Depth (m)	End Depth (m)	No. of Lines	No. of Samples	No. of Bands
1011	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 1_Core 1	0.49	56.2	56.69	533	42	242
1012	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 1_Core 2	0.49	56.69	57.17	530	41	242
1021	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 2_Core 1	0.49	57.17	57.66	528	43	242
1022	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 2_Core 2	0.49	57.66	58.14	529	42	242
1031	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 3_Core 1	0.49	58.14	58.63	535	42	242
1032	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 3_Core 2	0.49	58.63	59.11	535	45	242
1041	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 4_Core 1	0.49	59.11	59.6	534	42	242
1042	MRM13-CR0127_10_56m20_60m08_131218-141126_refl_Box 4_Core 2	0.49	59.6	60.08	534	44	242
1111	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 1_Core 1	0.46	60.08	60.54	533	42	242
1112	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 1_Core 2	0.46	60.54	61	530	41	242
1121	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 2_Core 1	0.46	61	61.46	528	43	242
1122	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 2_Core 2	0.46	61.46	61.92	529	42	242
1131	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 3_Core 1	0.46	61.92	62.37	535	42	242
1132	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 3_Core 2	0.46	62.37	62.83	535	45	242
1141	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 4_Core 1	0.46	62.83	63.29	534	42	242
1142	MRM13-CR0127_11_60m08_63m75_131218-142020_refl_Box 4_Core 2	0.46	63.29	63.75	534	44	242
1211	MRM13-CR0127_12_63m75_67m53_131218-142950_refl_Box 1_Core 1	0.47	63.75	64.22	533	42	242
1212	MRM13-CR0127_12_63m75_67m53_131218-142950_refl_Box 1_Core 2	0.47	64.22	64.69	530	41	242
1221	MRM13-CR0127_12_63m75_67m53_131218-142950_refl_Box 2_Core 1	0.47	64.69	65.17	528	43	242
1222	MRM13-CR0127_12_63m75_67m53_131218-142950_refl_Box 2_Core 2	0.47	65.17	65.64	529	42	242

Table 3-1: An example of the output file holding information of all individual data files with core samples, created using the program to determine total depth, start depth, and end depth of the core samples. Filename: "FilesDatabase_120_CoreFiles.csv"

After Step 1, vertical profiles were calculated using two methods. The methods are created to process core mosaic files instead of box mosaic files because the desired vertical profiles represent data in the cores and using box files adds undesirable noise to the data. Method No. 1

creates profiles of data by computing line by line values of mean TBC and standard deviation using every single pixel each time the program is executed. Method No. 2 is a more efficient way of generating the same representations as it stores line-byline values of mean and standard deviation in a file at an intermediate step. It then uses that file to compute another mean and standard deviation using values stored in the file. Therefore, it saves the user a lot of time as it does not scan every pixel each time the method is called. The difference can be illustrated using figure 3-3.

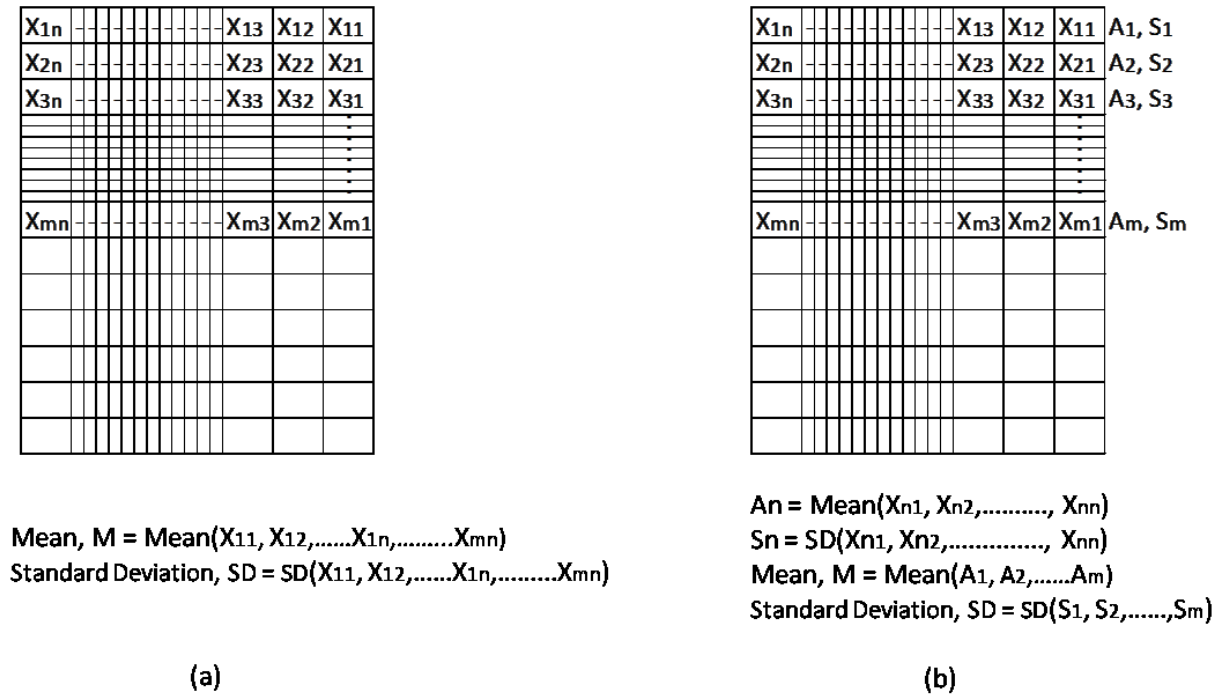

Figure 3-3: Schematic representations exhibiting calculations of two methods used to create bitumen profiles a) Method No. 1 b) Method No. 2

The difference between the computations of the two methods can be seen in figure 3-3 where Xij represents a pixel. If for instance, a profile is created at a scale of 25 cm which corresponds to a section of m lines with n pixels in each line then figure 3-3 (a) explains the computation process of Method No. 1. The mean, M, and standard deviation, SD, of the entire 25 cm section is

calculated using m x n pixels. Method No. 2 first calculates mean, An, and standard deviation, Sn, for each line and then calculates a final mean, M, and standard deviation, SD, of the section using line average and standard deviation values as shown in figure 3-3 (b). The two methods are described step by step in the following sections.

3.3.2 Method No. 1 (Basic approach)

Because of the nature of the computation process, this method leads to no substantial loss in data or accuracy of values obtained after the TBC values are calculated. It uses one more step in addition to step 1 (section 3.3.1) to complete the process of profile creation. Step No. 2 is described in detail below.

3.3.2.1 Step 2: Create vertical profiles

At this step, a program in IDL was created to generate vertical data profiles (Appendix A-2). This program asks the user to input the mosaic data file, the data sheet created in Step 1 and the desired length (depth) of the profile interval. The length of the interval refers to the distance between two consecutive output data points, and the depth of each section in the core samples considered to generate data points on the profile. For instance, if the user sets the interval at 10 cm then after every 10 cm interval, a section of 10 cm in depth is taken to produce a data point on the profile from the average of all pixels included in that interval. This choice of profile view is referred to as a "Interval Profile" in this study. Rather than defining a fixed interval to produce a profile the user can also choose to calculate a moving average to generate the profile. For the same example of a10 cm interval setting, this option would generate data points from successive sections of 10 cm in depth by averaging all pixels included in the section. This is referred to as a "Moving Average Profile" throughout this study. Moving average profiles are hence much smoother than the interval profiles. Standard deviation of TBC values are shown as an error

around the mean values on the profiles. The envelope around the mean values therefore indicates values representing mean plus standard deviation and mean minus standard deviation. A typical interval of 10 cm would encompass the average of about 10540 pixels (85 lines X 124 samples).

The program scans through the TBC mosaic file line by line and calculates average and standard deviation values for the depth of the section specified by the user. It writes a csv file named "MosaicFilename_DepthoftheInterval_MovingAverage/Interval_Data.csv". The file holds start depth, end depth and central depth of each section, and calculated average, and standard deviation of TBC values of that section for each band. Table 3-2 shows an excerpt of the file written by the program as a result of creating interval profiles at an interval of 10 cm.

S.No.	Start Depth(m)	End Depth(m)	Central Depth(m)	Average_TBC% from Broad Band-Core	Average_TBC% from Wavelet-Core	Average_TBC% from Wavelet-Homogenized Sample	SD_TBC% from Broad Band-Core	SD_TBC% from Wavelet-Core	SD_TBC% from Wavelet-Homogenized Sample
1	6.6	6.7	6.65	3.27	3.5	6.74	1.2	0.83	0.64
2	6.7	6.8	6.75	3.77	3.79	6.96	1.4	0.94	0.7
3	6.8	6.9	6.85	3.91	3.91	7.06	1.52	1.01	0.76
4	6.9	7	6.95	3.98	4.75	7.64	2.21	1.77	2.18
5	7	7.1	7.05	1.22	2.13	3.32	2.01	2.91	4.37
6	7.1	7.2	7.15	0	0.03	0.04	0.02	0.44	0.67
7	7.2	7.3	7.25	0	0	0	0	0	0
8	7.3	7.4	7.35	0	0	0	0	0	0
9	7.4	7.5	7.45	0	0	0	0	0	0
10	7.5	7.6	7.55	2.68	3.73	5.6	2.51	3.1	4.45
11	7.6	7.7	7.65	4.27	4.92	7.85	1.86	1.61	1.76
12	7.7	7.8	7.75	4.43	4.25	7.3	2.32	1.6	1.23
13	7.8	7.9	7.85	1.48	1.67	2.76	2.67	2.49	3.87
14	7.9	8	7.95	0.29	1.32	2.32	0.68	2.14	3.71
15	8	8.1	8.05	1.73	3.14	5.96	1.24	1.59	2.83
16	8.1	8.2	8.15	3.17	3.39	6.68	0.97	0.66	0.58
17	8.2	8.3	8.25	3.46	3.43	6.69	1.02	0.64	0.48
18	8.3	8.4	8.35	6.3	5.61	8.34	3.61	3	2.31
19	8.4	8.5	8.45	6.57	5.56	8.05	3.4	2.65	2.68
20	8.5	8.6	8.55	10.16	9.04	10.99	4.04	3.65	2.82
21	8.6	8.7	8.65	8.06	6.84	9.29	3.47	2.88	2.2
22	8.7	8.8	8.75	8.36	7.11	9.49	4.34	3.59	2.74
23	8.8	8.9	8.85	11.82	10.11	11.81	3.02	2.76	2.13
24	8.9	9	8.95	7.22	6.25	8.85	3.78	3.11	2.4
25	9	9.1	9.05	6.82	6.09	8.58	4.12	3.4	3.16

Table 3-2: An example of the output generated using Method 1 (Basic approach). Filename: "MRM13-CR0127-SWIR_Core_horizontal_mosaic_TBC_10cms_Interval_Data.csv"

The program creates a vertical profile for each core and these profiles are mosaicked in the order of their respective core position in the original mosaic data file. It generates six images of vertical profile representations. Two images are produced for each band (model). For one of the images, profiles are arranged horizontally in the order of core samples (figure 3-4b) and in the other they are organized vertically (figure 3-4c). Nomenclature followed to name images is "MosaicFilename_BandName_DepthoftheInterval_(MA)_HorPlot/VertPlot.jpg". To the filenames, "MA" is added only if the user decided to view moving average of the profiles. The program also creates a subfolder with the same name as the mosaic filename and saves the outputs (csv file and images) in that subfolder.

Figures 3-5 and 3-6 show examples of vertical profiles created using Method No.1. Fig 3.5 (a) and Fig 3.6 (a) shows core no. 1 and core no. 6 of the core mosaic file : "MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat". Fig 3.5 (b) and Fig 3.6 (b) shows the TBC profiles of the respective core at a scale of 1 mm, Fig 3.5 (c, e) and Fig 3.6 (c, e) shows TBC moving average profiles at a scale of 10 mm and 50 mm respectively, and Fig 3.5 (d, f) anf Fig 3.6 (d, f) shows TBC interval profiles at a sclae of 10 mm and 50 mm respectively.

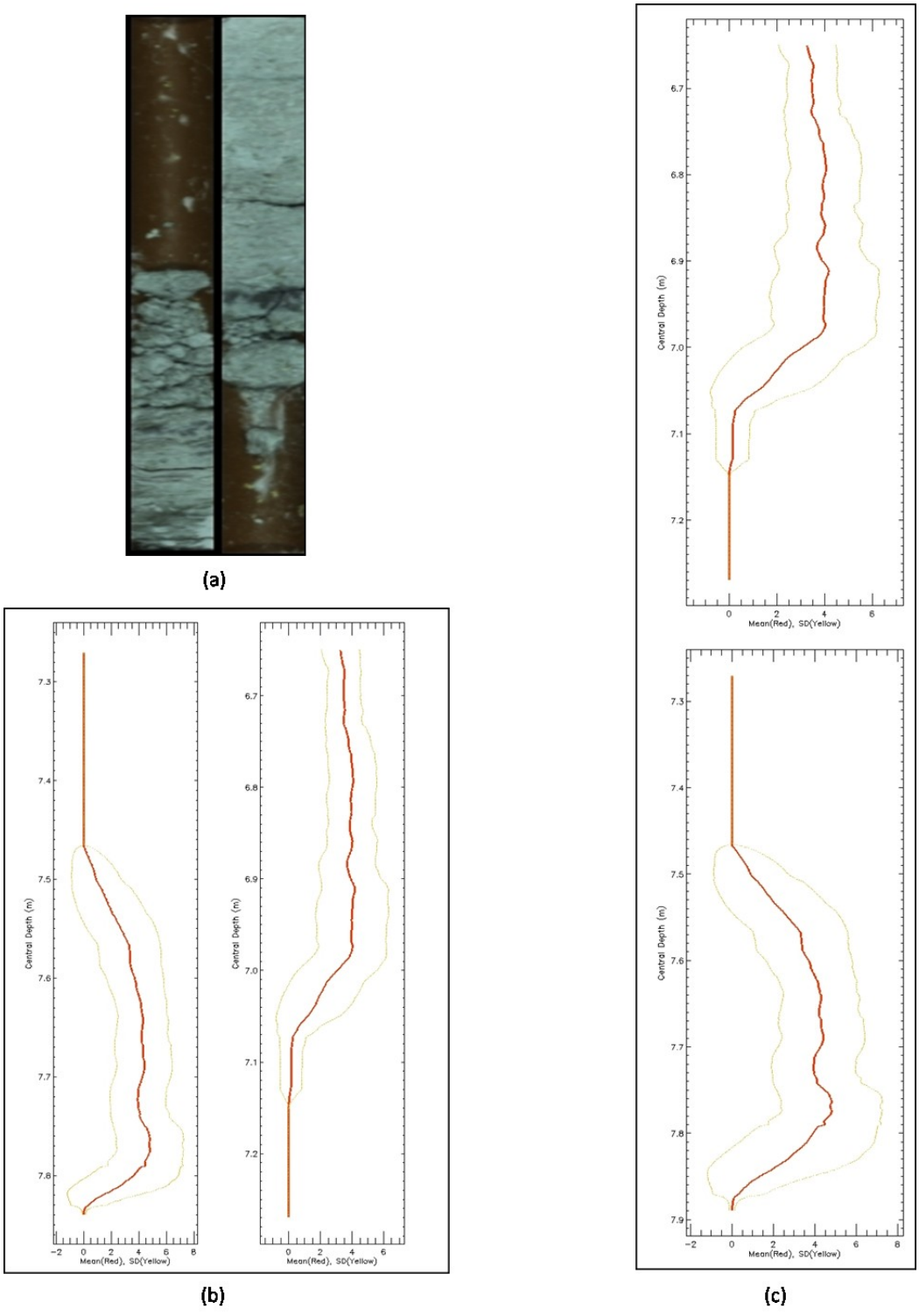

Figure 3-4: (a) Sample from the image of core mosaic file "MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" showing core no. 1 and core no. 2, (b) Vertical TBC profiles of the shown cores produced from Method 1 with moving average at an interval of 10 cm, arranged horizontally (c) same profiles of the shown cores arranged vertically. The vertical profiles represent variation of TBC as a function of depth (red colour) and an envelope of standard deviation around it (yellow colour)

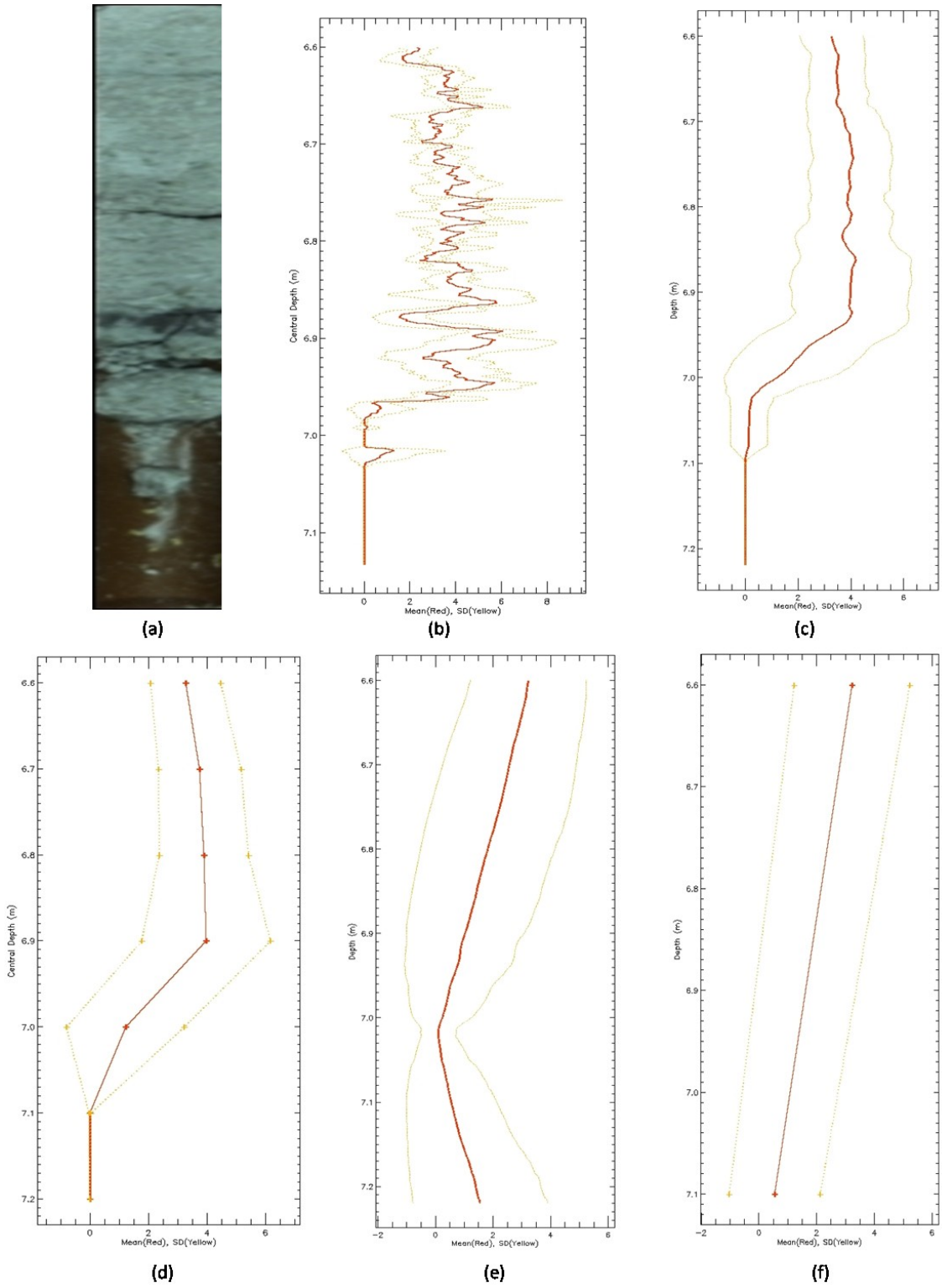

Figure 3-5: Samples from vertical TBC profiles created for " MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" using Method 1. It is an example of profiles showing homogenous composition of core a) Image of core no. 1 from the mosaic file, b) Profile of the core at a scale of 1 mm, c) Profile with moving average at a scale of 10 cm, d) Interval Profile at a scale of 10 cm, e) Profile with moving average at a scale of 50 cm, f) Interval Profile at a scale of 50 cm.

