JJM SYSTEMS INC

B921020-U-2R00

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - FINAL REPORT

NAS5-30127

28 October 1992 IN-43-CR 128405

P-286

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771

Prepared by:

JJM Systems, Inc.
One Ivybrook Boulevard, Suite 190
Ivyland, PA 18974

TABLE OF CONTENTS

LIST OF ACRONYMS

KEE	Knowledge Engineering Environment
VEG	VEGetation Workbench

INTRODUCTION

The NASA VEGetation Workbench (VEG) is a knowledge based system that infers vegetation characteristics from reflectance data. VEG is described in detail in several references (1, 2). The first generation version of VEG has been extended. An interface to a file of unknown cover type data has been constructed. An interface that allows the results of VEG to be written to a file has also been implemented. A learning system that learns class descriptions from a data base of historical cover type data and then uses the learned class descriptions to classify an unknown sample has been built. This system has an interface that integrates it into the rest of VEG. The VEG subgoal PROPORTION.GROUND.COVER has been completed and a number of additional techniques that infer the proportion ground cover of a sample have been implemented.

VEG is written using the Knowledge Engineering Environment (KEE) by Intellicorp. Data and methods are contained in KEE units in the file veg4.u. VEG also uses Lisp methods contained in Lisp files. The file veg-methods.lisp contained the Lisp methods for the first generation version of VEG. Additional Lisp files were created to hold the Lisp code for each of the recent extensions to VEG. However, each extension of VEG also required some minor changes to the file veg-methods.lisp. The current versions of all the VEG files have been delivered to the NASA GSFC technical representative on a Sun cartridge tape.

The interface to an input file of unknown cover type data was described in detail in the JJM Systems report B921015-U-2R01. This report is included as Appendix A. The interface is available when VEG is operating in the "Automatic Mode." The interface enables the user to name the input file and specify the format for the file. The format can be a standard, previously-defined or newly-defined format. Using this format, the input file is read and the cover type data is stored for processing in the system. Formats that have been newly defined using the interface can be stored for subsequent use. The report describes the testing of all the options in the input file interface.

In the previous version of VEG, the results of processing were displayed on the screen if the "Research Mode" was in use. In the "Automatic Mode," the results were written to a file in a simple format. The options available for writing the results of VEG to a file have been extended. The new options are available in both VEG modes. These options are described in detail in JJM Systems report B921016-U-2R02 which is included as Appendix B.

If the "Research Mode" is in use, the results from VEG are first displayed on the screen. The user then is asked whether the results should be written to a file. If the user left clicks on "YES," he/she is prompted for the name of the file. If the named file exists, the user is asked to name a new file or confirm that the existing file can be overwritten. The user is then asked to select the names of the parameters that should be written to the file and the format that is to be used. Alternatively, the user can select a standard template. Once the format has been selected, the results are written to the file.

When VEG is being run in the "Automatic Mode," the user is required to enter the name of the output file before the data is processed. When the user enters the output file name, the interface described in the previous paragraph is opened and the user selects the format for the file. However, the results for each sample are not written to the file until after the data has been processed. All options in the output interface were tested.

The most significant recent extension to VEG was the implementation of the learning system. This system is described in JJM Systems report B921014-U-2R03 which is included in Appendix C. As part of this work, the browser in VEG was extended so that any knowledge base including the learning system and the KEE systems knowledge bases can be browsed.

The learning system can be operated in both VEG modes. In the "Research Mode," the learning system presents the user with three different options. In option 1, the system uses the data base of historical cover type data to learn class descriptions of one or more classes of cover types. These classes can include broad classes such as soil or vegetation or more specific classes such as forest, grass and wheat. The classes can also include subclasses based on continuous parameters such as 0-30% ground cover, 31-70% ground cover and 71-100% ground cover. The learning system is designed to handle any combination of directional view angles such as (0 0), (30 50), (45 60), (10 135), (40 225) where the first value in each pair is the zenith angle and the second value is the relative azimuth angle. The learning system uses sets of positive and negative examples from the data base of historical cover types to find the most important features that uniquely distinguish each class. For example, the learning system found that for solar zenith 45°, wavelength 0.68 μm and the view angles (15 182), (75 90), (0 0) and (35 45), the class 0-30% ground cover was best distinguished by the following hypotheses:-

((FIRST-MAX (75 90)) NIL)
((FIRST-MAX (15 182)) T)
((GREATER-THAN (15 182) (75 90)) T)

The first hypothesis says that the maximum reflectance value is not at the view angle (75 90). The second hypothesis says that the maximum reflectance value is at the view angle (15 182). The third hypothesis says that the reflectance at the view angle (15 182) is greater than the reflectance at the view angle (75 90). These hypotheses describe the class 0-30% ground cover for the solar zenith angle, view angles and wavelength specified. In a typical run of the learning system, class descriptions for several alternative classes such as the classes 0-30%, 31-70% and 71-100% ground cover are learned from the data base of historical cover types. In option 2, the system learns class descriptions for one or more classes and then uses the learned classes to classify an unknown sample by finding the class that best matches the unknown cover type data. Option 3 allows the user to test the system's classification performance. In this option, the system learns class descriptions for one or more classes and then classifies the appropriate samples in the data base. The percentage of correctly classified samples is then used to summarize the degree of classification accuracy achieved by the learning system.

The learning system can also be run in the VEG "Automatic Mode." In this mode, the user enters the classes that are to be learned. Samples of unknown cover type data are read from a file and stored in the system. The data is processed a sample at a time. The wavelengths, view angles and solar zenith of the sample, together with the training classes define the training problems. The system learns class descriptions of the training problems and uses these to classify the unknown sample. The classification performance of the system is also calculated.

The VEG subgoal PROPORTION.GROUND.COVER has been completed and a number of additional techniques that infer the proportion ground cover of a sample have been implemented. Some techniques operate on sample data at a single wavelength. The techniques previously incorporated in VEG for estimating SPECTRAL.HEMISPHERICAL.REFLECTANCE and VIEW.ANGLE.EXTENSION operated on data at a single wavelength, so implementing the additional single wavelength techniques required no changes to the structure of VEG. Two techniques which use data at multiple wavelengths to infer proportion ground cover were also implemented. This work involved modifying the structure of VEG so that multiple wavelength techniques could be incorporated. All the new techniques were tested using both the VEG "Research Mode" and "Automatic Mode." This work is described in the JJM Systems report B921019-U-2R04 which forms Appendix D of this report.

REFERENCES

1. Kimes, D. S., Harrison, P. R. and Ratcliffe, P. A. A Knowledge-Based Expert System for Inferring Vegetation Characteristics, International Journal of Remote Sensing, Vol 12, 10, pp. 1987-2020, 1991.

2. Kimes, D. S., Harrison, P. A. and Harrison, P. R. New Developments of a Knowledge Based System (VEG) for Inferring Vegetation Characteristics, International Geoscience and Remote Sensing Symposium, Houston, Texas, May 1992.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle An Expert System Shell for Inferring Vegetation Characteristics - Final Report		5. Report Date October 1992 6. Performing Organization Code
7. Author(s) P. Ann Harrison Patrick R. Harrison		8. Performing Organization Report No. B921020-U-2R00 10. Work Unit No. 462-61-14
9. Performing Organization Name and Address JJM Systems, Inc. One Ivybrook Blvd., Suite 190 Ivyland, PA 18974		11. Contract or Grant No. NAS5-30127
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001 NASA/Goddard Space Flight Center Greenbelt, MD 20771		13. Type of Report and Period Covered Final Report July - October 1992 14. Sponsoring Agency Code

15. Supplementary Notes

The Lisp and KEE code for this work is available on a Sun Cartridge Tape.

