

Spectroradiometer Data Structuring, Pre-Processing and Analysis – an IT Based Approach

A. Hueni* and M. Tuohy

Institute of Natural Resources,
Massey University, Private Bag, Palmerston North 5301, New Zealand
Phone: +64-6-3569099 ext 7371, Fax: +64-6-3505632

Email: a.hueni@massey.ac.nz

Email: m.tuohy@massey.ac.nz

*Corresponding author

ABSTRACT

Hyperspectral data collection results in huge datasets that need pre-processing prior to analysis. A review of the pre-processing techniques identified repetitive procedures with consequently a high potential for automation. Data from different hyperspectral field studies were collected and subsequently used as test sets for the described system. A relational database was utilized to store hyperspectral data in a structured way. Software was written to provide a graphical user interface to the database, pre-processing and analysis functionality. The resulting system provides excellent services in terms of organised data storage, easy data retrieval and efficient pre-processing. It is suggested that the use of such a system can improve the productivity of researchers significantly.

INTRODUCTION

Field spectroradiometry has experienced an ever increasing popularity in the last few years. The technology has advantages over conventional techniques, allowing the non destructive sampling of objects and possibly enabling the user to gain critical information more quickly and cheaply. As a result, many scientists are now actively researching applications of hyperspectral sensing. The operation of the instruments tends to be relatively easy and data are collected quickly. However, the interpretation of these data is not so simple. The main issue when dealing with hyperspectral data is their dimensionality which is the result of sampling a wide spectral range in very narrow bands. This is in itself a problem because the influence of noise on narrow channels is much higher than on traditional broadband channels. Hyperspectral data are more complex than previous multispectral data and different approaches for data handling and information extraction are needed (Vane and Goetz, 1988, Landgrebe, 1997).

Hyperspectral data are essentially multivariate data, consisting of hundreds or even thousands of variables. It has been shown that more bands do not automatically imply better results. Although the separability of classes does increase with growing dimensionality, the classification accuracy does not follow this trend endlessly but will decrease at a certain point. This is called the Hughes Phenomenon and is caused by the ever increasing number of samples needed to build sound statistics if the dimensionality grows (Landgrebe, 1997). In practice this means that more samples must be collected to ensure successful statistical analyses. It is necessary, therefore, to collect a large number of spectral data files, each containing a hyperspectral spectrum. The sheer number of files and variables can become overwhelming. Interestingly, very few studies concerned with hyperspectral data have ever mentioned how the data had been organised and stored.

A further issue that is rarely addressed is the reusability of the data. Reference data is usually compiled in so called spectral libraries. The majority of the publicly available spectral libraries are distributed as physical files. This has drawbacks such as low flexibility and low query performance (Bojinski et al., 2003). Another drawback of most libraries is their restriction in the number of spectra per class. In many cases, only one reference spectrum is supplied. This reduces any statistical analysis to first order statistics. The use of average values may be useful in some circumstances, however, Landgrebe (1997) noted that the reduction of data to mean values results in a loss of information. Second order statistics contain vital information about the distribution of data in spectral or feature space and should therefore be included in spectral data collections.

The time and effort that is spent in collecting spectral data, combined with the characteristically large number of files, makes it clear that spectral data should be well organised. Otherwise valuable data can be lost or loses its value because of missing metadata. Considering the above,

it seems logical to employ a database to store spectral data in a suitable form. Only one example of such a database has been found: SPECCHIO (Bojinski et al., 2003) contains spectral metadata ordered by campaigns, information about sensors, instrument models, landuse type of the sampled area, spatial position and descriptions of the target. A relational database management system (DBMS) is used to hold the above data in several tables. The actual reflectance data is not stored in the DB but held on a dedicated file server and the spectral database links the metadata to the reflectance file via a file path.

A further characteristic of hyperspectral data is the data redundancy. It has been shown that neighbouring wavebands have a high degree of correlation (Thenkabail et al., 2004). This redundancy is created by oversampling, i.e. the spectral signal is sampled at small enough steps to describe very narrow features that could be discriminating (Shaw and Manolakis, 2002). The redundancy and general noisiness of the data usually mean that certain pre-processing must be carried out before any useful analysis can be performed.