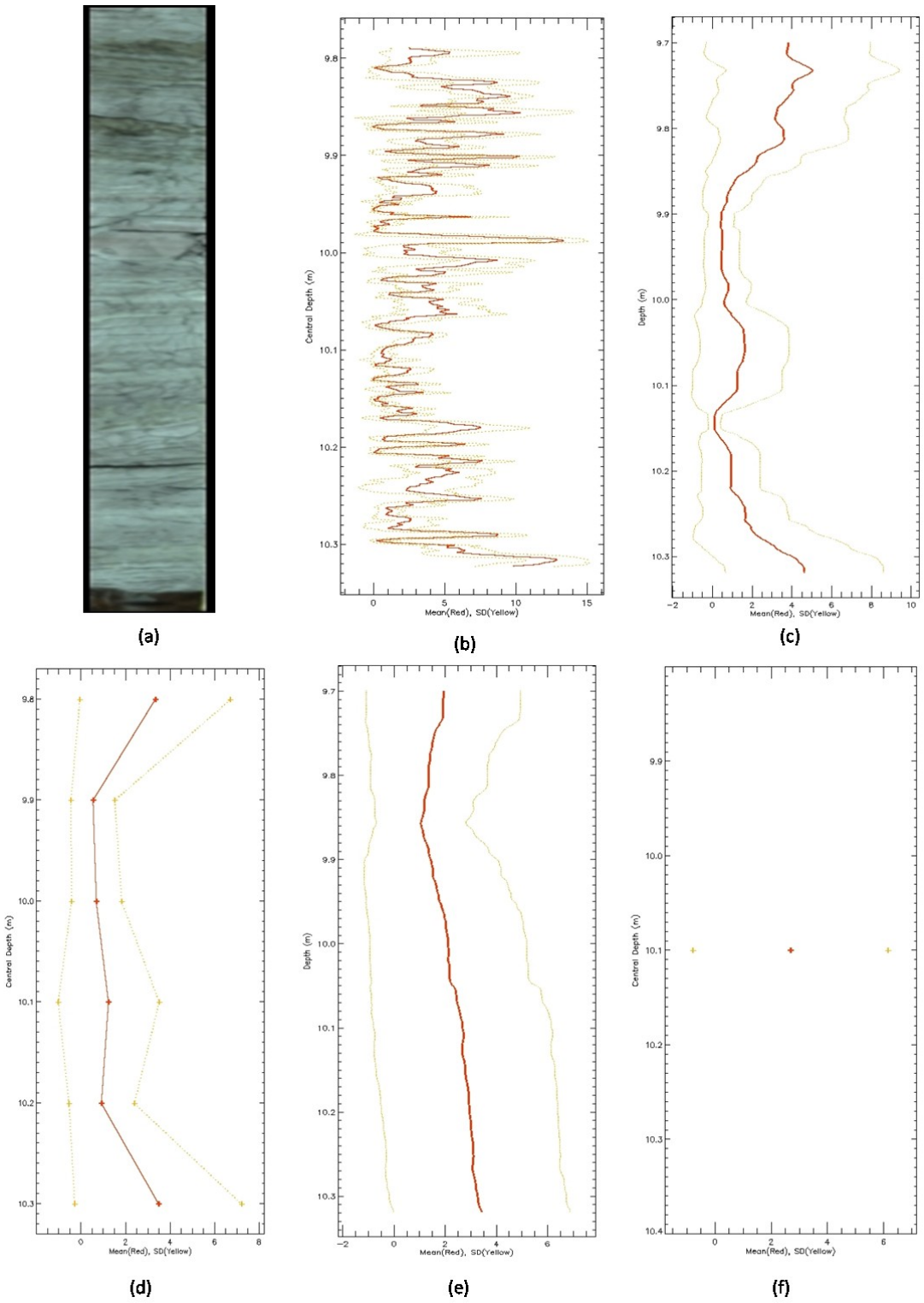

Figure 3-6: Samples from vertical TBC profiles created for " MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" using Method 1. It is an example of profiles showing heterogeneous composition of core a) Image of core no. 6 from the mosaic file, b) Profile of the core at a scale of 1 mm, c) Profile with moving average at a scale of 10 cm, d) Interval Profile at a scale of 10 cm, e) Profile with moving average at a scale of 50 cm, f) Interval Profile at a scale of 50 cm.

Figure 3-5 is an example of a core from the file "MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" with profiles exhibiting a low variation of standard deviation. This corresponds to a section of core of homogenous composition. It also shows interval profiles and profiles following a moving average trend at varied intervals of 1 mm, 10 cm, and 50 cm. However, interval profile and moving average profile at a scale of 1 mm produce same values and therefore a single profile at 1 mm scale is shared in the figures. This is due to the fact that every line along the core has a depth of approximately 1 mm.. Figure 3.6 shows the same variation of profiles for another core from the file where the standard deviation is highly variable. This suggests heterogeneity in the composition of the core along the length of the core.

3.3.3 Method No. 2 (An improved approach)

Method 2 is an improved version of Method 1 because the processing time is comparatively very low and yet, it maintains a high level of accuracy as seen through comparisons (section 3.3.4). This method involves two steps in addition to step 1 (section 3.3.1).

3.3.3.1 Step 2: Create a base file holding data values for each line

At this intermediate step, a program was written in ENVI/IDL to create a binary file that would hold average and standard deviation values of TBC for each line in each core along the mosaic file (Appendix A-3). This program allows the user to input a mosaic data file of core samples, and the data sheet created in Step 1 (section 3.3.1). The algorithm scans through each pixel in the mosaic file, discards missing (NaN) values, determines the number of lines with valid data in each core and sample spread (spread along the x-axis) of each core, and calculates average and standard deviation of TBC along every line in each core. This is done for all bands or models used to estimate TBC values and the values of average and standard deviation are stored in a binary file (MosaicFilename_BinaryDatafile.dat). Given there are three bands in the mosaic file,

each binary file created includes six sets of data. The program also writes values of start depth, end depth, total depth, no. of lines with valid data, depth per line in mm for each core in the mosaic file on a csv file (MosaicFilename_BaseDatafile.csv). A sample from the base data file is shown in the table 3-3. The two output files created in this step serve to execute the final step of profile creation.

S.No./Core No.	Start Depth(m)	End Depth(m)	Total Depth(m)	No. of Lines	Depth per line(mm)
1	6.6	7.22	0.62	533	1.1632
2	7.22	7.84	0.62	530	1.1698
3	7.84	8.46	0.62	528	1.1742
4	8.46	9.07	0.62	529	1.172
5	9.07	9.69	0.62	534	1.161
6	9.69	10.31	0.62	534	1.161
7	10.31	10.93	0.62	534	1.161
8	10.93	11.55	0.62	534	1.161
9	11.55	12.24	0.69	533	1.2946
10	12.24	12.94	0.69	530	1.3019
11	12.94	13.63	0.69	528	1.3068
12	13.63	14.33	0.69	529	1.3043
13	14.33	15.02	0.69	534	1.2921
14	15.02	15.71	0.69	534	1.2921
15	15.71	16.41	0.69	534	1.2921

Table 3-3:A sample from the base data file created as an intermediate output in the Step 2 of Method 2. Filename: "MRM13-CR0127-SWIR_Core_horizontal_mosaic_TBC_BaseDatafile.csv"

3.3.3.2 Step 3: Create vertical profiles

This is the final step in the process of vertical profile creation using Method No. 2. The idea is to use data stored on files created in Step 2 (section 3.3.3.1) as a base to draw vertical profiles of intended depth. A program was written (Appendix A-4) which upon execution the user is asked to input a mosaic file to draw profiles of, and related base data file and binary data file. It also allows the user to enter the depth/scale of profiles and lets him choose if he wants to view moving average or interval profiles. The program then uses this selection and base data to create vertical profiles arranged horizontally and vertically similar to Method No. 1.

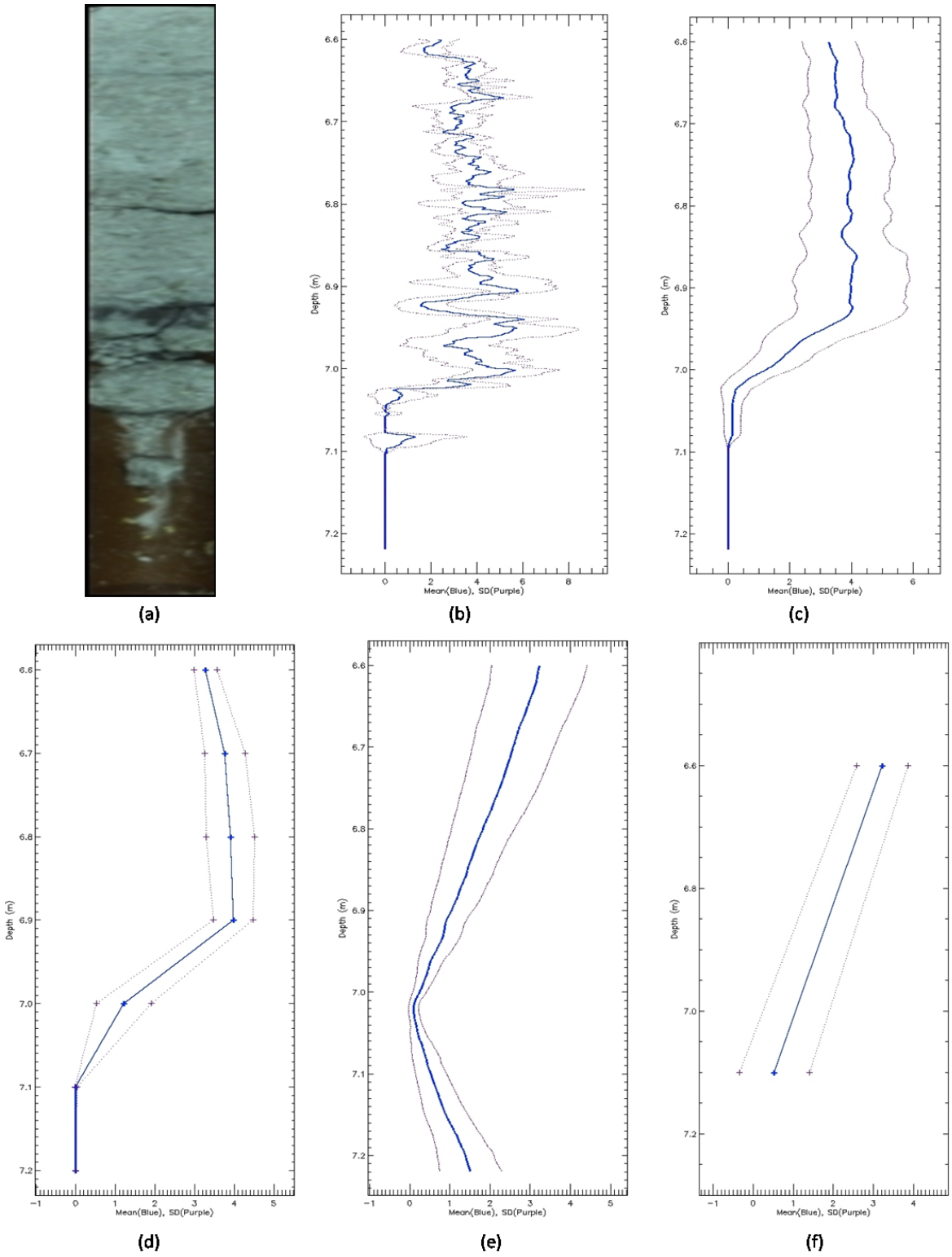

Figure 3-7: Samples from vertical TBC profiles created for " MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" using Method 2. It is an example of profiles showing low homogenous composition of core, a) Image of core no. 1 from the mosaic file, b) Profile of the core at a scale of 1 mm, c) Profile with moving average at a scale of 10 cm, d) Interval Profile at a scale of 10 cm, e) Profile with moving average at a scale of 50 cm, f) Interval Profile at a scale of 50 cm.

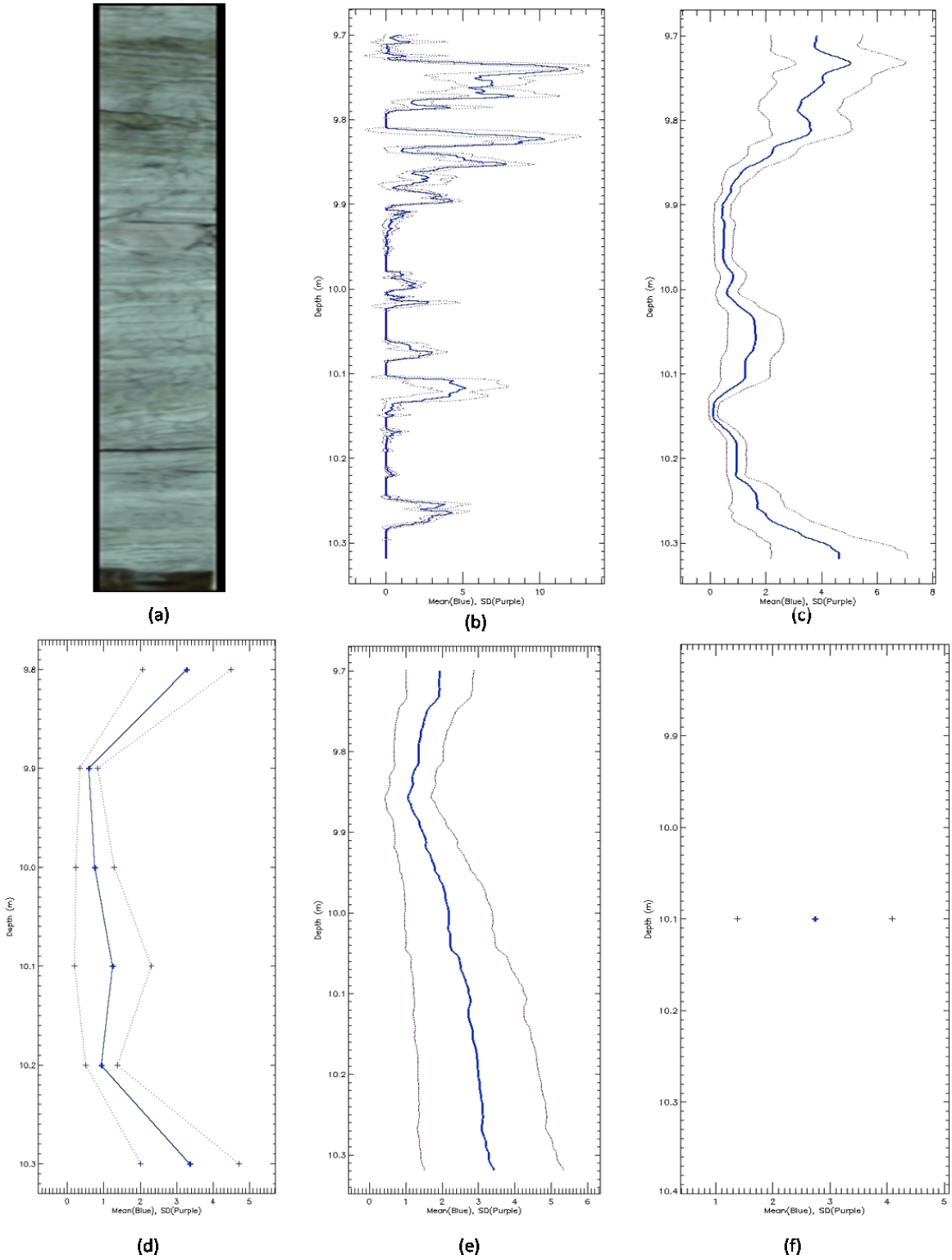

Figure 3-8: Samples from vertical TBC profiles created for " MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" using method no. 2. It is an example of profiles showing heterogeneous composition of core, a) Image of core no. 6 from the mosaic file, b) Profile of the core at a scale of 1 mm, c) Profile with moving average at a scale of 10 cm, d) Interval Profile at a scale of 10 cm, e) Profile with moving average at a scale of 50 cm, f) Interval Profile at a scale of 50 cm,

Examples of profiles created for "MRM13-CR0127-SWIR_Core_horizontal_mosaic_TBC.dat" using this method are shown in figures 3-7 and 3-8. Figure 3-7 shows a homogenous core with low variation in standard deviation and figure 3-8 exhibits a heterogeneous core with a high variation in standard deviation. The examples provide a look at profiles of varied scale of 1 mm, 10 cm, and 50 cm. These profiles show similar trend as seen in figures 3-5 and 3-6 taken from Method No. 1. Similar to Method No. 1, the tool produces a csv file that holds all data values with the depth of each interval, and six images, two for each model/band (given there are three bands). The nomenclature is also same as used in Method No. 1: "MosaicFilename_DepthoftheInterval_MovingAverage/Interval_Data.csv" for csv file and "MosaicFilename_BandName_DepthoftheInterval_HorPlot/VertPlot.jpg" for images. But this method takes significantly less amount of time (table 3-5) as it speeds up the process with an intermediate step of creating base data files. A detailed comparison of the two methods is discussed in the next section.