16. Abstract

The NASA VEGetation Workbench (VEG) is a knowledge based system that infers vegetation characteristics from reflectance data. The report describes the extensions that have been made to the first generation version of VEG. An interface to a file of unknown cover type data has been constructed. An interface that allows the results of VEG to be written to a file has been implemented. A learning system that learns class descriptions from a data base of historical cover type data and then uses the learned class descriptions to classify an unknown sample has been built. This system has an interface that integrates it into the rest of VEG. The VEG subgoal PROPORTION.GROUND.COVER has been completed and a number of additional techniques that infer the proportion ground cover of a sample have been implemented.

17. Key Words (Suggested by Author(s)) EXPERT SYSTEM, ARTIFICIAL INTELLIGENCE, REMOTE SENSING, LEARNING, DISCRIMINATION		18. Distribution Statement UNCLASSIFIED - UNLIMITED	
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of pages	22. Price

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171

APPENDIX A

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - INTERFACE TO A FILE OF UNKNOWN COVER TYPE DATA (TASK A)

JJM SYSTEMS INC

B921015-U-2R01

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - INTERFACE TO A FILE OF UNKNOWN COVER TYPE DATA (TASK A)

26 September 1992

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771

Prepared by:

JJM Systems, Inc.
One Ivybrook Boulevard, Suite 190
Ivyland, PA 18974

JJM SYSTEMS INC

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

KEE	Knowledge Engineering Environment
VEG	VEGetation Workbench

SECTION 1.0

INTRODUCTION

The current version of the VEGetation workbench (VEG) can be applied to sample data stored in the system and to data entered by the user at run time. VEG can also be applied to data stored in a file in a simple format when VEG is run in automatic mode. Various scientists have conducted experiments and produced data on which VEG could operate. It is anticipated that large volumes of reflectance data will be available in the future as advances are made in the technology for collecting the data from experiments on the ground and on aircraft and satellites. In order to enter this data into VEG, an interface is required. This interface will allow the scientist to select from a list of previously defined file formats or define a new file format, and then activate a method to read the file and enter the data into the system.

The techniques available in VEG operate on directional spectral reflectance data stored in Knowledge Engineering Environment (KEE) units which are subclasses and instances of the unit TARGET.DATA. As shown in Figure 1-1, twenty-one member slots were created in the unit TARGET.DATA, and these slots are inherited by all subclasses and instances of this unit. Some slots hold the data that is entered for unknown cover types. Other slots hold the intermediate results of processing the data. Most of the data, including the cover type description and the solar zenith, is entered at the sample level. Data, for the sample measured at different wavelengths, is stored in instances of the sample. For example, the units W1 and W2 hold wavelength and reflectance data measured for Sample 1 at wavelengths 0.68μm and 0.92μm, respectively. This is shown in Figure 1-2. In order to apply VEG to files of directional spectral reflectance data of unknown cover types, additional KEE units such as FILE-SAMPLE-1, W19, W20, and W21 are created dynamically as the data is read in.

COMPLETE DESCRIPTION
COVER.TYPE.DESCRIPTION
DRY.BIOMASS
GC.CALC.P
GROUND COVER
HEIGHT
I.E.RHD
INT.EXT.DATA
LAI.CALC.P
LEAF.AREA.INDEX
MEAN.SEPARATION
NADIR
NUMBER.VIEW.ANGLES
PORTION.GREEN
R.D.S
REFLECTANCE.DATA
SOLAR.ZENITH
STRING.OBJECTS
TECHNIQUES
WAVELENGTH
WET.BIOMASS

Figure 1-1

Slots in the Unit TARGET.DATA

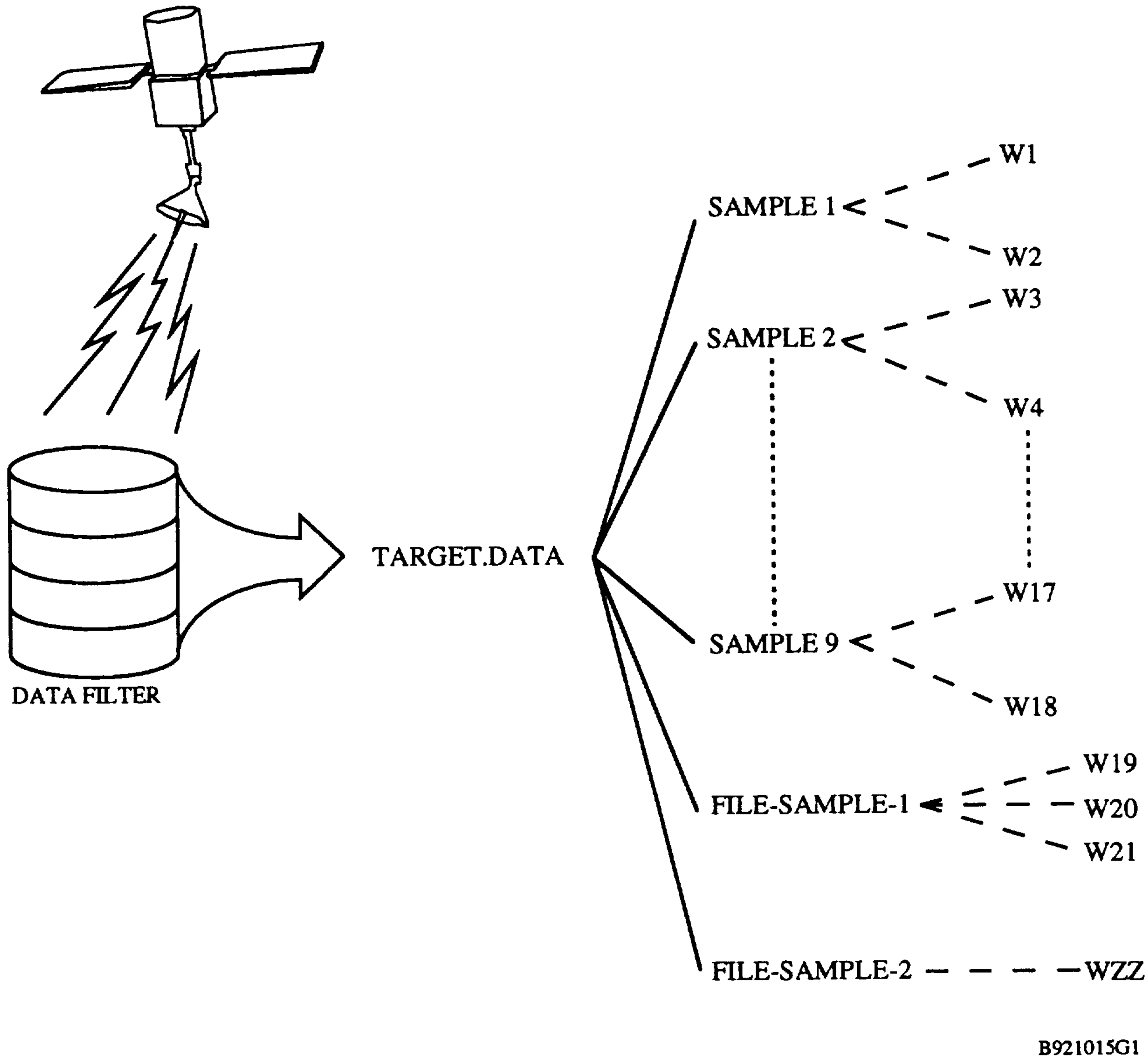


Figure 1-2

Cover Type Data from the File is Stored in KEE Units in VEG

Task A has been completed. An interface to a file of unknown cover type data has been implemented. This interface allows the scientist to select a file format from a list of previously defined file formats. It also allows the scientist to define a new file format and then store this file format for future use. This capability will allow VEG to be applied to new data files in new formats. The interface is described in detail in this report. The code for the Lisp methods involved is included in Appendix A. A sun cartridge tape containing these Lisp methods and the current version of VEG, including the completed interface to a file of unknown cover types, has been delivered to the NASA GSFC technical representative.