We present here a possible solution for the efficient storage and pre-processing of field spectroradiometer data. The system has been successfully used in studies concerned with New Zealand native vegetation, soil properties and pastures.

METHODS

Field Data Structuring

A hierarchical data structure that reflects the real world and the setup of sampling campaigns for vegetation was designed. This structure was derived from the following conditions:

1. Reflectances of several different species were captured
2. In order to describe the in-species variation, several specimens of a species were sampled
3. The variability of the specimens was described by several measurements per specimen

The spatial extent where a specimen was sampled was termed a sample site, thus a species contained a number of sample sites. The sites were numbered in the order of sampling. At each site, several readings were taken to capture the variation exhibited by the specimen in question. A site therefore contained a number of spectra. This led to a hierarchical directory structure (Figure 1). As a general rule at least 10 spectra were collected per site. The calculation of statistics like covariances requires at least 15 spectra per species to obtain meaningful representations in feature space. This implied that a minimum of two sites (replicates) per species were to be captured.

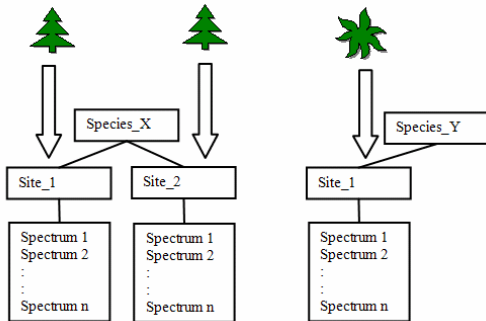


Figure 1: Hierarchical directory structure

Spectral Database Model

The spectral database was designed as a relational database. The presented table structure is in third normal form (3NF). The process of database normalisation reduces complex user views to a set of small, stable table structures (McFadden and Hoffer, 1988). Thus the transition of a model into 3NF removes data redundancy. In practice a certain redundancy is sometimes reintroduced in the form of foreign keys which simplify navigation and data queries in the operative system. Such added relations can be observed between the entities study, species, site and spectrum. For an overview of the spectral database model showing all entities and their relations please refer to Figure 2.

The desired feature list of a spectral database according to the requirements identified in this study was as follows:

- Implements the same hierarchical structure as used for the field data to store species, site and spectrum data
- Multiple studies: can hold spectral data of different field/laboratory campaigns
- Reflectance storage: stores the reflectance data in the database in its original form
- Processing parameters: holds parameters that are needed for the processing of the data
- Statistics: holds 1st and 2nd order statistics to enable classification, discriminant analysis and separability measurements to be carried out efficiently