3.3.4 Comparison between Method No.1 and Method No. 2

Method No. 1 and Method No. 2 vary from each other in the way computations are obtained. Both methods produce similar outputs exhibiting similar trends but the results for the standard deviation vary. As already mentioned, Method no. 1 computes line-by-line values of average and standard deviation, each time the program is executed and therefore the data calculated is quite accurate (but as previously mentioned it also assumes the error of TBC prediction tool). In the second method, line by line values are stored in a file at the intermediate step and these values are used as a base to calculate final data points. The values of standard deviation are calculated by computing standard deviation of base data for desired number of lines. The values of standard deviation are thus not highly accurate; but the values appear to follow the same trend as those in

the first method and therefore is likely acceptable to use. A close comparison between outputs from both methods can be seen in Figure 3-9. Table 3-4 shows comparison in data values of bitumen profiles shown in figure 3-9.

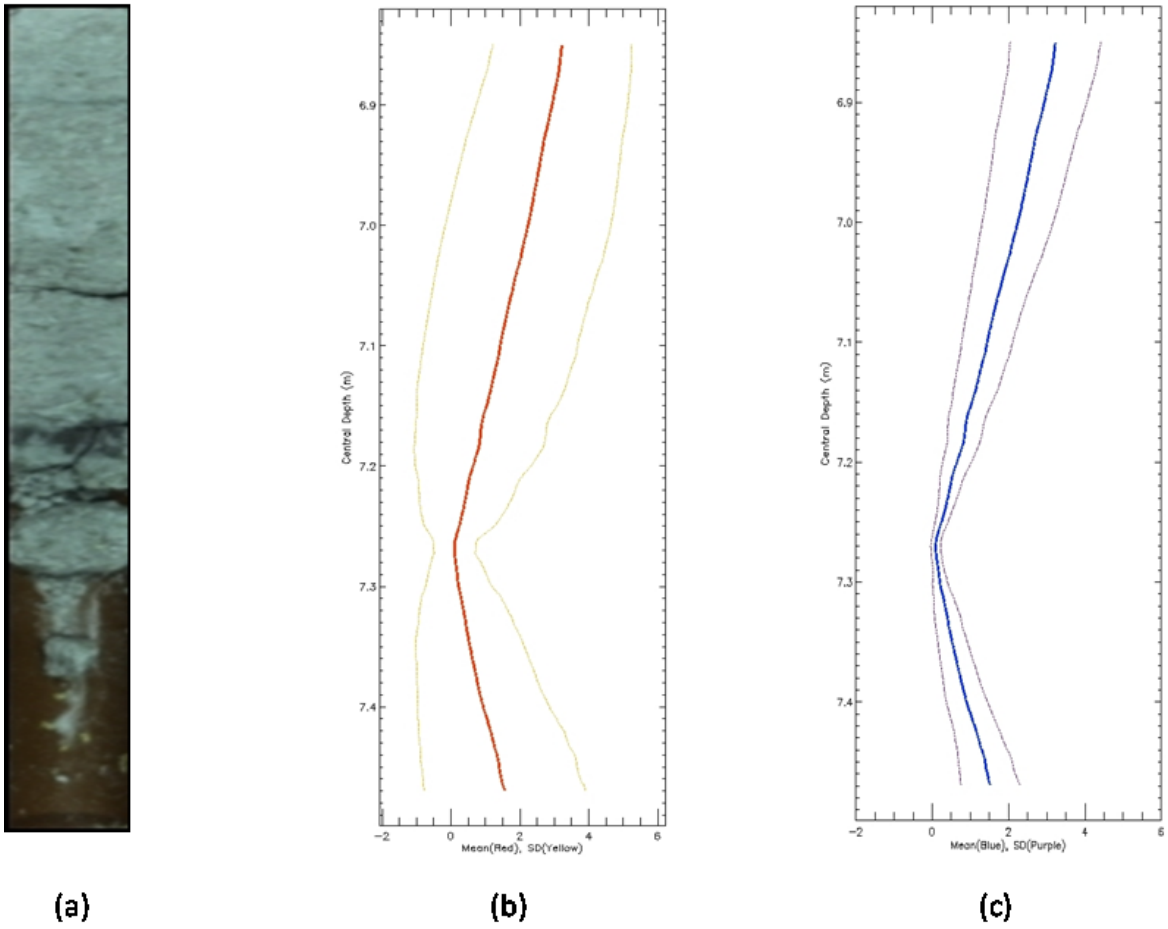

Figure 3-9**:** Comparison between Method No.1 and Method No.2. using moving average profiles drawn at a scale of 50 cm (a) Image of Core No. 1 from "MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" (b) Bitumen profile of the core using Method No.1 (c) Bitumen profile of the core using Method No. 2

S.No	Start Depth(m)	End Depth(m)	Central Depth(m)	Average (Method 1)	Average (Method 2)	% error in Average	SD (Method 1)	SD (Method 2)	% error in SD
1	6.6	7.1	6.85	3.2297	3.2223	0.23%	0.6399	2.0041	213%
2	6.6012	7.1012	6.8512	3.2243	3.2169	0.23%	0.6416	2.0092	213%
62	6.6709	7.1709	6.9209	2.7722	2.7658	0.23%	0.7519	2.253	200%
63	6.6721	7.1721	6.9221	2.7601	2.7537	0.23%	0.7536	2.2532	199%
160	6.7849	7.2849	7.0349	1.9517	1.9472	0.23%	0.8507	2.3669	178%
161	6.7861	7.2861	7.0361	1.9406	1.9361	0.23%	0.8498	2.363	178%
189	6.8186	7.3186	7.0686	1.668	1.6642	0.23%	0.8465	2.2905	171%
190	6.8198	7.3198	7.0698	1.6595	1.6557	0.23%	0.8472	2.2895	170%
227	6.8628	7.3628	7.1128	1.3547	1.3484	0.47%	0.8602	2.224	159%
228	6.864	7.364	7.114	1.3445	1.3383	0.46%	0.8588	2.2185	158%
287	6.9326	7.4326	7.1826	0.8234	0.8196	0.46%	0.7671	1.8801	145%
288	6.9338	7.4338	7.1838	0.8158	0.812	0.47%	0.7641	1.8746	145%
339	6.9931	7.4931	7.2431	0.3064	0.305	0.46%	0.4837	1.1469	137%
340	6.9943	7.4943	7.2443	0.2972	0.2958	0.47%	0.4812	1.1319	135%
384	7.0454	7.5454	7.2954	0.1875	0.1987	5.97%	0.5283	0.8943	69%
385	7.0466	7.5466	7.2966	0.1917	0.2036	6.21%	0.5395	0.9081	68%
485	7.1629	7.6629	7.4129	1.0429	1.0722	2.81%	0.9347	2.0248	117%
486	7.1641	7.6641	7.4141	1.0555	1.0844	2.74%	0.9344	2.0347	118%

Table 3-4: Values and comparison of a range of data points from moving average bitumen profiles of core no. 1 from "MRM13-CR0127-SWIR_Core_horizontal_ mosaic_TBC.dat" drawn at a scale of 50 cm using Method No. 1 and Method No. 2

A key difference between the two methods is the time for creation of bitumen profiles. A comparison between computation times for both methods is shown in table 3-5. The times are recorded on a machine operating on a 3.40GHz processor, with 16 GB installed memory, 64-bit operating system, and using IDL 8.2.1 version. In the case of method no. 2, step no. 2 takes about 10 minutes to complete. Because it is a one-time operation, the only relevant time taken for comparison is from step 3 (section 3.3.4.2). The comparison shows that method no. 2 is a lot faster than method no. 1. It is for this reason that method no. 2 is recommended for use to create bitumen profiles, because there is no significant loss of information and yet the method reduces the processing time from hours to a minute in some cases.

Profile Scale	Profile Type	Time (sec)	
		Method No. 1	Method No. 2
1 mm	-	55.25	55.05
10 cm	Interval	45.98	26.93
10 cm	Moving Average	3611.23	50.51
50 cm	Interval	37.87	25.95
50 cm	Moving Average	14386.93	50.51

Table 3-5: Comparison between bitumen profile creation time of Method No.1 and Method No. 2

3.3.5 Discussion and Limitations

The vertical profiles discussed in this chapter produce a view of TBC that is valuable for core sample analysis. When holes are drilled in oil fields, all sorts of geophysical methods are used to collect data points related to features like resistivity and magnetic susceptibility. The vertical profile tool would allow one to facilitate multidata analysis by providing a comparison to other forms of profiles (e.g. geophysical). However, the methods discussed above suffer from limitations that are important to discuss. When drill core samples are brought to the lab for analysis, some of the core display cracks (shown in figure 3-13 a). These cracks are very narrow but create gaps but the associated data is nevertheless used to generate TBC predictions. The estimation methods assume that there are no cracks in samples (or loss core) and hence the use of every pixel, good or bad, to create bitumen profiles can carry errors. Although, the error as a result of cracks is likely small (e.g. encompassing a relatively small % of pixels), it is still noteworthy. These methods also assume that there is a continuity in core samples and hence the calculated depth of each core of mosaic file is a true value. Sometimes, some portions of drill core samples are lost during drill core recovery or shipping (as shown in figure 3-13 b). It is vital to take into account those gaps in samples and to introduce depth correction in order to minimize errors in the output. Otherwise, this artifact can lead to significant values of error in the results.

Figure 3-10: a) Example of core with cracks as circled in red b) Example of missing core in the box as circled in red

Chapter 4

Conclusions and Recommendations for future work

4.1 Concluding Remarks

The first part of this research project reviewed a slurry sample dehydration experiment and an evaporation experiment performed on tailings samples at the University of Alberta. The review was conducted to qualitatively analyse both experiments to search for areas of improvements. The SIPOC model was used to build a structure of these experiments which were further broken down using an IDEF0 model to study the individual activities and the Process Value Analysis approach was used to assess the value of each activity. The analysis indicated that there were no activities in both experiments that could be eliminated or modified without affecting the process quality as no steps were deemed non-value added. However, a quantitative analysis could be performed to receive a second opinion. The exercise also helped in creating models of the experiments which could be used for future references.

The second part of this research was aimed to create profiles representing variation in TBC as a function of depth along the drill cores. Two methods were successfully used to create interval profiles and moving average profiles at a range of drill core downhole resolutions. The methods were compared on the basis of time and accuracy. The basic method is more accurate than the improved method; however, it takes considerably more time in creating profiles. The results as shown in Chapter 3 can be useful in comparing bitumen profiles to other forms of profiles, for example geophysical profiles. There are certain limitations to these methods, as they do not consider missing drill cores and ignore cracks in the core samples.

4.2 Future Work

We have compared the two tailings experiments using qualitative analysis and based on our analysis, concluded that no major changes are required. It would be interesting to perform a quantitative analysis on these experiments that could provide a solid basis for decision. A quantitative analysis is an analysis performed on a process to measure quantifiable performance dimensions such as time taken to perform a process and cost incurred while performing that process. These parameters are assessed at each level of a process and methods to minimize total time and total cost of the process are explored. The quantitative analysis methodology as described in section 2.7 can be applied in future work.

Earlier we discussed limitations of methods developed to create bitumen profiles of drill cores. These limitations should be addressed in future work performed to create representations. Cracks in the drill cores could be smoothed with the use of image processing in ENVI or image processing methods could attempt to explicitly maps the distribution of crack with the intent to create a "Crak" mask.. Missing drill cores should also be taken into account in the future. A depth correction procedure could be introduced that takes cares of missing sections of drill cores, which would improve the accuracy of these methods significantly.

Chapter 5 References

[1] G. o. Alberta, "Alberta Energy: Oil Sands," [Online]. Available: http://www.energy.gov.ab.ca.login.ezproxy.library.ualberta.ca/OilSands/793.asp. [Accessed 06 June 2015].

[2] G. o. Alberta, "Oil Sands Fact and Statistics: Energy Government of Alberta," [Online]. Available: http://www.energy.gov.ab.ca/oilsands/791.asp. [Accessed 06 June 2015].

[3] Total, "Total: Energies and Expertise, Oil and Gas," 2015. [Online]. Available: http://www.total.com/en/energies-expertise/oil-gas/exploration-production/strategic-sectors/heavy-oil/focus-oil-sands/canadas-oil-sands/production-techniques. [Accessed 06 June 2015].

[4] M. Speta, M. Rivard, J. Feng, M. Lipsett and M. Gingras, "Hyperspectral imaging for the characterization of athabasca oil sands drill core," in *IGARSS*, Melbourne, 2013.

[5] DiCorp, "Oil Sands Coring," [Online]. Available: http://www.di-corp.com/industries/oil-sands-coring. [Accessed June 2015].

[6] K. A. Clark and D. S. Pasternack, "Hot Water Separation of Bitumen from Alberta Bituminous Sand," *INDUSTRIAL AND ENGINEERING CHEMISTRY,* vol. 24, no. 12, p. 1410–1416, 1932.

[7] R. J. Chalaturnyk, J. D. Scott and B. Ozum, "Management of oil sands tailings.," *Petroleum Science and Technology,* vol. 20, no. 9-10, pp. 1025-1046, 2002.

[8] J. Masliyah, Z. Zhou, Z. Xu, J. Czarnecki and H. Hamza, "Understanding Water-Based Bitumen Extractionfrom Athabasca Oil Sands," *The Canadian Journal of Chemical Engineering,* vol. 82, pp. 628-654, August 2004.

[9] V. V. Rogers, K. Libera and M. D. MacKinnon, “Isolation and characterization of naphthenic acids from Athabasca oil sands tailings pond water,” *Chemosphere,* vol. 48, no. 5, pp. 519-527, 2002.

[10] J. Sobkowicz, “Oil Sands Tailings Technology Deployment Roadmap Project Report - Volume 2: Component 1 Results,” Consortium of Oil Sands Tailings Management Consultants (CTMC), Calgary, Alberta, 2012.

[11] M. Cannon, D. Koroluk and Y. Shuai, *Oil Sands Bitumen Recovery,* Department of Chemical Engineering, University of Saskatchewan, 2006-2007.

[12] D. W. Devenny, “A Screening Study of Oil Sand Tailings Technologies and Practices,” Alberta Energy Research Institute (AERI), 2010.

[13] J. R. McDougall, *Sustainable development of oil sands: Challenges in recovery and use, Presentation: Western U.S. Oil Sands Conference,* Salt Lake City, Utah: University of Utah, Institute for Clean and Secure Energy, 2006.

[14] M. Mech, “A comprehensive guide to the Alberta Oil Sands,” 2011.

[15] R. J. Mikula, K. L. Kasperski, R. D. Burns and M. D. MacKinnon, “Nature and fate of oil sands fine tailings,” *Advances in Chemistry Series,* no. 251, p. 677–723, 1996.

[16] C. A. o. P. Producers, “Canada's Oil Sands: Tailings Ponds,” 2015. [Online]. Available: http://www.oilsandstoday.ca/topics/Tailings/Pages/default.aspx. [Accessed 07 June 2015].

[17] A. Fair, “The past, present and future of tailings at Syncrude,” in *International Oil Sands Tailings Conference*, Edmonton, 2008.

[18] I. Entezari, B. Rivard, M. G. Lipsett, J. Feng, G. W. Wilson and R. Mahood, “Deployment of hyperspectral imaging instruments for remote monitoring of soft tailings water content,” in *International Oil Sands Tailings Conference*, Lake Louise, 2014.

[19] S. Energy, “OSQAR,” 25 November 2010. [Online]. Available:

http://osqar.suncor.com/2010/11/in-situ-oil-sands-minings-hotter-younger-sibling.html. [Accessed 06 June 2015].

[20] R. M. Butler and D. J. Stephens, "The gravity drainage of steam-heated heavy oil to parallel horizontal wells," *Journal of Canadian Petroleum Technology,* vol. 20, no. 02, 1981.

[21] R. Butler, "SAGD Comes of AGE!," *The Journal of Canadian Petroleum Technology,* vol. 37, no. 7, pp. 9-12, 1998.

[22] C. A. o. P. Prducers, "Canada's Oil Sands: Recovering the Oil," 2015. [Online]. Available: http://www.oilsandstoday.ca/whatareoilsands/Pages/RecoveringtheOil.aspx. [Accessed 07 June 2015].