SECTION 2.0

DETAILED DESCRIPTION OF THE INTERFACE

VEG can be operated in two different modes. In the "Research Mode," the scientist must separately execute each step in the processing of unknown cover type data. This mode allows the scientist to study the intermediate results in detail. VEG can also be operated in the "Automatic Mode." In this mode, the scientist selects the operations to be carried out. Then the cover type data is read from a file, processed and the results are output to another file without any further intervention from the user. The NASA GSFC technical representative advised that the interface to a file of unknown cover type data should be incorporated into the running of VEG in the "Automatic Mode" but not the "Research Mode." It was decided that it would not be practical to apply VEG in the "Research Mode" to large volumes of data from a file.

When the user selects the "Automatic Mode," the screen shown in Figure 2-1 is displayed. This screen enables the scientist to specify how the system is to be run, including naming the input file and selecting the VEG goal such as SPECTRAL.HEMISPHERICAL.REFLECTANCE.

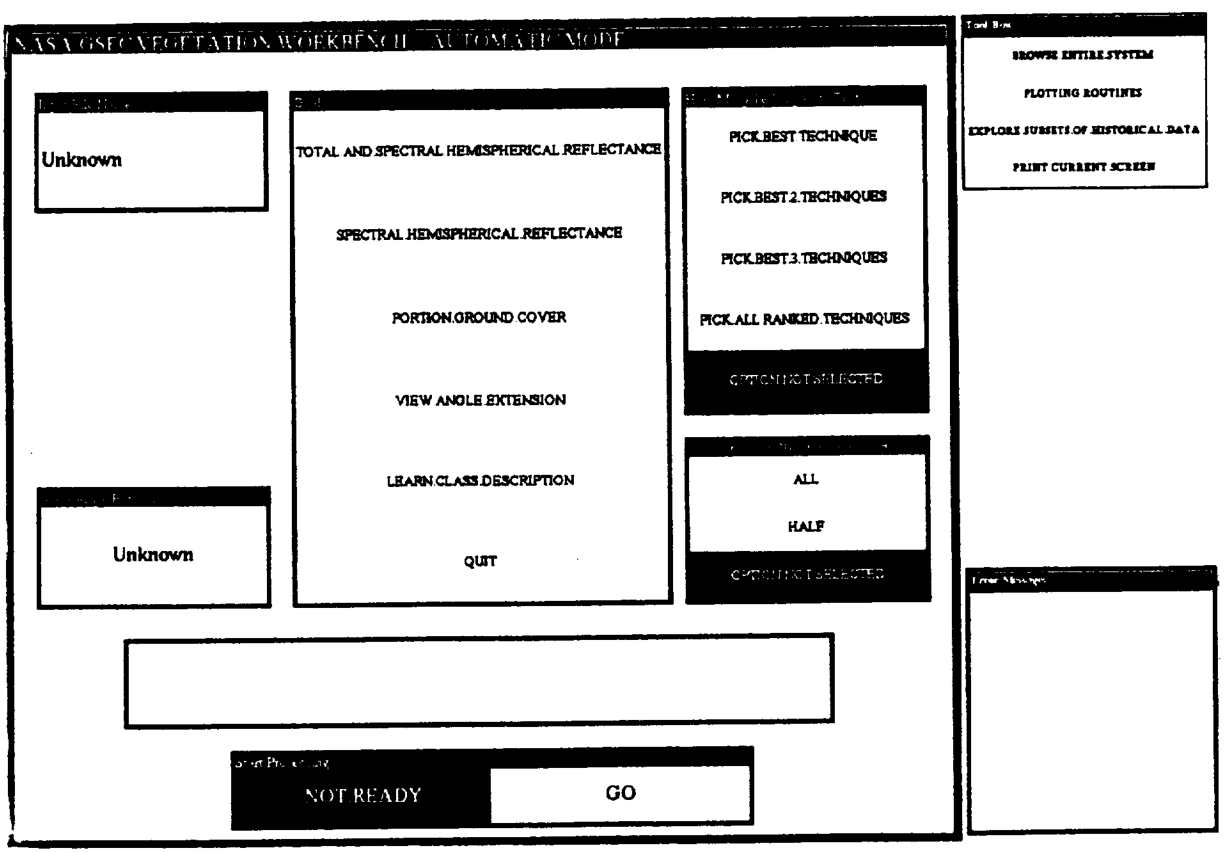


Figure 2-1

The Main Screen in the VEG "Automatic Mode"

In order to enter an input file name, the user must left click on the box labelled "Input File Name" (Figure 2-1) and then type in the name of the required file. This name is automatically stored in the slot AUTO.INPUT.FILE.NAME of the unit AUTOMATIC.PROCESS. An ActiveValue unit is attached to this slot. Each time the value in the slot is changed, a method stored in the AVPUT slot of the ActiveValue unit is activated. This method checks that the file exists. If it does not exist, an error message is displayed in the "Error Messages" box. If the file does exist, the screen shown in Figure 2-2 is displayed. This screen enables the user to specify the format for the file.