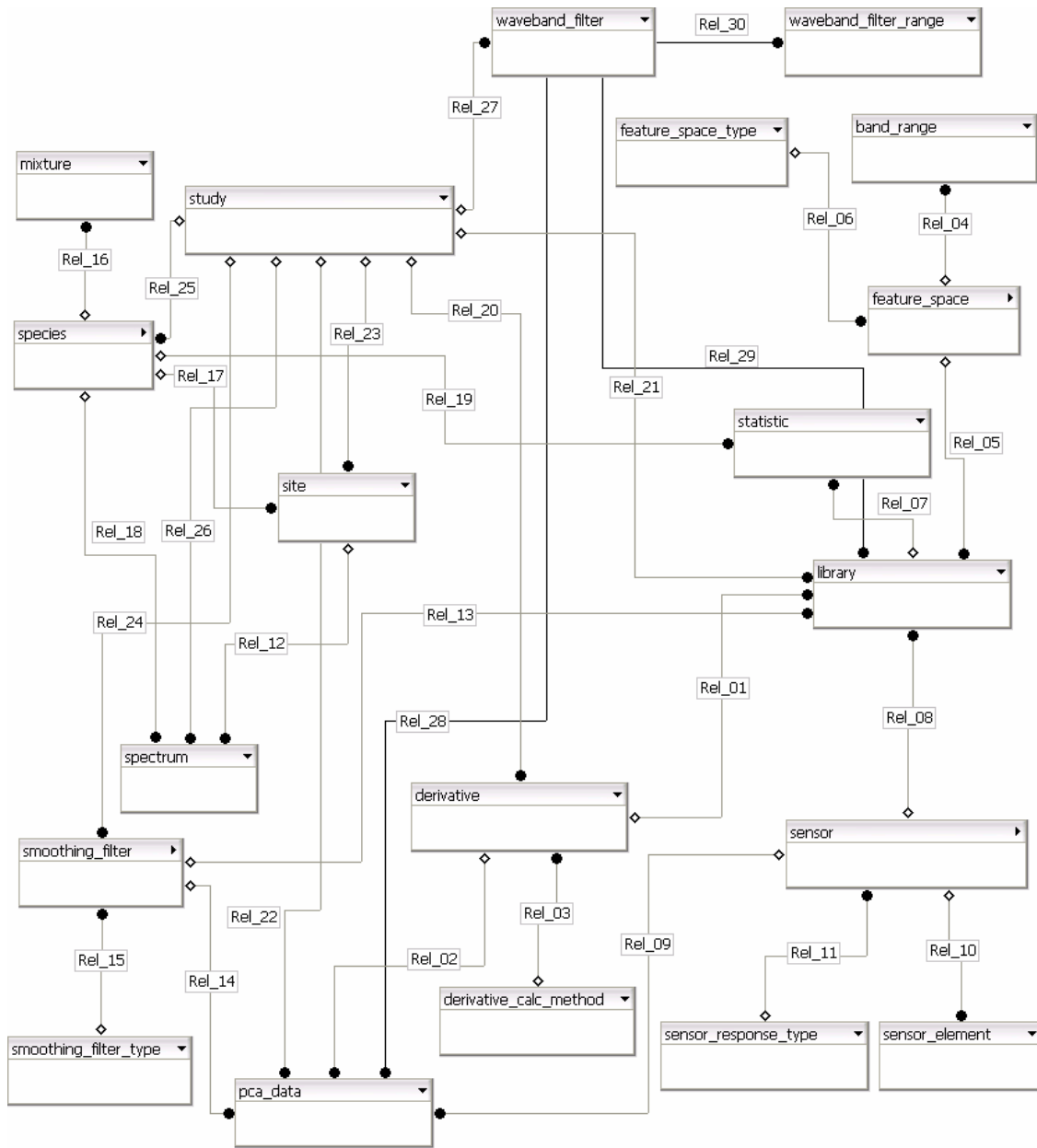


Figure 2: Database model overview at entity level

The entities species, site and spectrum reflect the hierarchical structure that was introduced previously. The study entity was added to the top of this structure to enable the storage of data belonging to different studies in the same database.

The waveband_filter and waveband_filter_range entities hold data that are needed for the removal of noisy or uncalibrated bands from the spectra. These were defined at the study level because every study might have different requirements for the data filtering. E.g. a study that contains data collected by a contact probe will not need to remove water bands as the influence

of the atmosphere is effectively non-existent. Similarly, if a study wishes to concentrate on a certain part of the spectrum only, the unused wavebands can be removed by entering them into the filter structure.

The library can be thought of as a collection of data that can be referred to for the identification of unknown signatures. A library is built for certain settings of the data processing chain, namely waveband filtering, smoothing, sensor convolution, derivative calculation and feature space transformation. The resulting library is setup for classification of data that has been processed in exactly the same way. In other words, before a classification can be carried out on a dataset, its library must be built. A library therefore references the entities `waveband_filter`, `smoothing_filter`, `sensor`, `derivative` and `feature_space`. The actual data needed for a classification is held in the `statistic` entity in the form of a mean vector and a covariance matrix for every species.

The `smoothing_filter` entity holds data needed for the smoothing by a Savitzky-Golay filter (Savitzky and Golay, 1964, Tsai and Philpot, 1998).

The `sensor` entity contains data for the synthesizing of sensor responses. Two general classes of sensors exist, defined by the description of the response type of their elements:

1. Gaussian: each sensor element response is modelled by a Gaussian function. The Gaussian curve is defined by the average wavelength and the full width at half the maximum (FWHM).
2. Ratio: each sensor element response is modelled by ratios applied to narrow band data over a certain range of wavelengths.