[23] T. Chung, W. Bae, J. Lee and B. Jung, "A Review of Practical Experience and Management of the SAGD Process for Oil Sands Development," *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects,* vol. 34, no. 3, pp. 219-226, 2011.

[24] S. Akin and S. Bagci, "A laboratory study of single-well steam-assisted gravity drainage process, doi:10.1016/S0920-4105(01)00145-0," *Journal of Petroleum Science and Engineering,* vol. 32, no. 1, pp. 23-33, 2001.

[25] C.-I. Chang, Hyperspectral Imaging: Techniques for Spectral Detection and Classification, New York: Kluwer Academic/Plenum Publishers, 2003.

[26] A. A. Gowena, C. P. O'Donnell, P. J. Cullen, G. Downey and J. M. Frias, "Hyperspectral imaging – an emerging process analytical tool for food quality and safety control," *Trends in Food Science & Technology,* vol. 18, no. 12, p. 590–598, December 2007.

[27] D. Bannon, "Hyperspectral imaging: Cubes and slices," *Nature photonics,* vol. 3, no. 11, pp. 627-629, 2009.

[28] I. Entezari, B. Rivard, M. Lipsett and G. Wilson, "Application of hyperspectral remote sensing in estimation of oil sands tailings water content," in *5th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Gainesville,

Florida, 2013.

[29] I. Entezari, B. Rivard, M. Lipsett and W. Ward, "Prediction of Water Content and Normalized Evaporation (Pending Publication)," *Canadian Geotechnical Journal.*

[30] J. T. Bulmer and J. Starr, "Syncrude analytical methods for oil sand and bitumen processing.," in *Alberta Oil Sands Technology and Research Authority*, Edmonton, AB, 1979.

[31] M. Speta, B. Rivard, J. Feng, M. Lipsett and M. Gingras, "Hyperspectral imaging for the determination of bitumen content in Athabasca oil sands core samples," *AAPG Bulletin,* pp. 1-15, 2015.

[32] H. J. Vinegar and P. N. Tutunjian, "Method for determining preselected properties of a crude oil". Washington, DC Patent US 5306640 A, 26 april 1994.

[33] R. C. Shaw and B. Kratochvil, "Near-infrared diffuse reflectance analysis of Athabasca oil sand," *Analytical Chemistry,* vol. 62, no. 2, pp. 167-174, 1990.

[34] K. K. Donkor, B. Kratochvil and G. R. Thompson, "Analysis of Athabasca oil sand by near-infrared–diffuse reflectance spectroscopy," *Analyst,* vol. 120, pp. 2713-2717, 1995.

[35] D. Lyder, J. Feng, B. Rivard, A. Galie and E. Cloutis, "Remote bitumen content estimation of Athabasca oil sand from hyperspectral infrared reflectance spectra using Gaussian singlets and derivate of Gaussian wavelets," *Fuel,* vol. 89, pp. 760-767, 2010.

[36] B. Rivard, D. Lyder, J. Feng, A. Gallie and E. Cloutis, "Bitumen content estimation of Athabasca oil sand from braod band infrared reflectance spectra," *Canadian Journal of Chemical Engineering,* vol. 88, no. 5, pp. 830-838, 2010.

[37] S. Camera, "Drill Core Imaging Systems," [Online]. Available: http://www.spectralcameras.com/sisu-rock. [Accessed June 2015].

[38] S. N. Haubrock, S. Chabrillat, C. Lemmnitz and H. Kaufmann, "Surface soil moisture

quantification models from reflectance data under field conditions," *International Journal of Remote Sensing,* vol. 29, no. 1, pp. 3-29, 2008.

[39] A. B. Badiru and S. O. Osisanya, Project management for the oil and gas industry : a world system approach, CRC Press, 2013.

[40] D. o. T. a. Industry, "Gov.UK," [Online]. Available: http://www.businessballs.com/dtiresources/TQM_process_improvement_tools.pdf.

[41] K. Simon, "iSix Sigma," [Online]. Available: http://www.isixsigma.com/tools-templates/sipoc-copis/sipoc-diagram/. [Accessed 15 May 2015].

[42] J. Dorador and R. Young, "Application of IDEF0, IDEF3 and UML methodologies in the creation of information models," *International Journal of Computer Integrated Manufcturing,* vol. 13, no. 5, pp. 430-445, 2000.

[43] P. FIPS, "Integration definition for Function Modeling (IDEF0)," *Federal information processing standards publication,* vol. 183, 1993.

[44] S. Kappes, "Putting your IDEF0 model to work," *Business Process Management Journal,* vol. 3, no. 2, pp. 151-161, 1997.

[45] Investopedia, "Process Value Analysis - PVA," [Online]. Available: http://www.investopedia.com/terms/p/process-value-analysis-pva.asp. [Accessed 09 06 2015].

[46] T. U. o. Minnesota, "Office of Planning and Analysis: Process Mapping & Value Analysis," [Online]. Available: http://www.planning.umn.edu/resources/project-management-resources/process-mapping-value-analysis. [Accessed 09 06 2015].

[47] I. Entezari, B. Rivard, M. Lipsett, J. Feng, G. Wilson and R. Mahood, "Deployment of hyperspectral imaging instruments for remote monitoring of soft tailings water content," in *International Oil Sands Tailings Conference*, Lake Louise, 2014.

[48] J. Feng, B. Rivard, A. Gallie and E. Cloutis, “Feature selection and broad band bitumen content estimation of Athabasca oil sand from infrared reflectance spectra.,” *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution of Remote Sensing (WHISPERS '09),* pp. 1-5, 2009.

[49] M. Dumas, M. L. Rosa, J. Mendling and H. A. Reijers, Fundamentals of Business Process Management, Springer Berlin Heidelberg, 2013.

Appendix A-1

```
;****PROGRAM TO CREATE PROFILE SHEETS***

pro profilesheet
OriginFolder = dialog_pickfile(TITLE = 'Select the folder', /directory)
CD, OriginFolder
SourceFile   =   File_search(OriginFolder,   'BoxandCore',   Count   =   countfiles,
/test_directory )
datfiles = make_array(countfiles, 12, /string)
for i = 0, countfiles-1 do begin
     datfiles[i,*] = file_search(sourcefile[i], '*.dat', count = countdat)
endfor
nbox = (countfiles*12)/3
ncore = (2*countfiles*12)/3
boxfiles = make_array(nbox, /string)
box_sns= INTARR(nbox)
box_snl= INTARR(nbox)
box_snb= INTARR(nbox)
corefiles = make_array(ncore, /string)
core_sns = INTARR(ncore)
core_snl = INTARR(ncore)
core_snb = INTARR(ncore)
box_sns = INTARR(nbox)
box_snl = INTARR(nbox)
box_snb = INTARR(nbox)
boxfiles = datfiles[Where(strmatch(datfiles, '*_Core*') eq 0)]
corefiles= datfiles[Where(strmatch(datfiles, '*_Core*') eq 1)]
boxfiles = boxfiles[sort(boxfiles)]
corefiles = corefiles[sort(corefiles)]
m1box = StrPos(boxfiles[0], 'm', /REVERSE_SEARCH)
m2box = StrPos(boxfiles[0], 'm', m1box-1, /REVERSE_SEARCH)
llimitbox = StrMid(boxfiles[0], m1box-2, 2)+'.'+StrMid(boxfiles[0], m1box+1, 2)
ulimitbox = StrMid(boxfiles[0], m2box-2, 2)+'.'+StrMid(boxfiles[0], m2box+1, 2)
for i=0, ncore-1 do begin
     envi_open_file, corefiles[i], r_fid=sfid, /No_Realize
      if (sfid eq -1) then return
     envi_file_query,sfid,nb=snb, ns=sns, nl=snl, data_type=sdata_type, dims = dims
     core_sns[i]= sns
     core_snb[i] = snb
     core_snl[i] = snl
endfor
openw, datlun, 'FilesDatabase_'+strtrim(ncore,2)+'_CoreFiles.csv' , /get_lun
printf, datlun, 'S.No.,', 'File Name,', 'Total Depth (m),', 'Start Depth (m),', 'End
Depth (m),', $
'No. of Lines,', 'No. of Samples,','No. of Bands,' ,format = '('+strtrim(8,2)+'(A))
for p=0,ncore-1   do begin
     m1core = StrPos(corefiles[p], 'm', /REVERSE_SEARCH)
     m2core = StrPos(corefiles[p], 'm', m1core-1, /REVERSE_SEARCH)
     corenum = StrMid(corefiles[p], StrPos(corefiles[p], '.', /REVERSE_SEARCH)-1,1)
     boxnum = StrMid(corefiles[p], StrPos(corefiles[p], 'Box', /Reverse_Search)+4,1)
     ll_und1 = StrPos(corefiles[p], '_', m1core-1, /reverse_search)
     ll_und2 = StrPos(corefiles[p], '_', m1core-1)
```

```
     ul_und1= StrPos(corefiles[p], '_', m2core-1, /reverse_search)
     ul_und2 = StrPos(corefiles[p], '_', m2core-1)
     llimitcore      =      StrMid(corefiles[p],      ll_und1+1,      m1core-ll_und1-
1)+'.'+StrMid(corefiles[p],$     m1core+1, ll_und2-m1core-1)
     ulimitcore      =      StrMid(corefiles[p],      ul_und1+1,      m2core-ul_und1-
1)+'.'+StrMid(corefiles[p],$     m2core+1, ul_und2-m2core-1)
     print, llimitcore,'*****', ulimitcore
     height = (float(llimitcore)-float(ulimitcore))/8
     uheight = ulimitcore+height*(2*fix(boxnum)+fix(corenum)-3)
     lheight = ulimitcore+height*(2*fix(boxnum)+fix(corenum)-2)
     DatPos = StrPos(corefiles[p], '.dat')
     SepPos = StrPos(corefiles[p], '\', /REVERSE_SEARCH)
     FileName = StrMid(corefiles[p], SepPos+1, DatPos-SepPos-1)
     scorepos = StrPos(corefiles[p], '_')
     Order  =  StrMid(corefiles[p],scorepos+1,StrPos(corefiles[p],  '_',  scorepos+1)-
scorepos-1)
     comb = fix(Order+boxnum+corenum)
     printf,  datlun,  strtrim(comb,2),  filename,height,uheight,  lheight,core_snl[p],
core_sns[p], $      core_snb[p],  format  =  '(I0.0,",",A,  ",",'+strtrim(3,2)+'(F0.2,
",",'+strtrim(3,2)+'(I, ","))'
endfor
free_lun, datlun
for i=0, nbox-1 do begin
     envi_open_file, boxfiles[i], r_fid=sfid, /No_Realize
      if (sfid eq -1) then return
     envi_file_query,sfid,nb=snb, ns=sns, nl=snl, data_type=sdata_type, dims = dims
     box_sns[i]= sns
     box_snb[i] = snb
     box_snl[i] = snl
endfor
openw, datlun, 'FilesDatabase_'+strtrim(nbox,2)+'_BoxFiles.csv' , /get_lun
printf, datlun, 'S.No.,', 'File Name,', 'Total Depth (m),', 'Start Depth (m),', 'End
Depth (m),', $
'No. of Lines,', 'No. of Samples,','No. of Bands,' ,format = '('+strtrim(8,2)+'(A))
for p=0,nbox-1   do begin
     m1box = StrPos(boxfiles[p], 'm', /REVERSE_SEARCH)
     m2box = StrPos(boxfiles[p], 'm', m1box-1, /REVERSE_SEARCH)
     ;corenum = StrMid(corefiles[p], StrPos(corefiles[p], '.', /REVERSE_SEARCH)-1,1)
     boxnum = StrMid(boxfiles[p], StrPos(boxfiles[p], 'Box', /Reverse_Search)+4,1)
     ll_und1 = StrPos(boxfiles[p], '_', m1box-1, /reverse_search)
     ll_und2 = StrPos(boxfiles[p], '_', m1box-1)
     ul_und1= StrPos(boxfiles[p], '_', m2box-1, /reverse_search)
     ul_und2 = StrPos(boxfiles[p], '_', m2box-1)
     llimitbox       =       StrMid(boxfiles[p],       ll_und1+1,       m1box-ll_und1-
1)+'.'+StrMid(boxfiles[p],$      m1box+1, ll_und2-m1box-1)
     ulimitbox       =       StrMid(boxfiles[p],       ul_und1+1,       m2box-ul_und1-
1)+'.'+StrMid(boxfiles[p],$      m2box+1, ul_und2-m2box-1)
     height = (float(llimitbox)-float(ulimitbox))/4
     uheight = ulimitbox+height*(fix(boxnum)-1)
     lheight = ulimitbox+height*fix(boxnum)
     DatPos = StrPos(boxfiles[p], '.dat')
     SepPos = StrPos(boxfiles[p], '\', /REVERSE_SEARCH)
     FileName = StrMid(boxfiles[p], SepPos+1, DatPos-SepPos-1)
     scorepos = StrPos(boxfiles[p], '_')
Order=StrMid(boxfiles[p],scorepos+1,StrPos(boxfiles[p],'_',scorepos+1)-scorepos-1)
```

```
     comb = fix(Order+boxnum)
     printf,  datlun,  strtrim(comb,2),  filename,height,uheight,  lheight,box_snl[p],
box_sns[p],$ box_snb[p],    format    =    '(I0.0,",",A,    ",",'+strtrim(3,2)+'(F0.2,
",",'+strtrim(3,2)+'(I, ","))'
endfor
free_lun, datlun
end
```