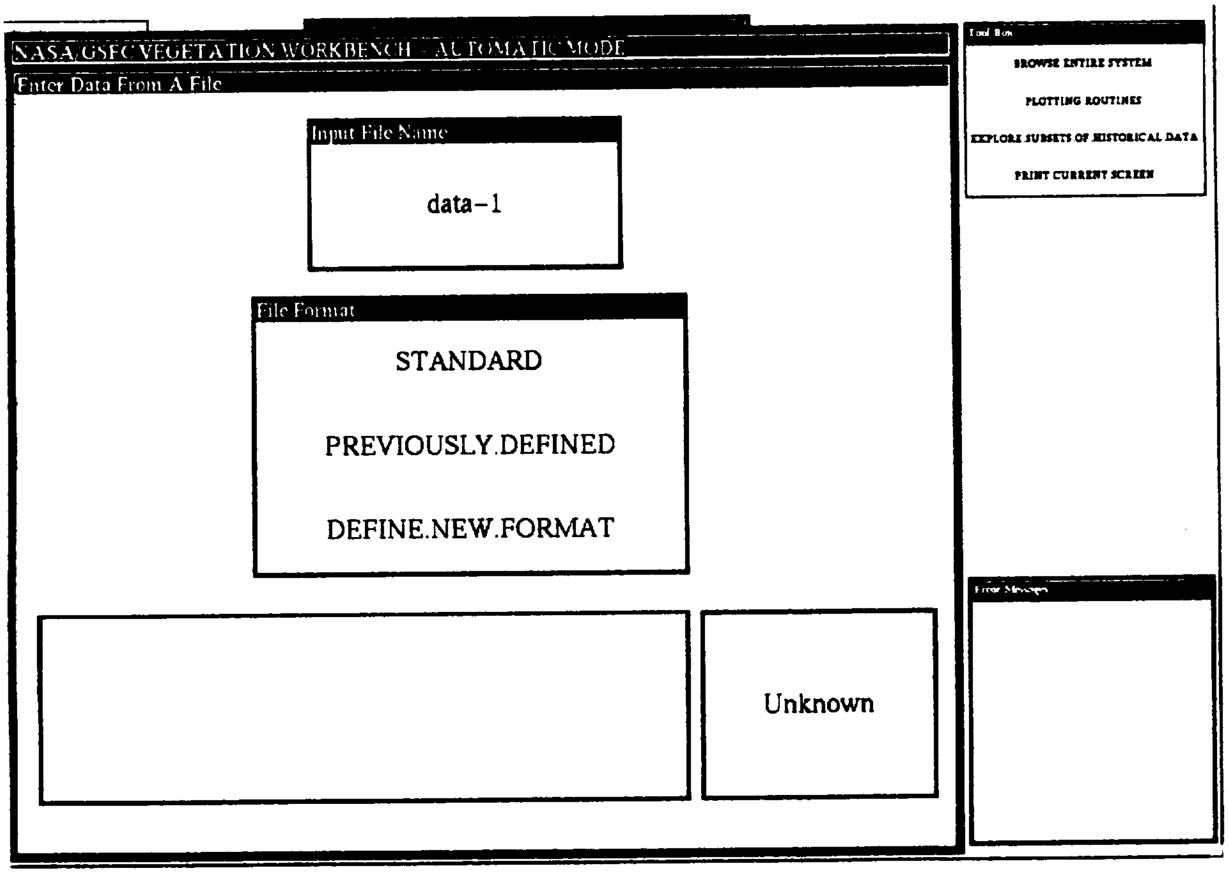


Figure 2-2

The Screen that Enables the User to Specify the File Format

Certain assumptions have been made about the format of the data in the file. It is assumed that the data is in ASCII format and that each data item is separated by white space. It is also assumed that the data which applies to all wavelengths for a particular cover type is placed before the wavelength-specific data in each record. The only permissible wavelength-specific fields are the wavelength, number of view angles, and the reflectance data. The reflectance data can be in one of three formats : a list of lists such as ((0 0 0.2)(30 45 0.3)), a list for each data point such as (30 45 0.3) or as separate data items such as 30, 45 and 0.3. In each case, it is assumed that each data point is specified by a zenith, azimuth, and reflectance values in that order. If either the second or third format is specified, the reflectance data must be preceded by the number of view angles. The file can contain some or all of the possible fields. However, a file must contain at least solar zenith, wavelength and reflectance data. The file may not contain additional fields that are not known to the system. The complete description and cover type description of the cover type must be in string format surrounded by double quotes.

As shown in Figure 2-2, the user can choose among three options for specifying the input file format. These are a standard format, previously defined format, or new format.

2.1 STANDARD FILE FORMAT

The previous version of VEG could be applied to file data in a simple format. This format is referred to as the "standard" format in the current version of VEG. An example record in this format is shown in Table 2-1. A global variable *STANDARD-SAMPLE-FORMAT* contains a list of fields at the sample level in the standard format and it has the value (COVER.TYPE.DESCRIPTION SOLAR.ZENITH LEAF.AREA.INDEX GROUND.COVER PORTION.GREEN DRY.BIOMASS WET.BIOMASS HEIGHT NUMBER.WAVELENGTHS). The global variable *STANDARD-WAVELENGTH-FORMAT* contains a list of fields at the wavelength level in the standard format, and it has the value (WAVELENGTH NUMBER.VIEW.ANGLES REFLECTANCE.DATA). The field names correspond to the names of the slots in which the data should be stored. If the user selects the standard format option from the file format menu shown in Figure 2-2, the values of *STANDARD-SAMPLE-FORMAT* and *STANDARD-WAVELENGTH-FORMAT* are put in the slots AUTO.INPUT.SAMPLE.FORMAT and AUTO.INPUT.WAVELENGTH.FORMAT of the unit AUTOMATIC.PROCESS. The screen shown in Figure 2-2 is then closed and the "Automatic Mode" main screen is visible once again.

Table 2-1

Standard File Format and an Example of Typical Values

Field Names	Typical Values
COVER.TYPE.DESCRIPTION	"Dense vegetation canopy"
SOLAR.ZENITH LEAF.AREA.INDEX	45 3.5
GROUND.COVER PROPORTION.GREEN	0.7 0.3
DRY.BIOMASS WET.BIOMASS HEIGHT	0.2 0.5 1000
NUMBER.WAVELENGTHS	2
WAVELENGTH	0.68
NUMBER.VIEW.ANGLES	1
REFLECTANCE.DATA	((0 0 0.43))
WAVELENGTH	0.92
NUMBER.VIEW.ANGLES	3
REFLECTANCE.DATA	((0 0 0.5)(30 45 0.51)(60 45 0.62))

B921015G3

2.2 SELECTING A PREVIOUSLY DEFINED FILE FORMAT

Formats that were defined using the DEFINE.NEW.FORMAT option can be stored for future use, and then selected using this option. The slot PREVIOUSLY.DEFINED.FORMATS contains a list of three elements for each previously defined format. These values are the format name, a list of fields at the sample level and a list of fields at the wavelength level, e.g., (SIMPLE (SOLAR.ZENITH NUMBER WAVELENGTHS)(WAVELENGTH REFLECTANCE.DATA).

When the user selects the option PREVIOUSLY.DEFINED from the File Format menu, a list of previously defined format names is displayed in the box below the menu, as shown in Figure 2-3. In this figure there is one previously defined format with the name "SIMPLE." The user can select a format by left clicking on the box to the right of the message box and then typing in the name of the selected format. The format name is stored in the slot AUTO.INPUT.FILE.FORMAT.NAME of the unit AUTOMATIC.PROCESS. An ActiveValue unit is attached to this slot. When the format name is entered, the method in the ActiveValue unit is activated. If the name that was typed in is not a valid format name, an error message is displayed in the box labelled "Error Message," and the user is prompted to re-enter the format name. Otherwise, the list of fields at the sample and wavelength levels for the selected format are stored in the slots AUTO.INPUT.SAMPLE.FORMAT and AUTO.INPUT.WAVELENGTH.FORMAT respectively of the unit AUTOMATIC PROCESS. The screen shown in Figure 2-3 is then closed, and the main menu for the "Automatic Mode" is once again visible.