The entity `sensor_element` holds both Gaussian and Ratio settings, depending on the type of sensor. In the case of Gaussian sensors, one `sensor_element` entry describes one sensor band. For Ratio sensors, many `sensor_element` entries may be needed to describe one sensor band.

The `derivative` entity holds data for the calculation of derivatives either by an iterative method or by Savitzky-Golay coefficients.

The `feature_space` entity holds or refers to data needed for the feature space transformation. Three types of feature space were considered to be useful, although more possibilities exist:

1. Derivative Greenness Vegetation Indices (DGVI): a feature space is formed by calculating several DGVI (Elvidge and Chen, 1995, Thenkabail et al., 2004). The band ranges for these indices are held in the `band_range` entity.
2. Normalized Two Band Indices (NTBI): a feature space is formed by calculating several NTBIs. The two bands that define each index are held in the `band_range` entity. NTBIs are a generalized version of the well known NDVI (Normalized Difference Vegetation Index) which traditionally uses the values of red and infrared channels (Lillesand et al., 2004).
3. Principal Component Transformation (PCT): PCT is the most widely used algorithm for data reduction and de-correlation (Shaw and Manolakis, 2002). Principal component

analysis performs an eigen-decomposition, the resulting eigenvectors are used to build a transformation matrix, which is then applied to the original data. A feature space is thus formed by calculating a certain number of components. The transformation matrix is held in the `pca_data` entity. The number of components to be calculated is equal to the dimension of the feature space.

Like the library, the `pca_data` is calculated for a certain setup of waveband filtering, smoothing, sensor synthesizing and derivative calculation.

A Spectral Data Management and Processing Software

A spectral database such as that described above is not of much use on its own. Data must be fed into the database and data extraction routines must exist in order to exploit the benefits of the database. The technical requirements for such a system were identified as follows:

- Graphical user interface to the database
- Functions for loading spectral data into the database
- Data pre-processing functions
- Data analysis functions
- File export functions to allow data analysis and plotting in 3rd party packages

The resulting, object oriented software was called SpectraProc.

File System Interfaces

SpectraProc provides input and output interfaces to the file system as illustrated in Figure 3. Input file formats are: ASD binary file as produced by the ASD FieldSpecPro Spectroradiometer, ENVI Z-Profiles that are signatures extracted from hyperspectral imagery in ENVI and sensor specifications in a proprietary, tabulator separated format. ASD files can be imported into the database as part of a study or loaded into memory for classification against a study dataset. ENVI Z-Profiles can be loaded for classification only. Sensor specification files are a way of defining new sensors in the database.

Output can be written in three data formats: (1) CSV (Comma Separated Values) for import into various 3rd party applications like spreadsheets or statistical packages, (2) ENVI Spectral Library for import into ENVI and subsequent use for signature matching and (3) ARFF which is a special format used by WEKA (University of Waikato, 2005).

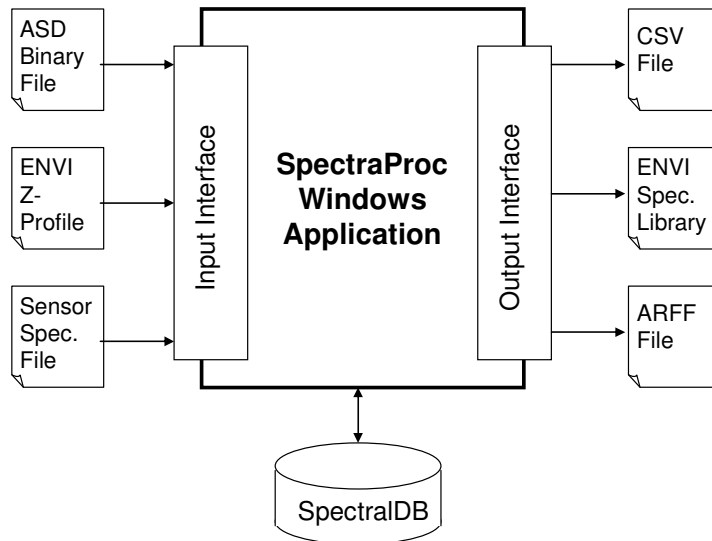


Figure 3: File system interfaces

Spectral Processing Concept

The spectral database stores only the raw spectral data. Further processing of the data is performed at runtime and the results are held in memory. Once a spectrum is loaded from the database it is put through a cascade of operations as shown in Figure 4. The result of every stage is saved in a separate data structure in memory. This allows easy file export of spectral data at any processing step.