Appendix A-2

```
;****PROGRAM TO CREATE BITUMEN PROFILES - METHOD 1***

pro PSD_TBC_Vertical_Profiling_mosaic_event, Event
 wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ?  $
 widget_info(Event.id, /tree_root) : event.id)
  wWidget =  Event.top
  case wTarget of
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_Input_Browse'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_TEXT_Resolution'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_MAv_check'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_Start'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_0'): begin
    end
    else:
  endcase
end

pro PSD_TBC_Vertical_Profiling_mosaic, GROUP_LEADER=wGroup, $
_EXTRA=_VWBExtra_

WID_BASE_0 = Widget_Base( GROUP_LEADER=wGroup, UNAME='WID_BASE_0'  $
      ,XOFFSET=5 ,YOFFSET=5 ,SCR_XSIZE=400 ,SCR_YSIZE=300 ,TITLE='Creating Profiles of
Mosaic Images' $
       ,SPACE=3 ,XPAD=3 ,YPAD=3)

WID_LABEL_0 = Widget_Label(WID_BASE_0, UNAME='WID_LABEL_0'  $
      ,XOFFSET=19 ,YOFFSET=20 ,SCR_XSIZE=56 ,SCR_YSIZE=18  $
      ,/ALIGN_LEFT ,VALUE='Input File:')

WID_TEXT_Input_File = Widget_Text(WID_BASE_0,  $
      UNAME='WID_TEXT_Input_File' ,FRAME=1 ,XOFFSET=78 ,YOFFSET=17  $
      ,SCR_XSIZE=240 ,SCR_YSIZE=24 ,/EDITABLE ,XSIZE=20 ,YSIZE=1)

WID_BUTTON_Input_Browse = Widget_Button(WID_BASE_0,  $
      UNAME='WID_BUTTON_Input_Browse' ,XOFFSET=320 ,YOFFSET=16  $
      ,SCR_XSIZE=57 ,SCR_YSIZE=25 ,EVENT_FUNC='Browse_Input_file'  $
      ,/ALIGN_CENTER ,VALUE='Browse')

WID_LABEL_3 = Widget_Label(WID_BASE_0, UNAME='WID_LABEL_3'  $
      ,XOFFSET=100 ,YOFFSET=84 ,SCR_XSIZE=130 ,SCR_YSIZE=18  $
      ,/ALIGN_LEFT ,VALUE='Length of Interval (cm):')

WID_TEXT_Resolution = Widget_Text(WID_BASE_0,  $
      UNAME='WID_TEXT_Resolution' ,XOFFSET=230 ,YOFFSET=80  $
      ,SCR_XSIZE=44 ,SCR_YSIZE=21 ,EVENT_FUNC='Get_Resolution'  $
      ,/EDITABLE ,/ALL_EVENTS ,XSIZE=20 ,YSIZE=1)
```

```
WID_BASE_1 = Widget_Base(WID_BASE_0, UNAME='WID_BASE_1' ,XOFFSET=70  $
      ,YOFFSET=150 ,TITLE='IDL' ,COLUMN=1 ,/NONEXCLUSIVE)

WID_BUTTON_MAv_check = Widget_Button(WID_BASE_1,  $
      UNAME='WID_BUTTON_MAv_check' ,SCR_XSIZE=250 ,SCR_YSIZE=26  $
      ,EVENT_FUNC='Check_MAv_status' ,/ALIGN_LEFT ,VALUE='Calculate'+ $
      ' moving average and standard deviation?')

WID_BUTTON_Start = Widget_Button(WID_BASE_0,  $
      UNAME='WID_BUTTON_Start' ,FRAME=1 ,XOFFSET=105 ,YOFFSET=200  $
      ,SCR_XSIZE=70 ,SCR_YSIZE=36 ,EVENT_FUNC='Start_process'  $
      ,/ALIGN_CENTER ,VALUE='Start')

WID_BUTTON_0 = Widget_Button(WID_BASE_0, UNAME='WID_BUTTON_0'  $
      ,FRAME=1 ,XOFFSET=220 ,YOFFSET=200 ,SCR_XSIZE=70 ,SCR_YSIZE=36  $
      ,EVENT_FUNC='Cancel_Process' ,/ALIGN_CENTER ,VALUE='Cancel')

Widget_Control, /REALIZE, WID_BASE_0
     sState={WID_TEXT_Input_File:WID_TEXT_Input_File,       $
             file_Data:0L,                                  $
             WID_TEXT_Resolution:WID_TEXT_Resolution,       $
             flag_MAv:0}
WIDGET_CONTROL, WID_BASE_0, Set_UVALUE=sState, /NO_COPY
XManager, 'PSD_TBC_Vertical_Profiling_mosaic', WID_BASE_0, /NO_BLOCK

end

function Browse_Input_file, Event
 WIDGET_CONTROL, Event.top, GET_UVALUE=sState, /NO_COPY
      lfname = Dialog_Pickfile(/MUST_EXIST, TITLE='Select a data file',
File='image.dat',FILTER = '*.*')
        IF lfname NE "" THEN BEGIN
             envi_open_file, lfname, r_fid=lfid,/NO_REALIZE
             if (lfid ne -1) then begin
                              sState.file_Data=lfid
                         widget_control, sState.WID_TEXT_Input_File,
Set_Value=lfname
                         endif
        ENDIF
     WIDGET_CONTROL, Event.top, SET_UVALUE=sState, /NO_COPY
     return, Event ; By Default, return the event.
end
function get_Resolution, Event
      return, Event ; By Default, return the event.
end
function Check_MAv_status, Event
 WIDGET_CONTROL, Event.top, GET_UVALUE=sState, /NO_COPY
       CASE sState.flag_MAv OF
          0: sState.flag_MAv=1
          1: sState.flag_MAv=0
       ENDCASE
   WIDGET_CONTROL, Event.top, SET_UVALUE=sState, /NO_COPY
     return, Event ; By Default, return the event.
end
function Cancel_Process, Event
```

```
 widget_control,event.top,/destroy
     return, Event ; By Default, return the event.
end
function Start_process, Event
WIDGET_CONTROL, Event.top, GET_UVALUE=sState, /NO_COPY
      widget_control,  sState.WID_TEXT_Input_File, Get_Value=input_filename
      widget_control,  sState.WID_TEXT_Resolution, Get_Value=Resolution
     fileId= sState.file_Data
Resolution=float(Resolution)
DepInterv = Resolution[0]
profile_tbc_psd_mosaic, input_filename, DepInterv, sState.Flag_MAv
end
pro profile_tbc_psd_mosaic, sfname, DepInterval, flag
;flag = 1 means moving average
;flag = 1
IF sfname NE "" THEN BEGIN
mosaic = STRMATCH(sfname, '*mosaic*', /FOLD_CASE)

;***********Code for Mosaic File starts here*************************
IF mosaic eq 1 then begin
If flag eq 0 then begin
mfname = sfname
mdir = file_dirname(mfname, /mark_directory)
mfile = file_basename(mfname, '.dat')
print, mfile
cd, mdir
file_mkdir, mfile
T = SYSTIME(1)
envi_open_file, mfname, r_fid=mfid, /No_Realize
             if (mfid eq -1) then return
envi_file_query,mfid,nb=snb, ns=sns, nl=snl, file_type=sfile_type,
data_type=sdata_type,$
wl=swl, xstart= sxstart, ystart= systart, dims=dims, bnames = bnames
print, sxstart, systart
mdat = envi_get_slice(FID=mfid, LINE=10, POS=0, XE=sns-1 , XS= sxstart)
ind_start= make_array(500,/long)
ind_end = make_array(500, /long)
ind_end(0)= sns
value = 0
for i = 0, sns-1 do begin
     if mdat(sns-1-i) ne mdat(sns-1-i) then begin
       if count eq 1 then begin
              value = value+1
              count = count+1
       endif
       ind_end(value) = sns-i-1
     endif else begin
       ind_start(value)= sns-i
       count = 1
     endelse
endfor
print, value
sx_start = ind_start(0:value-1) ; array with starting sample number of individual file
sx_end = ind_end(0:value-1) ; array with ending sample number of individual file
```

```
;findng effective number of lines
line_temp = 0
snl_eff = 0
for i=0, snl-1 do begin
     val = envi_get_slice(FID=mfid, LINE = i, POS = 0, XE = 10, XS = 10)
     if val ne val then begin
       line_temp = i+1
     endif else begin
       line_start = line_temp
       line_end = i
       snl_eff= snl_eff+1
     endelse
endfor
pfile = dialog_pickfile(title = 'Please select a profile sheet')
und1 = StrPos(pfile, '_')
und2 = StrPos(pfile, '_', /REVERSE_SEARCH)
num = Strmid(pfile, und1+1, und2-und1-1)
data = strarr(num+1)
if num ne value then begin
  result = dialog_message("Wrong Profile Sheet", /error)
  return
endif
openr, dlun,pfile, /get_lun
     readf, dlun, data
free_lun, dlun
pdata= strarr(num+1,8)
s= strarr(num+1,8)
for i = 0, num do begin
     s[i,*] = STRSPLIT(data[i], ',', /EXTRACT)
endfor
start_ht = findgen(num)
end_ht = findgen(num)
height = findgen(num)
DepInterval = 0.1 ;in cm
Resolution = DepInterval
comb = fix(s[1:num,0])
order = sort(comb)
ndim = findgen(num)
ndim_int = lindgen(num)
ndim_dec = findgen(num)
n = indgen(num)
LineTo_mm = findgen(num)
cmto_line = findgen(num)
for i = 0, num-1 do begin
     height[i] = s[order(i)+1, 2]
     start_ht[i] = s[order(i)+1, 3]
     end_ht[i] = s[order(i)+1, 4]
endfor

;defining height intervals between cores
for i =0, num-1 do begin
     if i ne 0 then begin
       if start_ht[i] ne end_ht[i-1] then begin
              height[i:i+7] = (float(end_ht[i+7])-float(end_ht[i-1]))/8.0
       endif
```

```
      endif
      LineTo_mm[i] = (height[i]*1000)/snl_eff
      ConvFactor = LineTo_mm[i]/10 ; a line equivalent in cms
      cmto_line[i] = 1/convfactor ;a cm equivalent in lines ; modification
      n[i] = round(Resolution/ConvFactor) ;number of lines in each set of the core
      if n[i] eq 0 then n[i] = 1
      ndim[i] = float(snl_eff/n[i]) ;number of sets in the core
      ndim_int[i] = floor(ndim[i])
      ndim_dec[i] = ndim[i]-ndim_int[i]
endfor
arrdim = max(ndim_int)+2
avg = make_array(num,arrdim,snb,/Double)
sd = make_array(num,arrdim,snb,/Double)
sdplus = make_array(num,arrdim,snb,/Double)
sdminus = make_array(num,arrdim,snb,/Double)
;yvalues = make_array(num,arrdim,/INTEGER)
count = make_array(num, /Long)
p = 0
lines_need = 0
      for m = 0, num-1 do begin
       if p gt 2 then begin ;modification begins
              p = p-1
              continue
       endif
       if (snl_eff-lines_need) eq 0 then begin
              line_need = 0
              continue
       endif
      a1 = sx_end(m)-sx_start(m)+1
       if m ne num-1 then begin
              a2 = sx_end(m+1)-sx_start(m+1)+1
              if a1 ge a2 then begin
                     a1 = a1
              endif else begin
                     a1 = a2
              endelse
       endif
       ns = sx_end(m)-sx_start(m)+1
       temp0 = make_array((n[m]*a1*10), /Double)
       temp1 = make_array((n[m]*a1*10), /Double)
       temp2 = make_array((n[m]*a1*10), /Double)
       count[m]=0
       print, m
       for l=line_start+lines_need, line_end, n[m] do begin; modification
              k=0
              for i=l, l+n[m]-1 do begin
                     if i le line_end then begin
                            si = k*ns
                            so = (k+1)*(ns)-1
temp0[si:so] = envi_get_slice(fid=mfid, line =i, pos = 0, xe =sx_end[m]-1, xs
=sx_start[m]-1)
temp1[si:so] = envi_get_slice(fid=mfid, line =i, pos = 1, xe =sx_end[m]-1, xs
=sx_start[m]-1)
temp2[si:so] = envi_get_slice(fid=mfid, line =i, pos = 2, xe =sx_end[m]-1, xs
=sx_start[m]-1)
```

```
                        k = k+1
                        if mean(temp0[si:so]) ne mean(temp0[si:so]) then begin
                              temp0[si:so]=0
                              temp1[si:so]=0
                              temp2[si:so]=0
                              ;k = k-1
                              si = si -ns
                              so = so -ns
                        endif
                  endif
                  if m ne num-1 then begin
                        if i eq line_end+1 then begin
                              p =1
                              cond =1
                              while (cond eq 1) do begin
                                    cond = 0
                                    ns = sx_end(m+p)-sx_start(m+p)+1
                                    if p eq 1 then begin
                        lines_need = round((lineto_mm[m]/lineto_mm[m+1])*(n[m]-k))
                                    endif else begin
lines_need = round(cmto_line[m+p]*(resolution-(total(height[m+1:m+p-1])*100)-$
(k*lineto_mm[m]*0.1)))
                                    endelse
                                    if lines_need gt snl_eff then begin
                                          lines_need = snl_eff
                                          cond = 1
                                    endif
                                    for j=line_start, line_start+lines_need-1 do
begin
                                          si=so+1
                                          so = si+ns-1
temp0[si:so] = envi_get_slice(fid=mfid,line =j,pos = 0,xe =sx_end[m+p]-1, xs
=sx_start[m+p]-1)
temp1[si:so] = envi_get_slice(fid=mfid, line=j,pos= 1, xe =sx_end[m+p]-1, xs
=sx_start[m+p]-1)
temp2[si:so] = envi_get_slice(fid=mfid, line =j,pos=2, xe =sx_end[m+p]-1, xs
=sx_start[m+p]-1)
                              if mean(temp0[si:so]) ne mean(temp0[si:so]) then
begin
                                                temp0[si:so]=0
                                                temp1[si:so]=0
                                                temp2[si:so]=0
                                                si = si -ns
                                                so = so -ns
                                          endif
                                    endfor
                                    p=p+1
                                    if m+p gt num-1 then break
                              endwhile
                        endif
                  endif
                  if i gt line_end then break
            endfor
            if so lt 0 then continue
            avg[m,count[m],0] = Mean(temp0[0:so])
```

```
              avg[m,count[m],1] = Mean(temp1[0:so])
              avg[m,count[m],2] = Mean(temp2[0:so])
              sd[m,count[m],0] = stddev(temp0[0:so])
              sd[m,count[m],1] = stddev(temp1[0:so])
              sd[m,count[m],2] = stddev(temp2[0:so])
              sdplus[m,count[m],*] = avg[m,count[m],*]+sd[m,count[m],*]
              sdminus[m,count[m],*] = avg[m,count[m],*]-sd[m,count[m],*]
              count[m] = count[m]+1
       endfor
      endfor
set_total = Total(count)
mpos = where(count gt 0, count_pos)
avg = strtrim(avg,2)
sd = strtrim(sd,2)
sdplus = strtrim(sdplus,2)
sdminus = strtrim(sdminus,2)
cent_depth = make_array(num, max(count), /float)
st_depth = make_array(num, max(count), /float)
end_depth = make_array(num, max(count), /float)
st_depth[0,0] = start_ht[0]
startdp = st_depth[0,0]
p=long(0)
cd, mfile
openw, datlun, strtrim(mfile, 2)+'_'+strtrim(DepInterval,2)+'cms_Data.csv', /get_lun
printf, datlun, 'S.No.,', 'Start Depth(m),', 'End Depth(m),', 'Central Depth(m),', $
'Average 1,', 'Average 2,', 'Average 3,', 'SD1,', 'SD2,', 'SD3,', format =
'('+strtrim(10,2)+'(A))
for m = 0,num-1 do begin
      if count[m] eq 0 then continue
      for j = 0, count[m]-1 do begin
       end_depth[m,j] = startdp+(float(resolution)/100)
       if m eq (num-1) && j eq (count[m]-1) then begin
              end_depth[m,j] = end_ht[num-1]
       endif
       cent_depth[m,j] = (startdp+end_depth[m,j])/2
       printf, datlun, p+1, startdp, end_depth[m,j], cent_depth[m,j],avg[m,j,*],
sd[m,j,*], $
       format = '(I32.0,",",'+strtrim(9,2)+'(F0.4,","))'
       p=p+1
       if p lt set_total then begin
              if j lt count[m]-1 then begin
                     st_depth[m,j+1] = end_depth[m,j]
                     startdp = st_depth[m,j+1]
              endif else begin
                     st_depth[m+1,0] = end_depth[m,j]
                     startdp = st_depth[m+1,0]
              endelse
       endif
      endfor
endfor
free_lun, datlun
PRINT, SYSTIME(1) - T, 'Seconds'
thisDevice = !D.Name
Set_Plot, 'Z'
Erase
```

```
Device, Set_Resolution=[54000,900], Set_Pixel_Depth=24, Decomposed=0
;Device, Set_Resolution=[450,65500], Set_Pixel_Depth=24, Decomposed=0
loadct, 39
for grph = 0, 2 do begin
     !P.MULTI = [0,count_pos,1]
     for m =num-1, 0, -1 do begin  ; change to num-1, 0
      if count[m] eq 0 then continue
      plot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),$ ;yrange =
[end_depth[m,(count[m]-1)],st_depth[m,0]]
      yrange = [max(float(cent_depth(m,0:(count[m]-
1))))+0.03,min(float(cent_depth(m,0:(count[m]-1))))-0.03 ] ,$
      bACKGROUND = 255, xtitle = 'Mean(Red), SD(Yellow)' , ytitle = 'Central Depth
(m)', ystyle = 1,xstyle = 1, color = 0,charsize = 1.75,$
      xrange = [min(float(sdMINUS(m,0:(count[m]-1), grph)))-1.0,
max(float(sdPLUS(m,0:(count[m]-1), grph)))+1.0] , thick =2, /nodata
      oplot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color = 227,
thick = 3
      oplot, sdPLUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1, thick = 2
      oplot, sdMINUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1, thick = 2
     endfor
     snapshot = TVRD(True=1)
!p.multi = 0
Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(DepInterval,2)+'cms_HorPlot.j
pg', snapshot, True=1, Quality=100
endfor
Set_Plot, thisDevice

thisDevice = !D.Name
Set_Plot, 'Z'
Erase
Device, Set_Resolution=[450,65500], Set_Pixel_Depth=24, Decomposed=0
loadct, 39
for grph = 0, 2 do begin
     !P.MULTI = [0,1,count_pos]
     for m =0,num-1,1 do begin  ; change to num-1, 0
      if count[m] eq 0 then continue
      plot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),$ ;yrange =
[end_depth[m,(count[m]-1)],st_depth[m,0]]
      yrange = [max(float(cent_depth(m,0:(count[m]-
1))))+0.03,min(float(cent_depth(m,0:(count[m]-1))))-0.03 ] ,$
      bACKGROUND = 255, xtitle = 'Mean(Red), SD(Yellow)' , ytitle = 'Central Depth
(m)', ystyle = 1,xstyle = 1, color = 0,charsize = 1.75,$
      xrange = [min(float(sdMINUS(m,0:(count[m]-1), grph)))-1.0,
max(float(sdPLUS(m,0:(count[m]-1), grph)))+1.0] , thick =2, /nodata
      oplot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color = 227,
thick = 3
     ;polyfill,[sdminus(m,0:(count[m]-1), grph), reverse(sdplus(m,0:(count[m]-1),
grph))], [cent_depth(m,0:(count[m]-1)), reverse(cent_depth(m,0:(count[m]-1)))], color
= 229
      oplot, sdPLUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1, thick = 2
```

```
      oplot, sdMINUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1,  thick =2
     endfor
     snapshot = TVRD(True=1)
!p.multi = 0
Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(DepInterval,2)+'cms_VertPlot.
jpg', snapshot, True=1, Quality=100
endfor
Set_Plot, thisDevice
PRINT, SYSTIME(1) - T, 'Seconds'
end if else begins
mdir = file_dirname(mfname, /mark_directory)
mfile = file_basename(mfname, '.dat')
print, mfile
cd, mdir
file_mkdir, mfile
T = SYSTIME(1)

envi_open_file, mfname, r_fid=mfid, /No_Realize
             if (mfid eq -1) then return
envi_file_query,mfid,nb=snb, ns=sns, nl=snl, file_type=sfile_type,
data_type=sdata_type,$ wl=swl, xstart= sxstart, ystart= systart, dims=dims, bnames =
bnames
print, sxstart, systart
mdat = envi_get_slice(FID=mfid, LINE=10, POS=0, XE=sns-1 , XS= sxstart)
ind_start= make_array(500,/long)
ind_end = make_array(500, /long)
ind_end(0)= sns
value = 0
for i = 0, sns-1 do begin
     if mdat(sns-1-i) ne mdat(sns-1-i) then begin
       if count eq 1 then begin
              value = value+1
              count = count+1
       endif
       ind_end(value) = sns-i-1
     endif else begin
       ind_start(value)= sns-i
       count = 1
     endelse
endfor
sx_start = ind_start(0:value-1) ; array with starting sample number of each individual
core starting from rightmost core
sx_end = ind_end(0:value-1) ; array with ending sample number of each individual core
starting from rightmost core
;findng effective number of lines
line_temp = 0
snl_eff = 0
for i=0, snl-1 do begin
     val = envi_get_slice(FID=mfid, LINE = i, POS = 0, XE = 10, XS = 10)
     if val ne val then begin
       line_temp = i+1
     endif else begin
       line_start = line_temp
```

```
       line_end = i
       snl_eff= snl_eff+1
     endelse
endfor
pfile = dialog_pickfile(title = 'Please select a profile sheet')
und1 = StrPos(pfile, '_')
und2 = StrPos(pfile, '_', /REVERSE_SEARCH)
num = Strmid(pfile, und1+1, und2-und1-1)
data = strarr(num+1)
if num ne value then begin
  result = dialog_message("Wrong Profile Sheet", /error)
  return
endif
openr, dlun,pfile, /get_lun
     readf, dlun, data
free_lun, dlun
pdata= strarr(num+1,8)
s= strarr(num+1,8)
for i = 0, num do begin
     s[i,*] = STRSPLIT(data[i], ',', /EXTRACT)
endfor
start_ht = findgen(num)
end_ht = findgen(num)
height = findgen(num)
DepInterval = 10;in cm
Resolution = DepInterval
comb = fix(s[1:num,0])
order = sort(comb)
ndim = findgen(num)
ndim_int = lindgen(num)
ndim_dec = findgen(num)
n = indgen(num)
LineTo_mm = findgen(num)
cmto_line = findgen(num)
for i = 0, num-1 do begin
     height[i] = s[order(i)+1, 2]
     start_ht[i] = s[order(i)+1, 3]
     end_ht[i] = s[order(i)+1, 4]
endfor
;defining height intervals between cores
for i =0, num-1 do begin
     if i ne 0 then begin
       if start_ht[i] ne end_ht[i-1] then begin
              height[i:i+7] = (float(end_ht[i+7])-float(end_ht[i-1]))/8.0
       endif
     endif
     LineTo_mm[i] = (height[i]*1000)/snl_eff
     ConvFactor = LineTo_mm[i]/10 ; a line equivalent in cms
     cmto_line[i] = 1/convfactor ;a cm equivalent in lines
     n[i] = round(Resolution/ConvFactor) ;number of lines in each set of the ith core
endfor
arrdim = snl_eff ;changed
avg = make_array(num,arrdim,snb,/Double)
sd = make_array(num,arrdim,snb,/Double)
sdplus = make_array(num,arrdim,snb,/Double)
```

```
sdminus = make_array(num,arrdim,snb,/Double)
count = make_array(num,/long)
for m = 0, 7 do begin ; change to 0, num-1
       a1 = sx_end(m)-sx_start(m)+1
       if m ne num-1 then begin
              a2 = sx_end(m+1)-sx_start(m+1)+1
              if a1 lt a2 then begin
                     a1 = a2
              endif
       endif
       ns = sx_end(m)-sx_start(m)+1
       temp0 = make_array((arrdim*a1*100), /Double)
       temp1 = make_array((arrdim*a1*100), /Double)
       temp2 = make_array((arrdim*a1*100), /Double)
       count[m]=0
       l=line_start
       while l le line_end do begin
              ;print, 'l =', l
              k=0
              for i=l, l+n[m]-1 do begin
                     ns = sx_end(m)-sx_start(m)+1
                     if i le line_end then begin
                            print, i
                            si = k*ns
                            so = (k+1)*(ns)-1
temp0[si:so] = envi_get_slice(fid=mfid, line =i, pos = 0, xe =sx_end[m]-1, xs
=sx_start[m]-1)
temp1[si:so] = envi_get_slice(fid=mfid, line =i, pos = 1, xe =sx_end[m]-1, xs
=sx_start[m]-1)
temp2[si:so] = envi_get_slice(fid=mfid, line =i, pos = 2, xe =sx_end[m]-1, xs
=sx_start[m]-1)
                            k = k+1
                            if mean(temp0[si:so]) ne mean(temp0[si:so]) then begin
                                   temp0[si:so]=0
                                   temp1[si:so]=0
                                   temp2[si:so]=0
                                   k = k-1
                                   si = si -ns
                                   so = so -ns
                            endif
                     endif
                     if m ne num-1 then begin
                            if i eq line_end+1 then begin
                                   p =1
                                   cond =1
                                   while (cond eq 1) do begin
                                          cond = 0
                                          ns = sx_end(m+p)-sx_start(m+p)+1
                                          if p eq 1 then begin
                            lines_needed = round((lineto_mm[m]/lineto_mm[m+1])*(n[m]-
k))
                                          endif else begin
lines_needed = round(cmto_line[m+p]*(resolution-(total(height[m+1:m+p-1])*100)-
(k*lineto_mm[m]*0.1)))
                                          endelse
```

```
                                        if lines_needed gt snl_eff then begin
                                               lines_needed = snl_eff
                                               cond = 1
                                        endif
                                    for j=line_start, line_start+lines_needed-1 do begin
                                               si=so+1
                                               so = si+ns-1
temp0[si:so] = envi_get_slice(fid=mfid, line=j, pos=0, xe =sx_end[m+p]-1, xs
=sx_start[m+p]-1)
temp1[si:so] = envi_get_slice(fid=mfid, line =j, pos=1,xe =sx_end[m+p]-1, xs
=sx_start[m+p]-1)
temp2[si:so] = envi_get_slice(fid=mfid, line =j, pos=2,xe =sx_end[m+p]-1, xs
=sx_start[m+p]-1)
                            if mean(temp0[si:so]) ne mean(temp0[si:so]) then begin
                                                    temp0[si:so]=0
                                                    temp1[si:so]=0
                                                    temp2[si:so]=0
                                                    si = si -ns
                                                    so = so -ns
                                               endif
                                        endfor
                                        p=p+1
                                    endwhile
                            endif
                      endif
                      if i gt line_end then break
              endfor
              avg[m,count[m],0] = Mean(temp0[0:so])
              avg[m,count[m],1] = Mean(temp1[0:so])
              avg[m,count[m],2] = Mean(temp2[0:so])
              sd[m,count[m],0] = stddev(temp0[0:so])
              sd[m,count[m],1] = stddev(temp1[0:so])
              sd[m,count[m],2] = stddev(temp2[0:so])
              sdplus[m,count[m],*] = avg[m,count[m],*]+sd[m,count[m],*]
              sdminus[m,count[m],*] = avg[m,count[m],*]-sd[m,count[m],*]
              count[m] = count[m]+1
              l = l+1
       endwhile
     endfor
set_total = Total(count)
mpos = where(count gt 0, count_pos)
avg = strtrim(avg,2)
sd = strtrim(sd,2)
sdplus = strtrim(sdplus,2)
sdminus = strtrim(sdminus,2)
cent_depth = make_array(num, max(count), /float)
st_depth = make_array(num, max(count), /float)
end_depth = make_array(num, max(count), /float)
st_depth[0,0] = start_ht[0]
p=long(0)
cd, mfile
openw, datlun, strtrim(mfile,
2)+'_'+strtrim(DepInterval,2)+'cms__MovingAverage_Data.csv', /get_lun
printf, datlun, 'S.No.,', 'Start Depth(m),', 'End Depth(m),', 'Central Depth(m),', $
```

```
'Average 1,', 'Average 2,', 'Average 3,', 'SD1,', 'SD2,', 'SD3,', format =
'('+strtrim(10,2)+'(A))
check =0
brk =0
for m = 0,num-1 do begin
     for j = 0, count[m]-1 do begin
      end_depth[m,j] = st_depth[m,j]+(float(resolution)/100)
      if end_depth[m,j] gt end_ht[num-1] then begin
            end_depth[m,j] = end_ht[num-1]
      endif
      cent_depth[m,j] = (st_depth[m,j]+end_depth[m,j])/2
     printf, datlun, p+1, st_depth[m,j], end_depth[m,j], cent_depth[m,j],avg[m,j,*],$
     sd[m,j,*],format = '(I32.0,",",'+strtrim(9,2)+'(F0.4,","))'
      p=p+1
      if p lt set_total then begin
            if j lt count[m]-1 then begin
                  ;st_depth[m,j+1] = st_depth[m,j]+0.01
                  st_depth[m,j+1] = st_depth[m,j]+(LineTo_mm[m]/1000);changed
                  check = st_depth[m,j+1]
            endif else begin
                  ;st_depth[m+1,0] = st_depth[m,j]+0.01
                  st_depth[m+1,0] = st_depth[m,j]+(LineTo_mm[m]/1000) ;changed
                  check = st_depth[m+1,0]
            endelse
      endif
      if check ge end_ht[num-1] then begin
            brk =1
            break
      endif
     endfor
     if brk eq 1 then break
endfor
free_lun, datlun
PRINT, SYSTIME(1) - T, 'Seconds'
thisDevice = !D.Name
Set_Plot, 'Z'
Erase
Device, Set_Resolution=[54000,900], Set_Pixel_Depth=24, Decomposed=0
loadct, 39
print, count_pos
for grph = 0, 2 do begin
     !P.MULTI = [0,count_pos,1]
     for m =num-1, 0, -1 do begin  ; change to num-1, 0
      print, count[m]
      ;if count[m] eq 0 then continue
      plot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),$
      yrange = [max(float(cent_depth(m,0:(count[m]-
1))))+0.03,min(float(cent_depth(m,0:(count[m]-1))))-0.03 ] ,$
      bACKGROUND = 255, xtitle = 'Mean(Red), SD(Yellow)' , ytitle = 'Central Depth
(m)', ystyle = 1,xstyle = 1, color = 0,charsize = 1.75,$
      xrange = [min(float(sdMINUS(m,0:(count[m]-1), grph)))-1.0,
max(float(sdPLUS(m,0:(count[m]-1), grph)))+1.0] , thick =2, /nodata
      oplot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color = 227,
thick = 3, psym =-3
```

```
       oplot, sdPLUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1, psym =-3
       oplot, sdMINUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1, psym =-3
      endfor
      snapshot = TVRD(True=1)
      !p.multi = 0
Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(DepInterval,2)+'cms_MA_HorPlo
t.jpg', snapshot, True=1, Quality=100
Quality=100
endfor
Set_Plot, thisDevice
PRINT, SYSTIME(1) - T, 'Seconds'
thisDevice = !D.Name
Set_Plot, 'Z'
Erase
Device, Set_Resolution=[400,65500], Set_Pixel_Depth=24, Decomposed=0
loadct, 39
for grph = 0, 2 do begin
      !P.MULTI = [0,1,count_pos]
      for m =0, num-1, 1 do begin  ; change to num-1, 0
       print, count[m]
       ;if count[m] eq 0 then continue
       plot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),$ ;yrange =
[end_depth[m,(count[m]-1)],st_depth[m,0]]
       yrange = [max(float(cent_depth(m,0:(count[m]-
1))))+0.03,min(float(cent_depth(m,0:(count[m]-1))))-0.03 ] ,$
       bACKGROUND = 255, xtitle = 'Mean(Red), SD(Yellow)' , ytitle = 'Central Depth
(m)', ystyle = 1,xstyle = 1, color = 0,charsize = 1.75,$
       xrange = [min(float(sdMINUS(m,0:(count[m]-1), grph)))-1.0,
max(float(sdPLUS(m,0:(count[m]-1), grph)))+1.0] , thick =2, /nodata
       oplot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color = 227,
thick = 3, psym =-3
       oplot, sdPLUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1, psym =-3
       oplot, sdMINUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
204, linestyle = 1, psym =-3
      endfor
      snapshot = TVRD(True=1)
      !p.multi = 0
Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(DepInterval,2)+'cms_MA_VertPl
ot.jpg', snapshot, True=1, Quality=100
;Write_jpeg, 'C:\Users\Shaina.889\Desktop\Test\testprofilerun.jpg', snapshot, True=1,
Quality=100
endfor
Set_Plot, thisDevice
PRINT, SYSTIME(1) - T, 'Seconds'
end else
EndIF
end
;***********Code for Mosaic File ends here************************
```