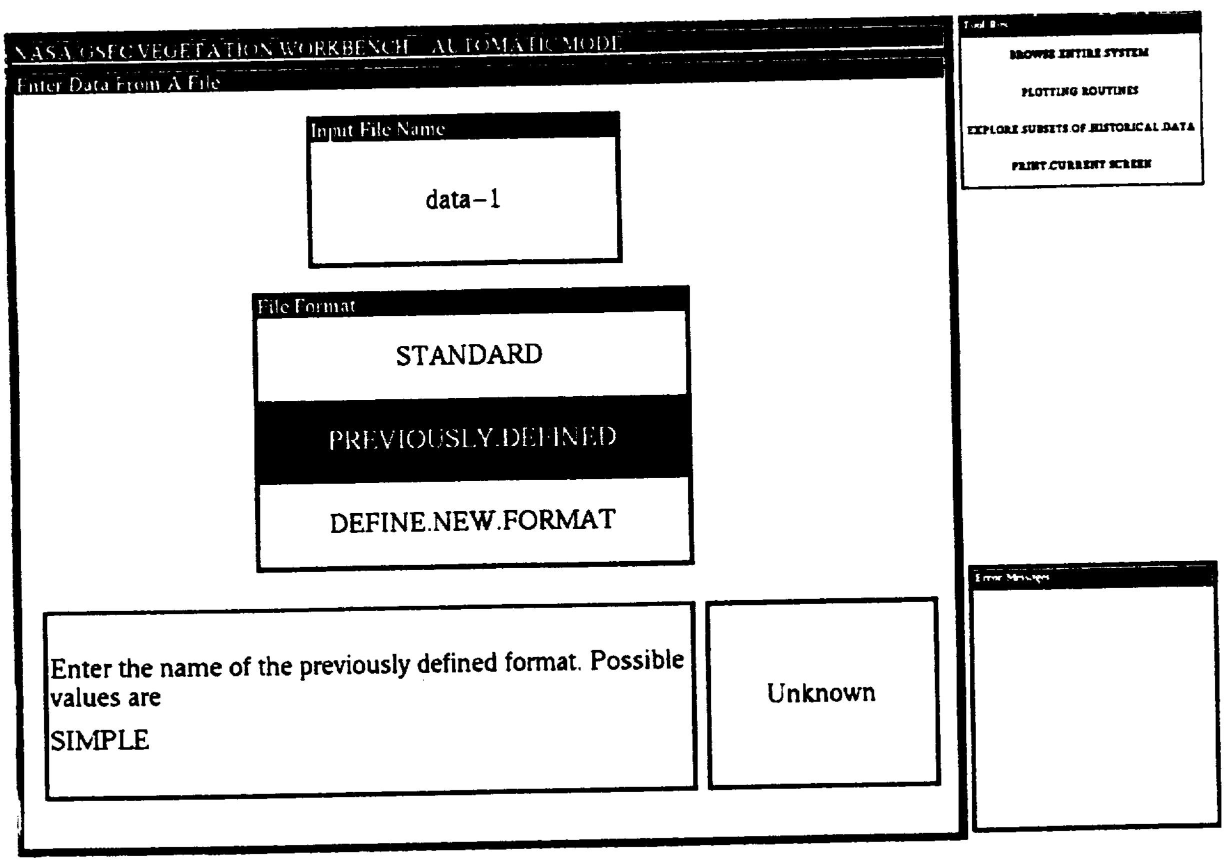


Figure 2-3

Selecting a Previously Defined Format

2.3 DEFINING A NEW FORMAT

If the user selects the option DEFINE.NEW.FORMAT from the File Format menu, the screen shown in Figure 2-4 is displayed. The user is prompted to enter the field numbers for all the data items in the input file. For example, if the first field on the file contains the solar zenith, the user must enter a "1" in the box labelled "Solar Zenith." If the new format is to be stored for future use, the user must enter a name for the new format in the box labelled "New Format Name." All the boxes in the "Define New File Format" screen are attached to slots in the unit AUTOMATIC.PROCESS so the values entered in the boxes are stored in the slots in this unit. When the user has entered all the field numbers he/she should left click on "DONE." This box is attached to the slot AUTO.INPUT.DONE of the unit AUTOMATIC.PROCESS and an ActiveValue unit is attached to this slot. When the user left clicks on "DONE," a method in the ActiveValue unit is activated. This method calls the Lisp function FINISH-FORMAT-DEFINITION.

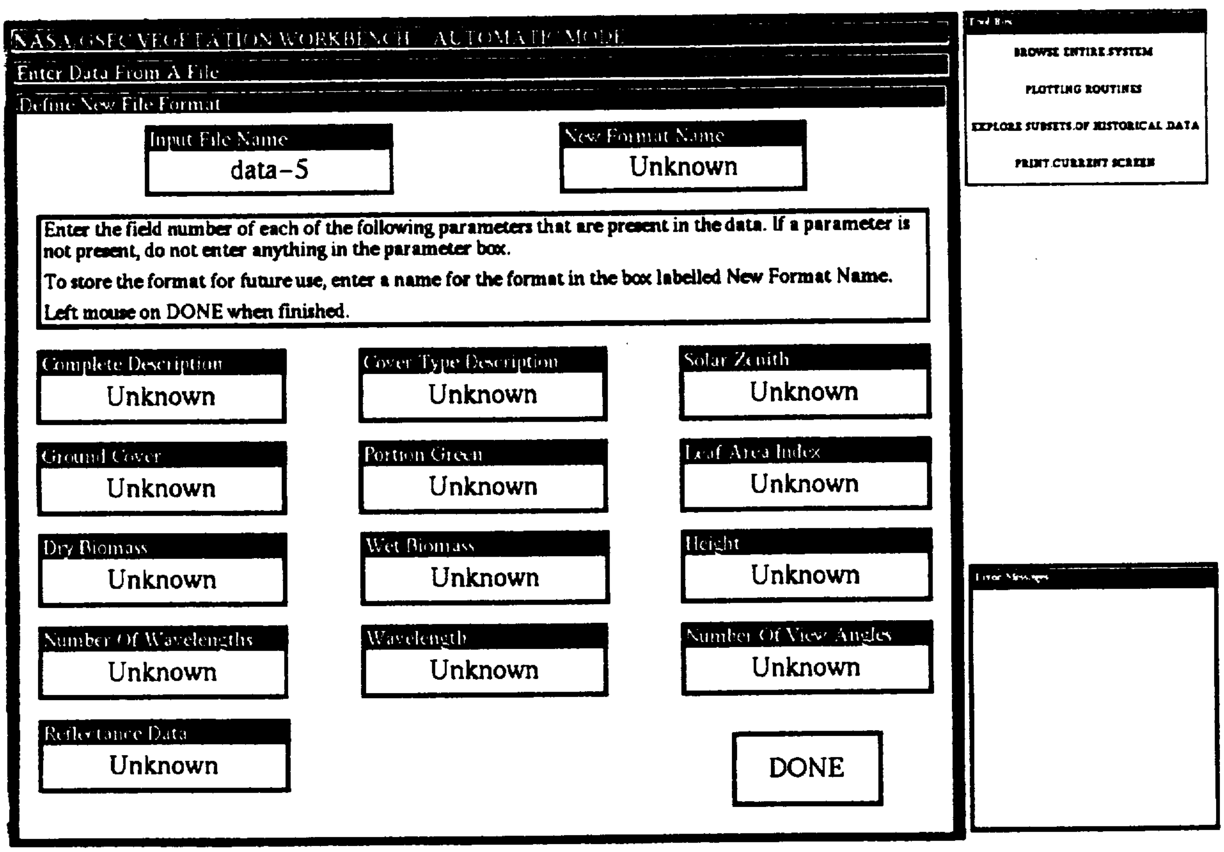


Figure 2-4

The Screen for Defining a New Format

The Lisp function FINISH-FORMAT-DEFINITION constructs a list of fields in the new format at the sample and wavelength levels. The function first checks that the format includes fields for solar zenith, wavelength, and reflectance data. This is the minimum set of data that VEG needs to process an unknown cover type. If any of these fields has not been given a field number, the user is prompted to add the field-numbers. Otherwise, the function PROCESS-SLOTS is called to process the sample level and then the wavelength level fields. A list of slots and field numbers is collected and then sorted by field number and stored in the local variable PRESENT-SLOTS, e.g. ((SOLAR.ZENITH 1)(NUMBER.WAVELENGTHS 2)). The function VALIDATE-FIELD-NUMBERS is called to check that the field numbers are consecutive, with the sample level fields preceding the wavelength level fields. If any of the field numbers are invalid, the user is prompted to make corrections. Otherwise, the list of slots is extracted from the list PRESENT-SLOTS. The list of slots at the sample and wavelength levels are then stored in the slots AUTO.INPUT.SAMPLE.FORMAT and AUTO.INPUT.WAVELENGTH.FORMAT in the unit AUTOMATIC.PROCESS.