The implemented pre-processing steps were:

- Removal of unwanted bands in freely configurable wavelength regions
- Data smoothing using a Savitzky-Golay filter
- Synthesizing of other sensor responses or downsampling
- Derivative calculation
- Feature space transformations: Derivative Indices (e.g. DGVI), Normalized two band indices (e.g. NDVI), Principal Component Transformation (PCT)

The processing parameters for the waveband filtering, synthesizing/downsampling and feature space transformation operations are read from the database. The parameters for smoothing and derivative calculation are taken directly from the settings in the user interface.

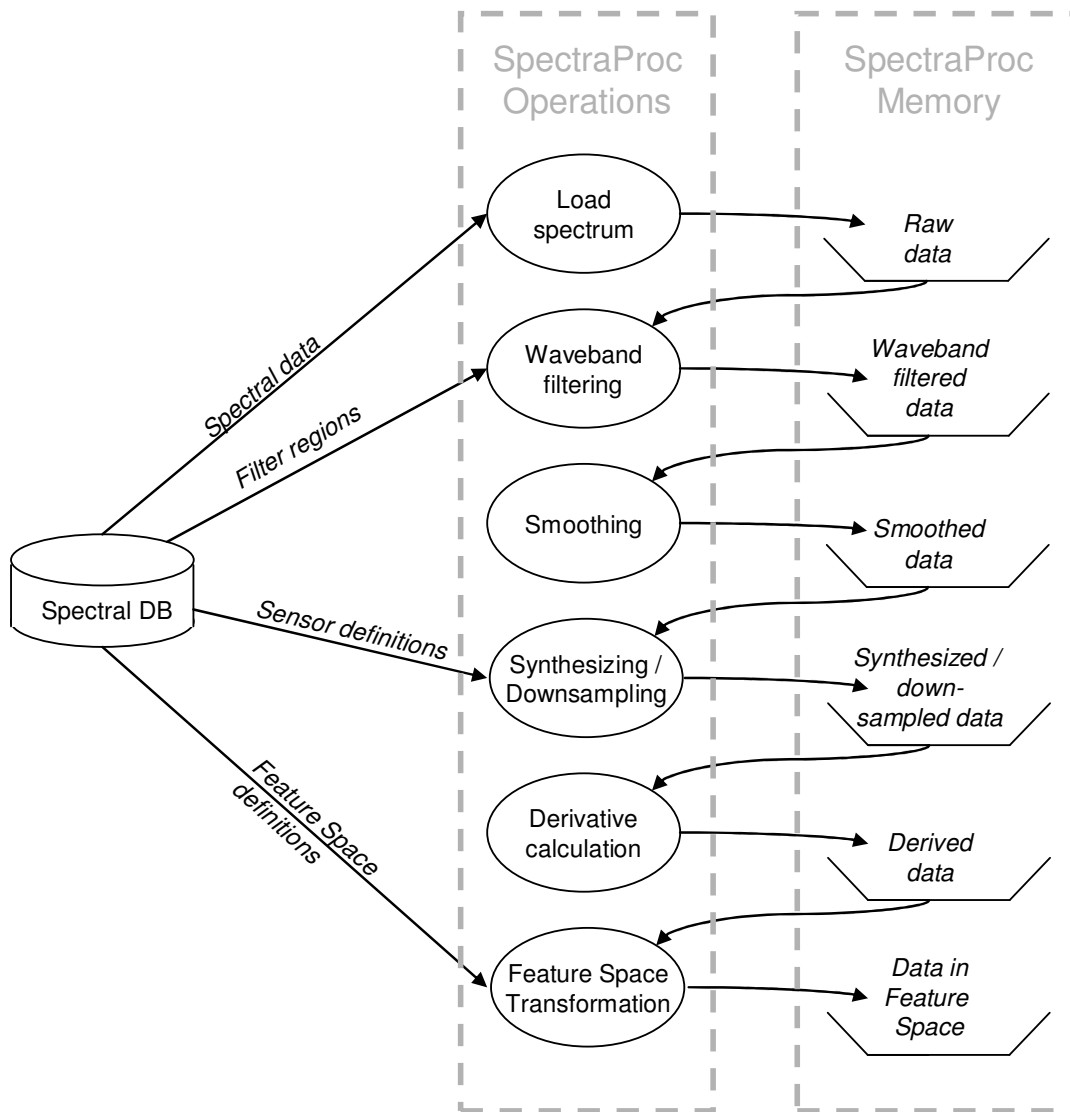


Figure 4: Spectral data processing cascade showing the intermediate storage of spectral data in memory and the processing parameters supplied by the database.

Analysis Functionality

Basic analysis functionality was built into the software: (a) separability analysis in the form of the Jeffries-Matusita (JM) and the Bhattacharya (B) distance (Richards, 1993, Schmidt and Skidmore, 2003), (b) discriminant analysis with the choice of three different discriminant functions (quadratic (gaussian) distance, general squared distance and Spectral Angle Mapper), resulting in the output of a confusion matrix including producer and user accuracy and (c) principal component analysis (PCA) with the output being the eigenvalues, proportions and cumulative proportions.

Implementation

The database was implemented in MySQL (MySQL AB, 2005), a GNU open source software. MySQL is a relational database management system that can handle large amounts of data, allows data access via standard SQL commands, provides multi-user access over TCP/IP and supports several APIs (Application Programming Interfaces) amongst which is C/C++.

The database interface software was developed for the Microsoft Windows environment using Microsoft Visual C++ V6.0. The graphical user interface was based on Microsoft Foundation Classes (MFC), using a simple Document-View architecture with one document and one associated view. MySQL C API was used for the database access from C++ code. Matrix calculations were based on the excellent C++ matrix library NewMat V10B (Davies, 2002) which is available free on the internet.

RESULTS

Field Data Structuring