Appendix A-3

```
;****PROGRAM TO CREATE INTERMEDIATE DATA & BINARY FILES***

pro mosaicfiles_creatbasedata
mfname = Dialog_pickfile(Title = 'Please select a mosaic
file',filter='*.dat',/must_exist)
mdir = file_dirname(mfname)
mfile = file_basename(mfname, '.dat')
print, mfile
cd, mdir
core = strmatch(mfname, '*_core_*', /fold_case)
; core = 0 means it is box mosaic
T = SYSTIME(1)
envi_open_file, mfname, r_fid=mfid, /No_Realize
             if (mfid eq -1) then return
envi_file_query,mfid,nb=snb, ns=sns, nl=snl, file_type=sfile_type,
data_type=sdata_type,$ wl=swl, xstart= sxstart, ystart= systart, dims=dims, bnames =
bnames
mdat = envi_get_slice(FID=mfid, LINE=10, POS=0, XE=sns-1 , XS= sxstart)
ind_start= make_array(500,/long)
ind_end = make_array(500, /long)
ind_end(0)= sns
value = 0
for i = 0, sns-1 do begin
     if mdat(sns-1-i) ne mdat(sns-1-i) then begin
       if count eq 1 then begin
              value = value+1
              count = count+1
       endif
       ind_end(value) = sns-i-1
     endif else begin
       ind_start(value)= sns-i
       count = 1
     endelse
endfor
print, value
sx_start = ind_start(0:value-1) ; array with starting sample number of each individual
core$ starting from rightmost core
sx_end = ind_end(0:value-1) ; array with ending sample number of each individual core
starting$ from rightmost core

pfile = dialog_pickfile(title = 'Please select the related profile sheet')
und1 = StrPos(pfile, '_')
und2 = StrPos(pfile, '_', /REVERSE_SEARCH)
num = Strmid(pfile, und1+1, und2-und1-1)
data = strarr(num+1)
if num ne value then begin
  result = dialog_message("Wrong Profile Sheet", /error)
  return
endif
totalelements=long(0)
openw, lun, strtrim(mfile, 2)+'_BinaryDatafile.dat', /get_lun
```

```
no_of_lines = make_array(value, /integer)
;adding average of each line to the binary file band by band
for b =0, snb-1 do begin
     for m = 0, value-1 do begin
       no_of_lines[m] = 0
       ns = sx_end(m)-sx_start(m)+1
       temp = make_array(ns,/Double)
       for l = 0, snl-1 do begin
              temp = envi_get_slice(fid=mfid, line =l, pos = b, xe =sx_end[m]-1, xs $
                     =sx_start[m]-1)
              if mean(temp) ne mean(temp) then continue
              avg = double(Mean(temp))
              totalelements = totalelements+1
              no_of_lines[m] = no_of_lines[m]+1
              writeu, lun, avg
       endfor
     endfor
endfor

;adding standard deviation of each line to the binary file band by band
for b =0, snb-1 do begin
     for m = 0, value-1 do begin
       ns = sx_end(m)-sx_start(m)+1
       temp = make_array(ns,/Double)
       for l = 0, snl-1 do begin
              temp = envi_get_slice(fid=mfid, line =l, pos = b, xe =sx_end[m]-1, xs$
                     =sx_start[m]-1)
              if mean(temp) ne mean(temp) then continue
              sd = double(Stddev(temp))
              totalelements = totalelements+1
              writeu, lun, sd
       endfor
     endfor
endfor
print, totalelements
writeu,lun, totalelements
close, lun
PRINT, SYSTIME(1) - T, 'Seconds'

openr, dlun,pfile, /get_lun
     readf, dlun, data
free_lun, dlun
pdata= strarr(num+1,8)
s= strarr(num+1,8)
for i = 0, num do begin
     s[i,*] = STRSPLIT(data[i], ',', /EXTRACT)
endfor
comb = fix(s[1:num,0])
order = sort(comb)
start_ht = findgen(num)
end_ht = findgen(num)
height = findgen(num)
ndim = findgen(num)
ndim_int = lindgen(num)
ndim_dec = findgen(num)
```

```
n = indgen(num)
Line_to_mm = findgen(num)
for i = 0, num-1 do begin
     height[i] = s[order(i)+1, 2]
     start_ht[i] = s[order(i)+1, 3]
     end_ht[i] = s[order(i)+1, 4]
endfor
openw, datlun, strtrim(mfile, 2)+'_BaseDatafile.csv', /get_lun
printf, datlun, 'S.No./Core No.,', 'Start Depth(m) ,', 'End Depth(m) ,', 'Total
Depth(m) ,', 'No. of$ Lines,', 'Depth per line(mm),', format = '('+strtrim(6,2)+'(A))

for i =0, num-1 do begin
     if i ne 0 then begin
       if start_ht[i] ne end_ht[i-1] then begin
              height[i:i+7] = (float(end_ht[i+7])-float(end_ht[i-1]))/8.0
                     for j = i, i+7 do begin
                            start_ht[j] = end_ht[j-1]
                            end_ht[j] = start_ht[j]+height[j]
                     endfor
       endif
     endif
     Line_to_mm[i] = (height[i]*1000)/no_of_lines[i] ;depth per line in mm
     printf, datlun, i+1,start_ht[i],end_ht[i], height[i], no_of_lines[i],
Line_to_mm[i], $
     format = '(I0.0,",",'+strtrim(3,2)+'(F0.4, ","),'+strtrim(1,2)+'(I0.0,
","),'+strtrim(1,2)+'$
     (F0.4, ","))'
endfor
free_lun, datlun
PRINT, SYSTIME(1) - T, 'Seconds'

end
```