If a name for the new format has been entered, a list is constructed consisting of the format name, the list of slots at the sample level, and the list of slots at the wavelength level. This list is added to the slot PREVIOUSLY.DEFINED.FORMATS of the unit AUTOMATIC.PROCESS. The unit AUTOMATIC.PROCESS contains a slot NEW.FORMAT.DEFINED.P which has an initial value of NIL. When a new format is defined and stored for future use, the value of the slot NEW.FORMAT.DEFINED.P is changed to T.

The new format is stored in the slot PREVIOUSLY.DEFINED.FORMATS and written to a permanent file when the user comes out of automatic mode. Each time the user selects the VEG "Automatic Mode," the file format-definitions is read, and the previously defined formats are stored once again in the slot PREVIOUSLY.DEFINED.FORMATS of the unit AUTOMATIC.PROCESS.

When the user has entered a valid new format definition, the screens shown in Figures 2-2 and 2-4 are closed and the main menu for the VEG "Automatic Mode" is once again visible.

2.4 READING AN INPUT FILE USING THE CORRECT FORMAT

Whether the user chose to use the standard format, a previously defined format, or a new format, a list of fields at the sample level and wavelength level are stored in the slots AUTO.INPUT.SAMPLE.FORMAT and AUTO.INPUT.WAVELENGTH.FORMAT when the format is specified. Before the data can be processed, the user must select the required options on the main screen for the VEG "Automatic Mode" shown in Figure 2-1 and then left click on "GO."

The first step in processing the data for all of the VEG goals except LEARN.CLASS.DESCRIPTIONS is to read in the data from the file using the Lisp function INPUT-DATA-FROM-FILE. This function opens the file and then calls the function STORE-SAMPLE-DATA with the arguments the file-stream, the list of sample level fields, and the list of wavelength level fields. Units are created as necessary to hold the data as in Figure 1-2. Using the field lists, the data is read from the file and stored in the appropriate slots of the newly created units.

The reflectance data can be in several different formats as previously discussed. VEG operates on reflectance data arranged as a list of lists such as ((0 0 0.5)(30 45 0.56)). (This particular example represents two view angles.) The reflectance data is converted into this format before it is stored in the knowledge base.

A check for the end of file is made each time the file is read. If the end of file is encountered prematurely, the function ABORT-READING is called. This function displays an error message, aborts the processing of the data and re-initializes the system. Each time data is read from the file, the data is checked to ensure that its value is in range for the slot in which it is to be stored. If the data is out of range, the function ABORT-READING is called and the processing of data is aborted. The premature occurence of the end of file or the failure of a validity check on a data value may be an indication that the wrong format has been specified for a file.

SECTION 3.0

TESTING THE INTERFACE

Using data from a file all the options in the interface were comprehensively tested. Six short data files were constructed for testing purposes. These files are listed in Appendix B. The data in these files was invented by the developer. In each data file, data values were of the correct type and within the valid range of values for the field. Discussion of these tests focuses on the selection of the correct file format, and the setting up of the appropriate KEE units within VEG to store the data read in from the file.

3.1 TEST 1

The file data-1 consisted of 1 record which contained only one wavelength. The file was in the "STANDARD" format, and the format for the reflectance data was a list of lists. This simple file was used for the initial testing of the interface.

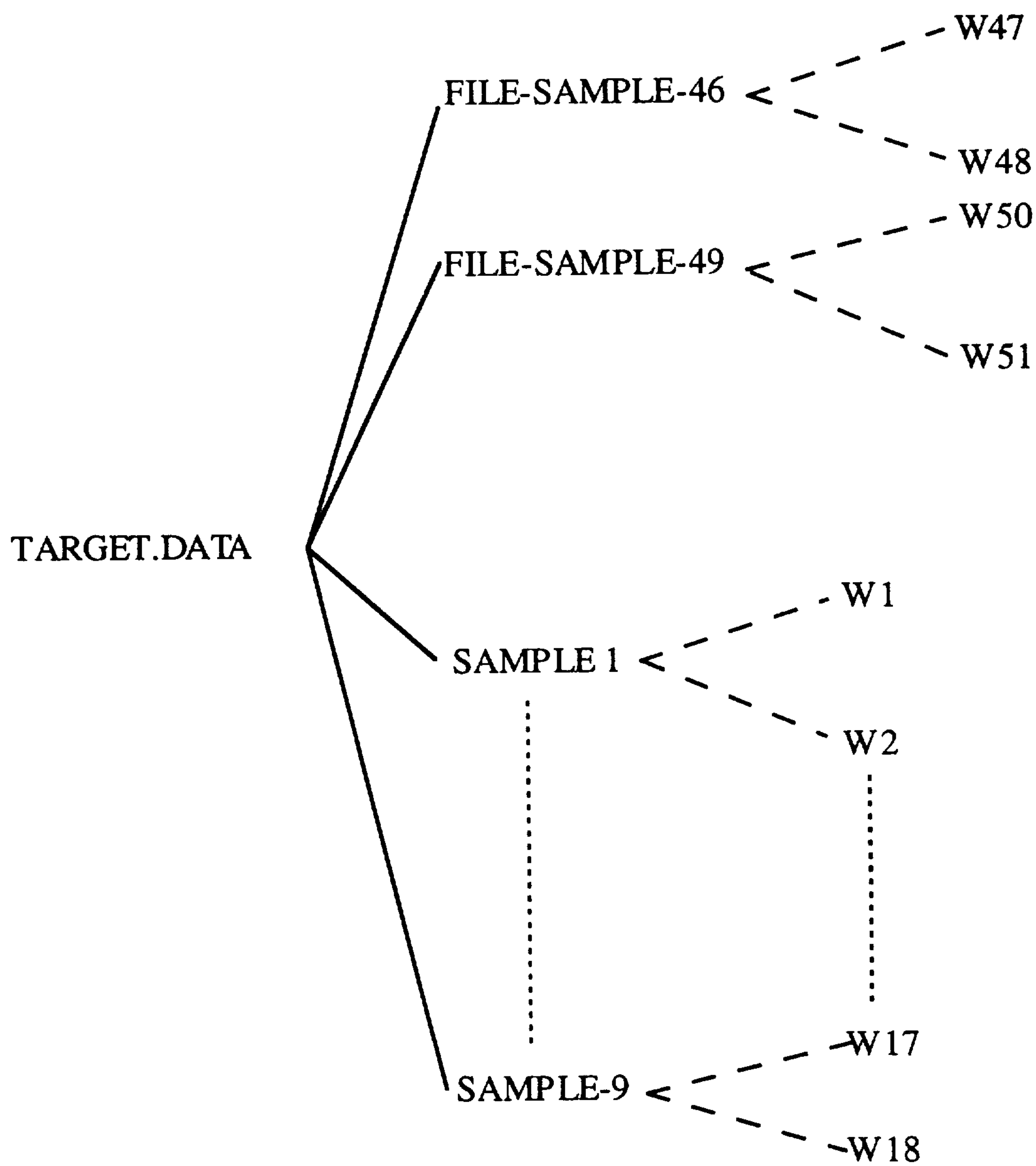
In the first test run, the file data-1 was named as the input file, and the file format STANDARD was selected from the file format menu. The goal SPECTRAL.HEMISPHERICAL.REFLECTANCE was selected from the main Automatic Process menu. After the option GO was selected, a unit called FILE-SAMPLE-1 was created as a subclass of the unit TARGET.DATA. The cover type description and solar zenith values were correctly stored in this unit. A unit called W19 was created as a member of FILE-SAMPLE-1, and the wavelength and reflectance data were successfully stored in this unit. VEG proceeded to complete the processing of this data successfully.