The structuring of the field data had two main influences on the data collection process: (a) the structure had to be setup before the actual sampling took place, a fact which led to better planned sampling campaigns and (b) the resulting spectral data files were well ordered and could be automatically loaded into the spectral database.

A Spectral Data Management and Processing Software

The combination of a relational database with associated software for data processing was found to be highly efficient while dealing with hyperspectral field spectra from vegetation, soil and pasture studies. Typically, the data from a full day of sampling could be loaded as a new study into the database in a matter of minutes. Subsequent analysis and availability of results at various stages followed almost instantly. The fast data processing allowed the use of the software for the experimental analysis of the influence of different pre-processing parameters on the analysis result. E.g. 1046 spectra of a study of New Zealand native plants could be pre-processed by waveband filtering, Savitzky-Golay smoothing, Hyperion sensor synthesizing and first derivative calculation using Savitzky-Golay coefficients and written to a file in just 10 seconds on a Pentium4 machine.

Collaboration with other researchers has confirmed that the presented solution greatly improved the speed of their research. Operations that would have taken hours or days with conventional ways could be carried out in seconds.

Graphical User Interface

The graphical user interface (GUI) was based on the structure of the processing chain (see Figure 5). The left side of the main window consists of controls for the selection of the study and the main settings for smoothing filter, synthesizing, derivative calculation, feature space transformation and classifier discriminant function. Processing details are entered in pop up windows, shown here with the example of the smoothing function. The text output panel in the middle of the main window is used to display processing and error information.

The listbox on top of the text output panel is used to display spectra files that are loaded directly into memory. The 'Indiv. Classify' button under it classifies the selected, individually loaded spectra against the current library.

The library status box on the top right of the screen indicates whether statistical information has been compiled for the current pre-processing settings.