Appendix A-4

```
;****PROGRAM TO CREATE BITUMEN PROFILES - METHOD 2***

pro profilesheet
pro PSD_TBC_Vertical_Profiling_mosaic_event, Event
 wTarget = (widget_info(Event.id,/NAME) eq 'TREE' ?  $
 widget_info(Event.id, /tree_root) : event.id)
  wWidget =  Event.top
  case wTarget of
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_Input_Browse'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_TEXT_Resolution'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_MAv_check'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_Start'): begin
    end
    Widget_Info(wWidget, FIND_BY_UNAME='WID_BUTTON_0'): begin
    end
    else:
  endcase
end

pro PSD_TBC_Vertical_Profiling_mosaic, GROUP_LEADER=wGroup, $
_EXTRA=_VWBExtra_

WID_BASE_0 = Widget_Base( GROUP_LEADER=wGroup, UNAME='WID_BASE_0'  $
      ,XOFFSET=5 ,YOFFSET=5 ,SCR_XSIZE=400 ,SCR_YSIZE=300 ,TITLE='Creating Profiles of
Mosaic Images' $
       ,SPACE=3 ,XPAD=3 ,YPAD=3)

WID_LABEL_0 = Widget_Label(WID_BASE_0, UNAME='WID_LABEL_0'  $
      ,XOFFSET=19 ,YOFFSET=20 ,SCR_XSIZE=56 ,SCR_YSIZE=18  $
      ,/ALIGN_LEFT ,VALUE='Input File:')

WID_TEXT_Input_File = Widget_Text(WID_BASE_0,  $
      UNAME='WID_TEXT_Input_File' ,FRAME=1 ,XOFFSET=78 ,YOFFSET=17  $
      ,SCR_XSIZE=240 ,SCR_YSIZE=24 ,/EDITABLE ,XSIZE=20 ,YSIZE=1)

WID_BUTTON_Input_Browse = Widget_Button(WID_BASE_0,  $
      UNAME='WID_BUTTON_Input_Browse' ,XOFFSET=320 ,YOFFSET=16  $
      ,SCR_XSIZE=57 ,SCR_YSIZE=25 ,EVENT_FUNC='Browse_Input_file'  $
      ,/ALIGN_CENTER ,VALUE='Browse')

WID_LABEL_3 = Widget_Label(WID_BASE_0, UNAME='WID_LABEL_3'  $
      ,XOFFSET=100 ,YOFFSET=84 ,SCR_XSIZE=130 ,SCR_YSIZE=18  $
      ,/ALIGN_LEFT ,VALUE='Length of Interval (cm):')

WID_TEXT_Resolution = Widget_Text(WID_BASE_0,  $
      UNAME='WID_TEXT_Resolution' ,XOFFSET=230 ,YOFFSET=80  $
      ,SCR_XSIZE=44 ,SCR_YSIZE=21 ,EVENT_FUNC='Get_Resolution'  $
      ,/EDITABLE ,/ALL_EVENTS ,XSIZE=20 ,YSIZE=1)
```

```
WID_BASE_1 = Widget_Base(WID_BASE_0, UNAME='WID_BASE_1' ,XOFFSET=70  $
      ,YOFFSET=150 ,TITLE='IDL' ,COLUMN=1 ,/NONEXCLUSIVE)

WID_BUTTON_MAv_check = Widget_Button(WID_BASE_1,  $
      UNAME='WID_BUTTON_MAv_check' ,SCR_XSIZE=250 ,SCR_YSIZE=26  $
      ,EVENT_FUNC='Check_MAv_status' ,/ALIGN_LEFT ,VALUE='Calculate'+ $
      ' moving average and standard deviation?')

WID_BUTTON_Start = Widget_Button(WID_BASE_0,  $
      UNAME='WID_BUTTON_Start' ,FRAME=1 ,XOFFSET=105 ,YOFFSET=200  $
      ,SCR_XSIZE=70 ,SCR_YSIZE=36 ,EVENT_FUNC='Start_process'  $
      ,/ALIGN_CENTER ,VALUE='Start')

WID_BUTTON_0 = Widget_Button(WID_BASE_0, UNAME='WID_BUTTON_0'  $
      ,FRAME=1 ,XOFFSET=220 ,YOFFSET=200 ,SCR_XSIZE=70 ,SCR_YSIZE=36  $
      ,EVENT_FUNC='Cancel_Process' ,/ALIGN_CENTER ,VALUE='Cancel')

Widget_Control, /REALIZE, WID_BASE_0
     sState={WID_TEXT_Input_File:WID_TEXT_Input_File,       $
             file_Data:0L,                                  $
             WID_TEXT_Resolution:WID_TEXT_Resolution,       $
             flag_MAv:0}
WIDGET_CONTROL, WID_BASE_0, Set_UVALUE=sState, /NO_COPY
XManager, 'PSD_TBC_Vertical_Profiling_mosaic', WID_BASE_0, /NO_BLOCK
end
function Browse_Input_file, Event
 WIDGET_CONTROL, Event.top, GET_UVALUE=sState, /NO_COPY
      lfname = Dialog_Pickfile(/MUST_EXIST, TITLE='Select a data file',
       File='image.dat',FILTER = '*.*')
        IF lfname NE "" THEN BEGIN
             envi_open_file, lfname, r_fid=lfid,/NO_REALIZE
             if (lfid ne -1) then begin
                              sState.file_Data=lfid
                  widget_control, sState.WID_TEXT_Input_File, Set_Value=lfname
                        endif
        ENDIF
     WIDGET_CONTROL, Event.top, SET_UVALUE=sState, /NO_COPY
     return, Event ; By Default, return the event.
end
function get_Resolution, Event
      return, Event ; By Default, return the event.
end
function Check_MAv_status, Event
 WIDGET_CONTROL, Event.top, GET_UVALUE=sState, /NO_COPY
       CASE sState.flag_MAv OF
          0: sState.flag_MAv=1
          1: sState.flag_MAv=0
       ENDCASE
   WIDGET_CONTROL, Event.top, SET_UVALUE=sState, /NO_COPY
     return, Event ; By Default, return the event.
end
function Cancel_Process, Event
 widget_control,event.top,/destroy
     return, Event ; By Default, return the event.
```

```
end
function Start_process, Event
WIDGET_CONTROL, Event.top, GET_UVALUE=sState, /NO_COPY
      widget_control,  sState.WID_TEXT_Input_File, Get_Value=input_filename
      widget_control,  sState.WID_TEXT_Resolution, Get_Value=Resolution
     fileId= sState.file_Data
Resolution=float(Resolution)
DepInterv = Resolution[0]
profile_tbc_psd_mosaic, input_filename, DepInterv, sState.Flag_MAv
end
pro profile_tbc_psd_mosaic, sfname, DepInterval, flag
;flag = 1 means moving average
;flag = 1
IF sfname NE "" THEN BEGIN
mosaic = STRMATCH(sfname, '*mosaic*', /FOLD_CASE)

;***********Code for Mosaic File starts here*************************
IF mosaic eq 1 then begin
;mfname = Dialog_pickfile(Title = 'Please select a mosaic
file',filter='*.dat',/must_exist)
mfname = sfname
mdir = file_dirname(mfname, /mark_directory)
cd, mdir
bfname = Dialog_pickfile(Title = 'Please select the related binary
file',filter='*.dat',/must_exist)
dfname = Dialog_pickfile(Title = 'Please select the related base data
file',filter='*.csv',/must_exist)
mfile = file_basename(mfname, '.dat')
file_mkdir, mfile
T = SYSTIME(1)
envi_open_file, mfname, r_fid=mfid, /No_Realize
             if (mfid eq -1) then return
envi_file_query,mfid,nb=snb, ns=sns, nl=snl, file_type=sfile_type,
data_type=sdata_type,$
wl=swl, xstart= sxstart, ystart= systart, dims=dims, bnames = bnames
num_lines = file_lines(dfname)
data = strarr(num_lines)

;READING BASE DATA FILE
openr, dlun,dfname, /get_lun
     readf, dlun, data
free_lun, dlun
base_data= strarr(num_lines,6)
for i = 0, num_lines-1 do begin
     base_data[i,*] = STRSPLIT(data[i], ',', /EXTRACT)
endfor
num_cores = num_lines-1
start_ht = float(base_data[1:num_cores,1])
end_ht = float(base_data[1:num_cores,2])
no_of_lines = fix(base_data[1:num_cores,4])
height = float(base_data[1:num_cores,3])
linetomm = float(base_data[1:num_cores,5])
linetomm = (height*1000)/no_of_lines
;READING BINARY FILE
openr, blun, bfname, /get_lun
```

```
while ~ EOF(blun) do begin
     totalelements = read_binary(blun, data_type=3, data_dims = 1)
endwhile
POINT_LUN, blun, 0
print, totalelements
Avg0_offset = 0
Avg1_offset = (totalelements/6)
Avg2_offset = 2*(totalelements/6)
SD0_offset = 3*(totalelements/6)
SD1_offset = 4*(totalelements/6)
SD2_offset = 5*(totalelements/6)
avg = make_array(num_cores,avg1_offset,3, /double)
sd = make_array(num_cores,avg1_offset,3, /double)
sdplus = make_array(num_cores,avg1_offset,3, /double)
sdminus = make_array(num_cores,avg1_offset,3, /double)
;Resolution = 100 ; in cms***********
Resolution = DepInterval
mov_avg =1 ; interval or moving average*********
count = make_array(num_cores, /Integer)

;calculating parameters needed for "MOVING AVERAGE"
;m = core
;l = line
if flag eq 1 then begin
for m = 0, num_cores-1 do begin
     lines_needed = round((resolution*10)/linetomm[m])
     if lines_needed eq 0 then lines_needed = 1
     for l = 0, no_of_lines[m]-1 do begin
      if lines_needed gt (no_of_lines[m]-l) then begin
            if m ne num_cores-1 then begin
                  p=0
                  brk = 0
                  while brk eq 0 do begin
                        if p eq 0 then begin
      gap = round((linetomm[m]/linetomm[m+1])*(lines_needed - (no_of_lines[m]-l)))
                        endif else begin
                              if m+p+1 ge num_cores-1 then begin
                                    gap = no_of_lines[num_cores-1]
                                    break
                              endif
                  gap = round(((Resolution*10) - (total(height[m+1:m+p])*1000)-$
                        ((no_of_lines[m]-l)*linetomm[m]))/linetomm[m+p+1])
                        endelse
                        p = p+1
                        if gap gt no_of_lines[m+p] then begin
                              brk = 0
                        endif else brk =1
                  endwhile
                  if p eq 1 then begin no_count = fix(gap + no_of_lines[m]-l)
                  endif
            else no_count = fix(gap + total(no_of_lines[m+1:m+p-1])+no_of_lines[m]-l)
            endif else no_count = fix(no_of_lines[m]-l)
      endif else no_count = fix(lines_needed)
      if m eq 0 then begin
            if no_count gt 1 then begin
```

```
                avg[m,count[m],0] = mean(read_binary(blun, data_type=4, data_dims
                =no_count, data_start = (Avg0_offset+l)*4))
                avg[m,count[m],1] = mean(read_binary(blun, data_type=4, data_dims
                =no_count, data_start = (Avg1_offset+l)*4))
                avg[m,count[m],2] = mean(read_binary(blun, data_type=4, data_dims
                =no_count, data_start = (Avg2_offset+l)*4))
                sd[m,count[m],0] = mean(read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (SD0_offset+l)*4))
                sd[m,count[m],1] = mean(read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (SD1_offset+l)*4))
                sd[m,count[m],2] = mean(read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (SD2_offset+l)*4))
        endif else begin
                avg[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (Avg0_offset+l)*4)
                avg[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (Avg1_offset+l)*4)
                avg[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (Avg2_offset+l)*4)
                sd[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (SD0_offset+l)*4)
                sd[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (SD1_offset+l)*4)
                sd[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
                      no_count, data_start = (SD2_offset+l)*4)
        endelse
  endif else begin
        if no_count gt 1 then begin
                avg[m,count[m],0] = mean(read_binary(blun, data_type=4, data_dims
  =no_count, data_start = (Avg0_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                avg[m,count[m],1] = mean(read_binary(blun, data_type=4, data_dims
  =no_count, data_start = (Avg1_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                avg[m,count[m],2] = mean(read_binary(blun, data_type=4, data_dims
  =no_count, data_start = (Avg2_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                sd[m,count[m],0] = mean(read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (SD0_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                sd[m,count[m],1] = mean(read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (SD1_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                sd[m,count[m],2] = mean(read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (SD2_offset+l+total(no_of_lines[0:m-1], /integer))*4))
        endif else begin
            avg[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
 no_count,data_start = (Avg0_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                avg[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (Avg1_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                avg[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (Avg2_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                sd[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (SD0_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                sd[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (SD1_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                sd[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
  no_count, data_start = (SD2_offset+l+total(no_of_lines[0:m-1], /integer))*4)
        endelse
  endelse
```

```
       sdplus[m,count[m],*] = avg[m,count[m],*]+sd[m,count[m],*]
       sdminus[m,count[m],*] = avg[m,count[m],*]-sd[m,count[m],*]
       count[m] = count[m]+1
      endfor
endfor
endif else begin ;calculating parameters needed for "INTERVAL"
gap =0
p=0
for m = 0, num_cores-1 do begin
      if p ge 2 then begin
       p = p-1
       continue
      endif
      if gap eq no_of_lines[m] then begin
       gap =0
       continue
      endif
      lines_needed = round((resolution*10)/linetomm[m])
      if lines_needed eq 0 then lines_needed = 1
      for l = gap, no_of_lines[m]-1, lines_needed do begin
       gap =0
       p=0
       if lines_needed gt (no_of_lines[m]-l) then begin
              if m ne num_cores-1 then begin

                     brk = 0
                     while brk eq 0 do begin
                            if p eq 0 then begin
       gap = round((linetomm[m]/linetomm[m+1])*(lines_needed - (no_of_lines[m]-l)))
                            endif else begin
                                   if m+p+1 ge num_cores-1 then begin
                                          gap = no_of_lines[num_cores-1]
                                          break
                                   endif
                     gap = round(((Resolution*10) - (total(height[m+1:m+p])*1000)-$
                            ((no_of_lines[m]-l)*linetomm[m]))/linetomm[m+p+1])
                            endelse
                            p = p+1
                            if gap gt no_of_lines[m+p] then begin
                                   brk = 0
                            endif else brk =1
                     endwhile
                     if p eq 1 then begin no_count = fix(gap + no_of_lines[m]-l)
                     endif
              else no_count = fix(gap + total(no_of_lines[m+1:m+p-1])+no_of_lines[m]-l)
              endif else no_count = fix(no_of_lines[m]-l)
       endif else no_count  = fix(lines_needed)
       if m eq 0 then begin
              if no_count gt 1 then begin
                     avg[m,count[m],0] = mean(read_binary(blun, data_type=4, data_dims
                     =no_count, data_start = (Avg0_offset+l)*4))
                     avg[m,count[m],1] = mean(read_binary(blun, data_type=4, data_dims
                     =no_count, data_start = (Avg1_offset+l)*4))
                     avg[m,count[m],2] = mean(read_binary(blun, data_type=4, data_dims
                     =no_count, data_start = (Avg2_offset+l)*4))
```

```
                    sd[m,count[m],0] = stddev(read_binary(blun, data_type=4, data_dims
                    =no_count, data_start = (SD0_offset+l)*4))
                    sd[m,count[m],1] = stddev(read_binary(blun, data_type=4, data_dims
                    =no_count, data_start = (SD1_offset+l)*4))
                    sd[m,count[m],2] = stddev(read_binary(blun, data_type=4, data_dims
                    =no_count, data_start = (SD2_offset+l)*4))
              endif else begin
                    avg[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
                          no_count, data_start = (Avg0_offset+l)*4)
                    avg[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
                          no_count, data_start = (Avg1_offset+l)*4)
                    avg[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
                          no_count, data_start = (Avg2_offset+l)*4)
                    sd[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
                          no_count, data_start = (SD0_offset+l)*4)
                    sd[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
                          no_count, data_start = (SD1_offset+l)*4)
                    sd[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
                          no_count, data_start = (SD2_offset+l)*4)
              endelse
       endif else begin
              if no_count gt 1 then begin
                    avg[m,count[m],0] = mean(read_binary(blun, data_type=4, data_dims
     = no_count, data_start = (Avg0_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                    avg[m,count[m],1] = mean(read_binary(blun, data_type=4, data_dims
= no_count, data_start = (Avg1_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                    avg[m,count[m],2] = mean(read_binary(blun, data_type=4, data_dims
= no_count, data_start = (Avg2_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                    sd[m,count[m],0] = stddev(read_binary(blun, data_type=4, data_dims
= no_count, data_start = (SD0_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                    sd[m,count[m],1] = stddev(read_binary(blun, data_type=4, data_dims
= no_count, data_start = (SD1_offset+l+total(no_of_lines[0:m-1], /integer))*4))
                    sd[m,count[m],2] = stddev(read_binary(blun, data_type=4, data_dims
= no_count, data_start = (SD2_offset+l+total(no_of_lines[0:m-1], /integer))*4))
              endif else begin
                  avg[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
no_count, data_start = (Avg0_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                    avg[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
       no_count, data_start = (Avg1_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                    avg[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
       no_count, data_start = (Avg2_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                    sd[m,count[m],0] = read_binary(blun, data_type=4, data_dims =
       no_count, data_start = (SD0_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                    sd[m,count[m],1] = read_binary(blun, data_type=4, data_dims =
       no_count, data_start = (SD1_offset+l+total(no_of_lines[0:m-1], /integer))*4)
                    sd[m,count[m],2] = read_binary(blun, data_type=4, data_dims =
       no_count, data_start = (SD2_offset+l+total(no_of_lines[0:m-1], /integer))*4)
              endelse
       endelse
       sdplus[m,count[m],*] = avg[m,count[m],*]+sd[m,count[m],*]
       sdminus[m,count[m],*] = avg[m,count[m],*]-sd[m,count[m],*]
       count[m] = count[m]+1
     endfor
endfor
endelse
```

```
close, blun
set_total = Total(count)
print, count
print, set_total
mpos = where(count gt 0, count_pos)
PRINT, SYSTIME(1) - T, 'Seconds'

;***PRINTING AVERAGE AND STANDARD DEVIATION VALUES ON A CSV FILE****
;avg = strtrim(avg,2)
;sd = strtrim(sd,2)
;sdplus = strtrim(sdplus,2)
;sdminus = strtrim(sdminus,2)
cent_depth = make_array(num_cores, max(count), /float)
st_depth = make_array(num_cores, max(count), /float)
end_depth = make_array(num_cores, max(count), /float)
st_depth[0,0] = start_ht[0,0]
cd, mdir
file_mkdir, mfile
cd, mfile
Resol_str=string(Resolution,format='(F0.2)')
p=long(0)
check =0
brk =0
if flag eq 1 then begin
openw, datlun, strtrim(mfile,
2)+'_'+strtrim(Resol_str,2)+'cms__MovingAverage_Data.csv', /get_lun
printf, datlun, 'S.No.,', 'Start Depth(m),', 'End Depth(m),', 'Central Depth(m),', $
'Average 1,', 'Average 2,', 'Average 3,', 'SD1,', 'SD2,', 'SD3,', format =
'('+strtrim(10,2)+'(A))
for m = 0,num_cores-1 do begin
     for j = 0, count[m]-1 do begin
       end_depth[m,j] = st_depth[m,j]+(float(resolution)/100)
       if end_depth[m,j] gt end_ht[num_cores-1] then begin
              end_depth[m,j] = end_ht[num_cores-1]
       endif
       cent_depth[m,j] = (st_depth[m,j]+end_depth[m,j])/2
       printf, datlun, p+1, st_depth[m,j], end_depth[m,j],cent_depth[m,j],$
       strtrim(avg[m,j,*],2),strtrim(sd[m,j,*],2), format =$
       '(I32.0,",",'+strtrim(9,2)+'(F0.4,","))'
       p=p+1
       if p lt set_total then begin
              if j lt count[m]-1 then begin
                     st_depth[m,j+1] = st_depth[m,j]+(linetomm[m]/1000)
                     check = st_depth[m,j+1]
              endif else begin
                     st_depth[m+1,0] = st_depth[m,j]+(linetomm[m]/1000)
                     check = st_depth[m+1,0]
              endelse
       endif
       if check ge end_ht[num_cores-1] then begin
              brk =1
              break
       endif
     endfor
     if brk eq 1 then break
```

```
endfor
endif else begin
openw, datlun, strtrim(mfile, 2)+'_'+strtrim(Resol_str,2)+'cms_Data.csv', /get_lun
printf, datlun, 'S.No.,', 'Start Depth(m),', 'End Depth(m),', 'Central Depth(m),', $
'Average_'+strtrim(bnames[0],2)+',', 'Average_'+strtrim(bnames[1],2)+',',$
'Average_'+strtrim(bnames[2],2)+',', 'SD_'+strtrim(bnames[0],2)+',',$
'SD_'+strtrim(bnames[1],2)+',', 'SD_'+strtrim(bnames[2],2)+',', format =
'('+strtrim(10,2)+'(A))
check = st_depth[0,0]
for m = 0,num_cores-1 do begin
     if count[m] eq 0 then continue
     for j = 0, count[m]-1 do begin
      end_depth[m,j] = check+(float(resolution)/100)
      if m lt num_cores-1 then begin
            if (total(count[m+1:num_cores-1]) eq 0 && j eq (count[m]-1)) then
                  end_depth[m,j] = end_ht[num_cores-1]
      endif
      if (m eq (num_cores-1) && j eq (count[m]-1))||(end_depth[m,j] gt
end_ht[num_cores-1])                    then begin
            end_depth[m,j] = end_ht[num_cores-1]
      endif
      cent_depth[m,j] = (st_depth[m,j]+end_depth[m,j])/2
      printf, datlun, p+1,check, end_depth[m,j],
cent_depth[m,j],strtrim(avg[m,j,*],2),$
     strtrim(sd[m,j,*],2),  format = '(I32.0,",",'+strtrim(9,2)+'(F0.4,","))'
      p=p+1
      if p lt set_total then begin
            if j lt count[m]-1 then begin
                  st_depth[m,j+1] = end_depth[m,j]
                  check = st_depth[m,j+1]
            endif else begin
                  st_depth[m+1,0] = end_depth[m,j]
                  check = st_depth[m+1,0]
            endelse
      endif
      if check ge end_ht[num_cores-1] then begin
            brk =1
            break
      endif
     endfor
     if brk eq 1 then break
endfor
endelse
free_lun, datlun
PRINT, SYSTIME(1) - T, 'Seconds'
thisDevice = !D.Name
Set_Plot, 'Z'
Erase
Device, Set_Resolution=[54000,900], Decomposed=0, Set_Pixel_Depth=24
loadct, 39
print, count_pos
if flag eq 1 then psymb = -3 else psymb = -1
for grph = 0, 2 do begin
     !P.MULTI = [0,count_pos,1]
     for m =num_cores-1, 0, -1 do begin  ; change to num-1, 0
```

```
       if count[m] eq 0 then continue
       plot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),$ ;yrange =
[end_depth[m,(count[m]-1)],st_depth[m,0]]
       yrange = [max(float(cent_depth(m,0:(count[m]-
1))))+0.03,min(float(cent_depth(m,0:(count[m]-1))))-0.03 ] ,$
       bACKGROUND = 255, xtitle = 'Mean(Blue), SD(Purple)' , ytitle = 'Central Depth
(m)', ystyle = 1,xstyle = 1, color = 0,charsize = 1.75,$
       xrange = [min(float(sdMINUS(m,0:(count[m]-1), grph)))-1.0,
max(float(sdPLUS(m,0:(count[m]-1), grph)))+1.0] , thick =2, /nodata
       oplot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), thick = 3,
color = 64, psym=psymb
       oplot, sdPLUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),  linestyle
= 1, color = 30, psym=psymb
       oplot, sdMINUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),
linestyle = 1, color = 30, psym=psymb
     endfor
     !p.multi = 0
       snapshot = TVRD(True=1)
     if flag eq 1 then begin
       Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(Resol_str,2)+'cms_MA_HorPlot.
jpg', snapshot, True=1, Quality=100
     endif else Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(Resol_str,2)+'cms_HorPlot.jpg
', snapshot, True=1, Quality=100
endfor
Set_Plot, thisDevice
PRINT, SYSTIME(1) - T, 'Seconds'
thisDevice = !D.Name
Set_Plot, 'Z'
Erase
Device, Set_Resolution=[400,65500], Set_Pixel_Depth=24, Decomposed=0
loadct, 39
print, count_pos
for grph = 0, 2 do begin
     !P.MULTI = [0,1,count_pos]
     for m =num_cores-1, 0, -1 do begin  ; change to num-1, 0
       if count[m] eq 0 then continue
       plot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)),$ ;yrange =
[end_depth[m,(count[m]-1)],st_depth[m,0]]
       yrange = [max(float(cent_depth(m,0:(count[m]-
1))))+0.03,min(float(cent_depth(m,0:(count[m]-1))))-0.03 ] ,$
bACKGROUND = 255, xtitle = 'Mean(Blue), SD(Purple)' , ytitle = 'Central Depth (m)',
ystyle = 1,xstyle = 1, color = 0,charsize = 1.75,xrange =
[min(float(sdMINUS(m,0:(count[m]-1), grph)))-1.0, max(float(sdPLUS(m,0:(count[m]-1),
grph)))+1.0] , thick =2, /nodata
       oplot, avg(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color = 64,
thick = 3, psym=psymb
       oplot, sdPLUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
30, linestyle = 1, psym=psymb
       oplot, sdMINUS(m,0:(count[m]-1), grph), cent_depth(m,0:(count[m]-1)), color =
30, linestyle = 1, psym=psymb
     endfor
     snapshot = TVRD(True=1)
     !p.multi = 0
```

```
     if flag eq 1 then begin
       Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(Resol_str,2)+'cms_MA_VertPlot
.jpg', snapshot, True=1, Quality=100
     endif else Write_jpeg,
strtrim(mfile,2)+'_'+strtrim(bnames[grph],2)+'_'+strtrim(Resol_str,2)+'cms_VertPlot.jp
g', snapshot, True=1, Quality=100
endfor
Set_Plot, thisDevice
PRINT, SYSTIME(1) - T, 'Seconds'

EndIF

;***********Code for Mosaic File ends here************************
```