3.2 TEST 2

In test 2, the data file data-2 and the file format STANDARD were selected. The file contained two records each of which contained data at two different wavelengths. The purposes of this test were to check that the correct hierarchy of KEE units was set up and to check that the data was stored correctly. The goal SPECTRAL.HEMISPHERICAL.REFLECTANCE and the option GO were selected. The test was successful. The correct hierarchy of units was set up. This is shown in Figure 3-1. The data was stored in the correct slots, and VEG completed the processing of the data successfully.

3.3 TEST 3

The purpose of this test was to confirm that the program would deal correctly with reflectance data stored in the file as a separate list of values for each point. In tests 1 and 2, the reflectance data for each wavelength was stored as a list of lists. The file data-3 was selected for test 3. This file was in the STANDARD format. The operations carried out in test 2 were repeated using this file. The test was successful. The reflectance data for each wavelength was read from the file and a list of lists was constructed from the reflectance data before it was stored in the appropriate VEG unit. The data was stored correctly and processed successfully.



B921015G2

Figure 3-1

The Hierarchy of Units Set Up to Hold the Data in Test 2

3.4 TEST 4

The purpose of this test was to test the third possible format for the reflectance data. The data file for this test was data-4. This file was identical to data-3 except for the format of the reflectance data. In the file data-4 the reflectance data was stored as separate values, with no parentheses around the set of data for each view angle. The operations carried out in test 2 were repeated using this data file. This test was successful. The reflectance data for each wavelength was correctly read from the file and converted to the required format for VEG before being stored in the appropriate VEG unit. The data was processed successfully by the system.

3.5 TEST 5

The purpose of tests 5 and 6 was to test the DEFINE.NEW.FORMAT option that was described in Section 2.3. In each of these tests, a new format was defined and then a file was read using the newly defined format. The formats defined in tests 5 and 6 were stored and then used to test the option PREVIOUSLY.DEFINED.FORMAT in tests 7 and 8.

In test 5, the data file data-5 and the file format option DEFINE.NEW.FORMAT were selected. The field numbers for the format of the file data-5 were entered into the interface, as shown in Figure 3-2. The new format was named data-5-format. After the user left clicked on DONE, the slots AUTO.INPUT.SAMPLE.FORMAT and AUTO.INPUT.WAVELENGTH.FORMAT held the lists (SOLAR.ZENITH GROUND.COVER NUMBER.WAVELENGTHS) and (WAVELENGTH NUMBER.VIEW.ANGLES REFLECTANCE.DATA) respectively. The list (DATA-5-FORMAT (SOLAR.ZENITH NUMBER.WAVELENGTHS) (WAVELENGTH NUMBER.VIEW.ANGLES REFLECTANCE.DATA)) was correctly added to the slot PREVIOUSLY.DEFINED.FORMATS of the unit AUTOMATIC.PROCESS. The goal SPECTRAL.HEMISPHERICAL.REFLECTANCE and the option GO were then selected. The correct units were created to hold the data, the data was read successfully from the file using the newly defined format and the data was processed successfully. The test was successful.

3.6 TEST 6

In this test, the input data file was data-6. The file format option DEFINE.NEW.FORMAT was once again selected. This test was designed to test the option DEFINE.NEW.FORMAT and, in particular, to test whether the program would detect invalid format definitions.

VEG needs, as a minimum, solar zenith, wavelength and reflectance data for each sample. The interface was programmed to disallow file formats that did not contain these three fields. The field numbers shown in Figure 3-3 were entered and the option DONE selected. No field number was entered for the solar zenith. The error message "Invalid field numbers - re-enter" was displayed in the error message box. This test was successful.

The field numbers in a new format must be consecutive with the sample level fields before the wavelength level fields. The field numbers shown in Figure 3-4 were entered, and the option DONE was selected. The field number 2 was missing. As expected, the error message was once again displayed in the error message box. The field numbers were then changed to those shown in Figure 3-5. In this example, the field number for the wavelength was less than the field number for the solar zenith. When the option DONE was selected again, the error message was again displayed in the error message box. This test was successful.