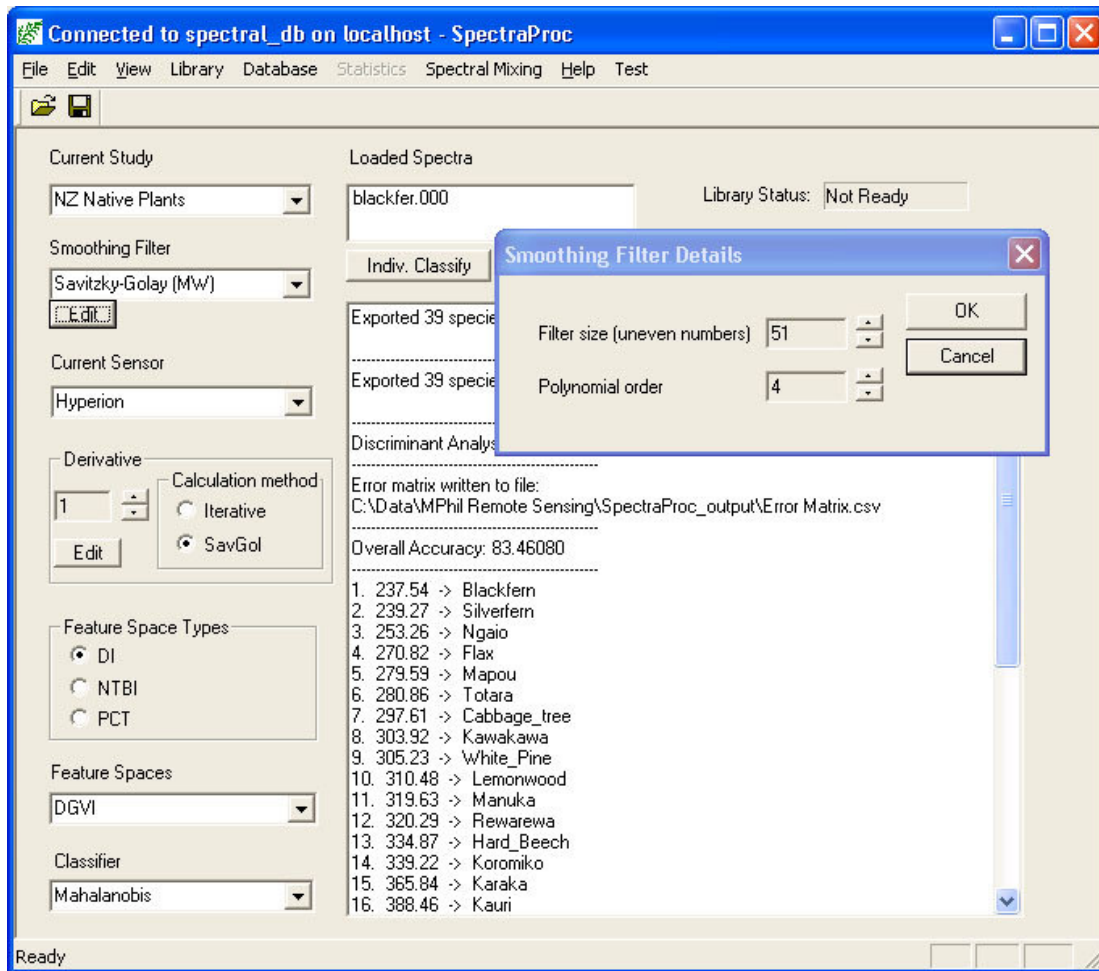


Figure 5: Screen capture of SpectraProc

DISCUSSION

Spectral databases

The database developed for this project proved to be ideal for the data analysis that was carried out. It was however not designed to act as a repository for spectra that could be accessed by persons having no prior knowledge of the stored spectra. Therefore information such as the instrument used, illumination conditions, collector details and extensive target description were not included. Furthermore, the hierarchical structuring featuring species, sites and spectra could be regarded as somewhat restrictive. The experiences gained so far indicate however that the chosen structure applies to most experiments. In some cases the site level might not be needed, but this inconvenience could be solved by a software modification leaving the database structure intact.

The database approach also enables the data to be stored in a central place and offers simultaneous data access to several users. The implemented system however does not offer multi-user capability, i.e. users can not store their own personalized settings.