Appendix A-5

ELSEVIER LICENSE
TERMS AND CONDITIONS

Jul 27, 2015

This is a License Agreement between Shaina Mahajan ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

Supplier	Elsevier Limited The Boulevard,Langford Lane Kidlington,Oxford,OX5 1GB,UK
Registered Company Number	1982084
Customer name	Shaina Mahajan
Customer address	113-5125, Riverbend Road NW Edmonton, AB T6H5K5
License number	3677310410959
License date	Jul 27, 2015
Licensed content publisher	Elsevier
Licensed content publication	Journal of Petroleum Science and Engineering
Licensed content title	A laboratory study of single-well steam-assisted gravity drainage process
Licensed content author	Serhat Akin,Suat Bagci
Licensed content date	15 December 2001
Licensed content volume number	32
Licensed content issue number	1
Number of pages	11
Start Page	23
End Page	33
Type of Use	reuse in a thesis/dissertation
Intended publisher of new work	other
Portion	figures/tables/illustrations

Number of figures/tables/illustrations	1
Format	both print and electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No
Original figure numbers	Fig. 1.
Title of your thesis/dissertation	Automating information flow in hyperspectral measurements for soil characterization in tailings
Expected completion date	Aug 2015
Estimated size (number of pages)	150
Elsevier VAT number	GB 494 6272 12
Permissions price	0.00 CAD
VAT/Local Sales Tax	0.00 CAD / 0.00 GBP
Total	0.00 CAD
Terms and Conditions	

permissions@elsevier.com)

6. If the permission fee for the requested use of our material is waived in this instance, please be advised that your future requests for Elsevier materials may attract a fee.

7. Reservation of Rights: Publisher reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

8. License Contingent Upon Payment: While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.

9. Warranties: Publisher makes no representations or warranties with respect to the licensed material.

10. Indemnity: You hereby indemnify and agree to hold harmless publisher and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

11. No Transfer of License: This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

12. No Amendment Except in Writing: This license may not be amended except in a writing signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).

13. Objection to Contrary Terms: Publisher hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and publisher (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

14. Revocation: Elsevier or Copyright Clearance Center may deny the permissions described in this License at their sole discretion, for any reason or no reason, with a full refund payable to you. Notice of such denial will be made using the contact information provided by you. Failure to receive such notice will not alter or invalidate the denial. In no event will Elsevier or Copyright Clearance Center be responsible or liable for any costs, expenses or damage incurred by you as a result of a denial of your permission request, other than a refund of the amount(s) paid by you to Elsevier and/or Copyright Clearance Center for denied permissions.

LIMITED LICENSE

The following terms and conditions apply only to specific license types:

15. **Translation**: This permission is granted for non-exclusive world **English** rights only unless your license was granted for translation rights. If you licensed translation rights you may only translate this content into the languages you requested. A professional translator must perform all translations and reproduce the content word for word preserving the integrity of the article. If this license is to re-use 1 or 2 figures then permission is granted for non-exclusive world rights in all languages.

16. **Posting licensed content on any Website**: The following terms and conditions apply as follows: Licensing material from an Elsevier journal: All content posted to the web site must maintain the copyright information line on the bottom of each image; A hyper-text must be included to the Homepage of the journal from which you are licensing athttp://www.sciencedirect.com/science/journal/xxxxx or the Elsevier homepage for books athttp://www.elsevier.com; Central Storage: This license does not include permission for a scanned version of the material to be stored in a central repository such as that provided by Heron/XanEdu.

Licensing material from an Elsevier book: A hyper-text link must be included to the Elsevier homepage at http://www.elsevier.com . All content posted to the web site must maintain the copyright information line on the bottom of each image.

Posting licensed content on Electronic reserve: In addition to the above the following clauses are applicable: The web site must be password-protected and made available only to bona fide students registered on a relevant course. This permission is granted for 1 year only. You may obtain a new license for future website posting.

17. **For journal authors:** the following clauses are applicable in addition to the above:

Preprints:

A preprint is an author's own write-up of research results and analysis, it has not been peer-reviewed, nor has it had any other value added to it by a publisher (such as formatting, copyright, technical enhancement etc.).

Authors can share their preprints anywhere at any time. Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the final versions of articles however authors can update their preprints on arXiv or RePEc with their Accepted Author Manuscript (see below).

If accepted for publication, we encourage authors to link from the preprint to their formal publication via its DOI. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help users to find, access, cite and use the best available version. Please note that Cell Press, The Lancet and some society-owned have different preprint policies. Information on these policies is available on the journal homepage.

Accepted Author Manuscripts: An accepted author manuscript is the manuscript of an article that has been accepted for publication and which typically includes author-incorporated changes suggested during submission, peer review and editor-author communications.

Authors can share their accepted author manuscript:

- – immediately
    - via their non-commercial person homepage or blog
    - by updating a preprint in arXiv or RePEc with the accepted manuscript
    - via their research institute or institutional repository for internal institutional uses or as part of an invitation-only research collaboration work-group

- directly by providing copies to their students or to research collaborators for their personal use
- for private scholarly sharing as part of an invitation-only work group on commercial sites with which Elsevier has an agreement

- – after the embargo period
  - via non-commercial hosting platforms such as their institutional repository
  - via commercial sites with which Elsevier has an agreement

In all cases accepted manuscripts should:

- – link to the formal publication via its DOI
- – bear a CC-BY-NC-ND license - this is easy to do
- – if aggregated with other manuscripts, for example in a repository or other site, be shared in alignment with our hosting policy not be added to or enhanced in any way to appear more like, or to substitute for, the published journal article.

Published journal article (JPA): A published journal article (PJA) is the definitive final record of published research that appears or will appear in the journal and embodies all value-adding publishing activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination and online enrichment.

Policies for sharing publishing journal articles differ for subscription and gold open access articles:

Subscription Articles: If you are an author, please share a link to your article rather than the full-text. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help your users to find, access, cite, and use the best available version.

Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

If you are affiliated with a library that subscribes to ScienceDirect you have additional private sharing rights for others' research accessed under that agreement. This includes use for classroom teaching and internal training at the institution (including use in course packs and courseware programs), and inclusion of the article for grant funding purposes.

Gold Open Access Articles: May be shared according to the author-selected end-user license and should contain a CrossMark logo, the end user license, and a DOI link to the formal publication on ScienceDirect.

Please refer to Elsevier's posting policy for further information.

18. **For book authors** the following clauses are applicable in addition to the above: Authors are permitted to place a brief summary of their work online only. You are not allowed to download and post the published electronic version of your chapter, nor may you scan the printed edition to create an electronic version. **Posting to a repository:** Authors are permitted to post a summary of their chapter only in their institution's repository.

19. **Thesis/Dissertation**: If your license is for use in a thesis/dissertation your thesis may be submitted to your institution in either print or electronic form. Should your thesis be published commercially, please reapply for permission. These requirements include permission for the Library and Archives of Canada to supply single copies, on demand, of the complete thesis and include permission for Proquest/UMI to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission. Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal

- – Charging fees for document delivery or access
- – Article aggregation
- – Systematic distribution via e-mail lists or share buttons

Posting or linking by commercial companies for use by customers of those companies.

20. **Other Conditions**:

v1.7

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Appendix A-6

NATURE PUBLISHING GROUP LICENSE TERMS AND CONDITIONS

Jul 27, 2015

This is a License Agreement between Shaina Mahajan ("You") and Nature Publishing Group ("Nature Publishing Group") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Nature Publishing Group, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3677310010079
License date	Jul 27, 2015
Licensed content publisher	Nature Publishing Group
Licensed content publication	Nature Photonics
Licensed content title	Hyperspectral imaging: Cubes and slices
Licensed content author	David Bannon
Licensed content date	Nov 1, 2009
Volume number	3
Issue number	11
Type of Use	reuse in a dissertation / thesis
Requestor type	academic/educational
Format	print and electronic
Portion	figures/tables/illustrations
Number of figures/tables/illustrations	1
High-res required	no

Figures	Figure 3: Hyperspectral imaging, also known as chemical sensing, allows for the differentiation, display and classification of objects based on their chemical composition.
Author of this NPG article	no
Your reference number	None
Title of your thesis / dissertation	Automating information flow in hyperspectral measurements for soil characterization in tailings
Expected completion date	Aug 2015
Estimated size (number of pages)	150
Total	0.00 CAD

Terms and Conditions

Terms and Conditions for Permissions

Nature Publishing Group hereby grants you a non-exclusive license to reproduce this material for this purpose, and for no other use,subject to the conditions below:

1. NPG warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to Nature Publishing Group and does not carry the copyright of another entity (as credited in the published version). If the credit line on any part of the material you have requested indicates that it was reprinted or adapted by NPG with permission from another source, then you should also seek permission from that source to reuse the material.

2. Permission granted free of charge for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to the work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.Where print permission has been granted for a fee, separate permission must be obtained for any additional, electronic re-use (unless, as in the case of a full paper, this has already been accounted for during your initial request in the calculation of a print run).NB: In all cases, web-based use of full-text articles must be authorized separately through the 'Use on a Web Site' option when requesting permission.

3. Permission granted for a first edition does not apply to second and subsequent editions and for editions in other languages (except for signatories to the STM Permissions Guidelines, or where the first edition permission was granted for free).

4. Nature Publishing Group's permission must be acknowledged next to the figure, table or abstract in print. In electronic form, this acknowledgement must be visible at the same time as the figure/table/abstract, and must be hyperlinked to the journal's homepage.

5. The credit line should read:
   Reprinted by permission from Macmillan Publishers Ltd: [JOURNAL NAME] (reference citation), copyright (year of publication)
   For AOP papers, the credit line should read:

Reprinted by permission from Macmillan Publishers Ltd: [JOURNAL NAME], advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].XXXXX)

Note: For republication from the *British Journal of Cancer,* the following credit lines apply.
Reprinted by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME] (reference citation), copyright (year of publication)For AOP papers, the credit line should read:
Reprinted by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME], advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].XXXXX)

6. Adaptations of single figures do not require NPG approval. However, the adaptation should be credited as follows:

   Adapted by permission from Macmillan Publishers Ltd: [JOURNAL NAME] (reference citation), copyright (year of publication)

   Note: For adaptation from the *British Journal of Cancer,* the following credit line applies.
   Adapted by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME] (reference citation), copyright (year of publication)

7. Translations of 401 words up to a whole article require NPG approval. Please visithttp://www.macmillanmedicalcommunications.com for more information.Translations of up to a 400 words do not require NPG approval. The translation should be credited as follows:

   Translated by permission from Macmillan Publishers Ltd: [JOURNAL NAME] (reference citation), copyright (year of publication).

   Note: For translation from the *British Journal of Cancer,* the following credit line applies.
   Translated by permission from Macmillan Publishers Ltd on behalf of Cancer Research UK: [JOURNAL NAME] (reference citation), copyright (year of publication)

We are certain that all parties will benefit from this agreement and wish you the best in the use of this material. Thank you.

Special Terms:

v1.1

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.