Future spectral databases should provide multi-user access to studies and more information on the instrumentation and environmental conditions of the sampling. The direct linkage with a geographic information system (GIS) should also be considered when designing the database.

Spectral Processing Chain

The spectral processing chain consisted of waveband filtering, smoothing, sensor synthesizing/downsampling, derivative calculation and feature space transformation. These are the most commonly used operations in hyperspectral studies. It is however clear that the implemented steps are not conclusive. Other data processing such as continuum removal and special indices like band depth indices are in use in the research community. Such operations do not fit into the current chain. Furthermore one could argue about the logical order of the processing steps. E.g. the derivatives could be calculated before or after the data reduction (sensor synthesizing / downsampling). For such a modification, a more flexible approach would be needed where the processing methods would be modularised allowing the interactive building of processing chains.

Analysis Functionality

Only basic analysis functionality in the form of separability, discriminant and principal components analysis was implemented. It was found that the effort in writing analysis functions was only justified if the concerned function was used often. For more rarely used or more complex functions the use of 3rd party software on the pre-processed data proved to be more efficient.

Availability of SpectraProc to the Remote Sensing Community

The presented software has raised considerable interest among potential users and we are currently evaluating different options as to SpectraProc can be made available to the remote sensing community. Expressions of interest are welcome and should be directed to the corresponding author.

CONCLUSION

Fast and repeatable data processing is a key factor to the efficient study of hyperspectral data. By storing the spectral data in a database, all subsequent operations can be carried out on the original dataset which remains unchanged. The implementation of software with a database interface that handled data input, processing and output proved to be a most effective way of hyperspectral data processing. The processing chain developed in this study contains methods that are most commonly used in hyperspectral studies. It is recommended that future processing chains should be of a modular nature to accommodate more varieties of data processing steps. Statistical research should be carried out in other software packages and only if a certain method has proven to be useful and often needed, should it be implemented in the database interface software.

REFERENCES

- Bojinski, S., Schaepman, M., Schlaepfer, D. & Itten, K. (2003). SPECCHIO: a spectrum database for remote sensing applications. *Computers & Geosciences* 29: 27-38.
- Davies, R. (2002). *NewMat*. <http://www.robertnz.net>.
- Elvidge, C. D. & Chen, Z. (1995). Comparison of broadband and narrowband red and near-infrared vegetation indices. *Remote Sensing of Environment* 54: 38-48.
- Landgrebe, D. (1997). *On Information Extraction Principles for Hyperspectral Data*, Purdue University.
- Lillesand, T. M., Kiefer, R. W. & Chipman, J. W. (2004). *Remote Sensing and Image Interpretation*, John Wiley & Sons.
- McFadden, F. R. & Hoffer, J. A. (1988). *Database Management*. Redwood City, The Benjamin/Cummings Publishing Co.
- MySQL AB (2005). *MySQL*. <http://www.mysql.com>.
- Richards, J. A. (1993). *Remote Sensing Digital Image Analysis*. Berlin, Springer Verlag.
- Savitzky, A. & Golay, M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry* 36(8): 1627-1639.

Schmidt, K. S. & Skidmore, A. K. (2003). Spectral discrimination of vegetation types in a coastal wetland. *Remote Sensing of Environment* 85: 92-108.

Shaw, G. & Manolakis, D. (2002). Signal Processing for Hyperspectral Image Exploitation. *IEEE Signal Processing Magazine* 19(1): 12-16.

Thenkabail, P. S., Enclona, E. A. & Ashton, M. S. (2004). Accuracy assessment of hyperspectral waveband performance for vegetation analysis applications. *Remote Sensing of Environment* 91: 354-376.

Tsai, F. & Philpot, W. (1998). Derivative Analysis of Hyperspectral Data. *Remote Sensing of Environment* 66: 41-51.

University of Waikato (2005). WEKA. <http://www.cs.waikato.ac.nz/~ml/weka/>.

Vane, G. & Goetz, A. F. H. (1988). Terrestrial Imaging Spectroscopy. *Remote Sensing of Environment* 24: 1-29.