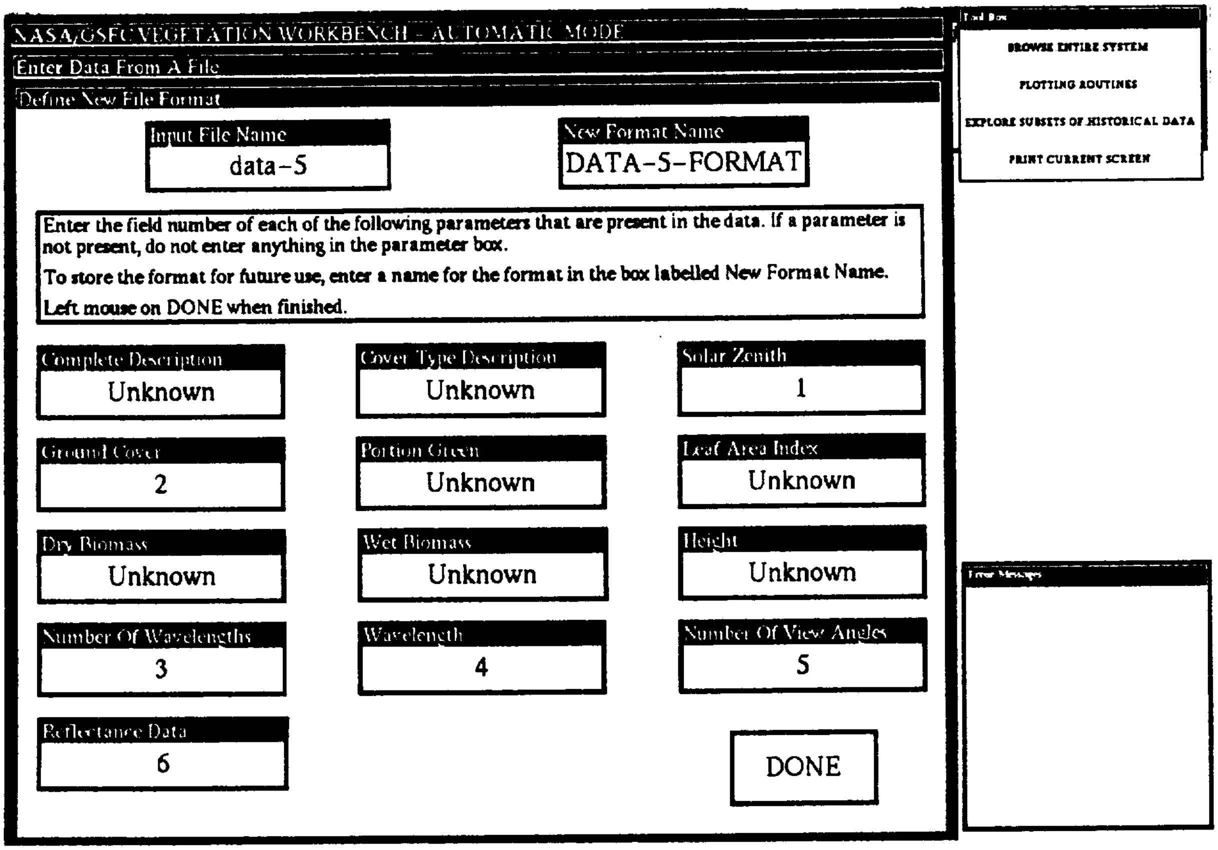


Figure 3-2

Defining the Format for the File data-5

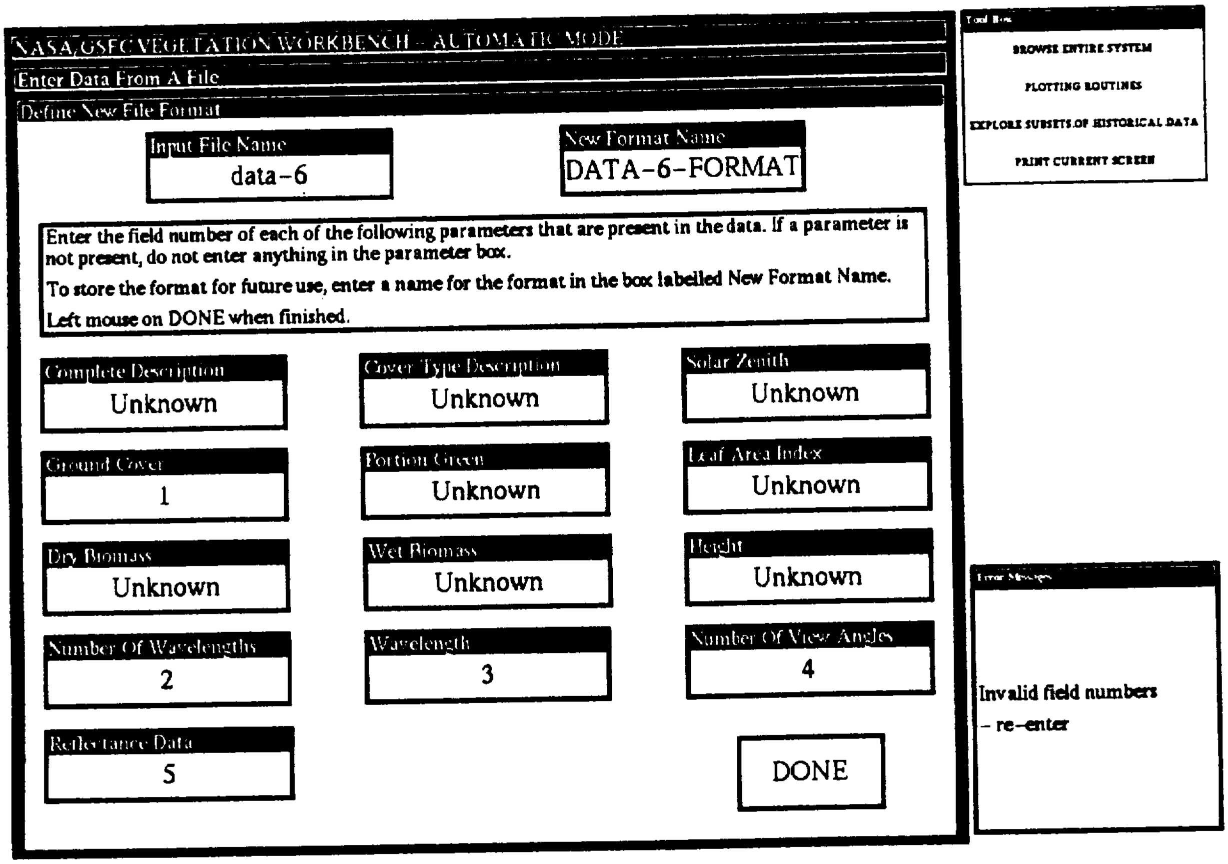


Figure 3-3

Attempting to Define a Format Excluding the Solar Zenith Field

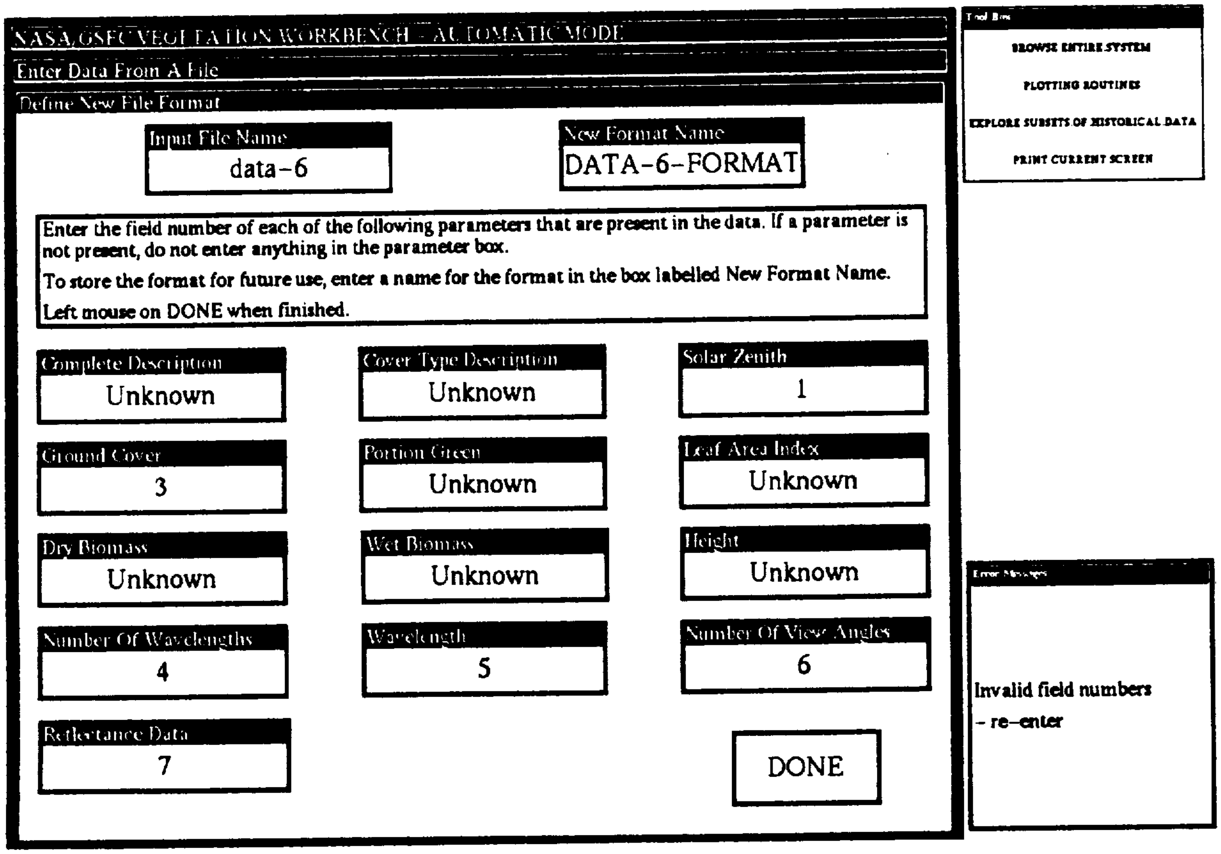


Figure 3-4

Attempting to Define a Field with a Missing Field Number

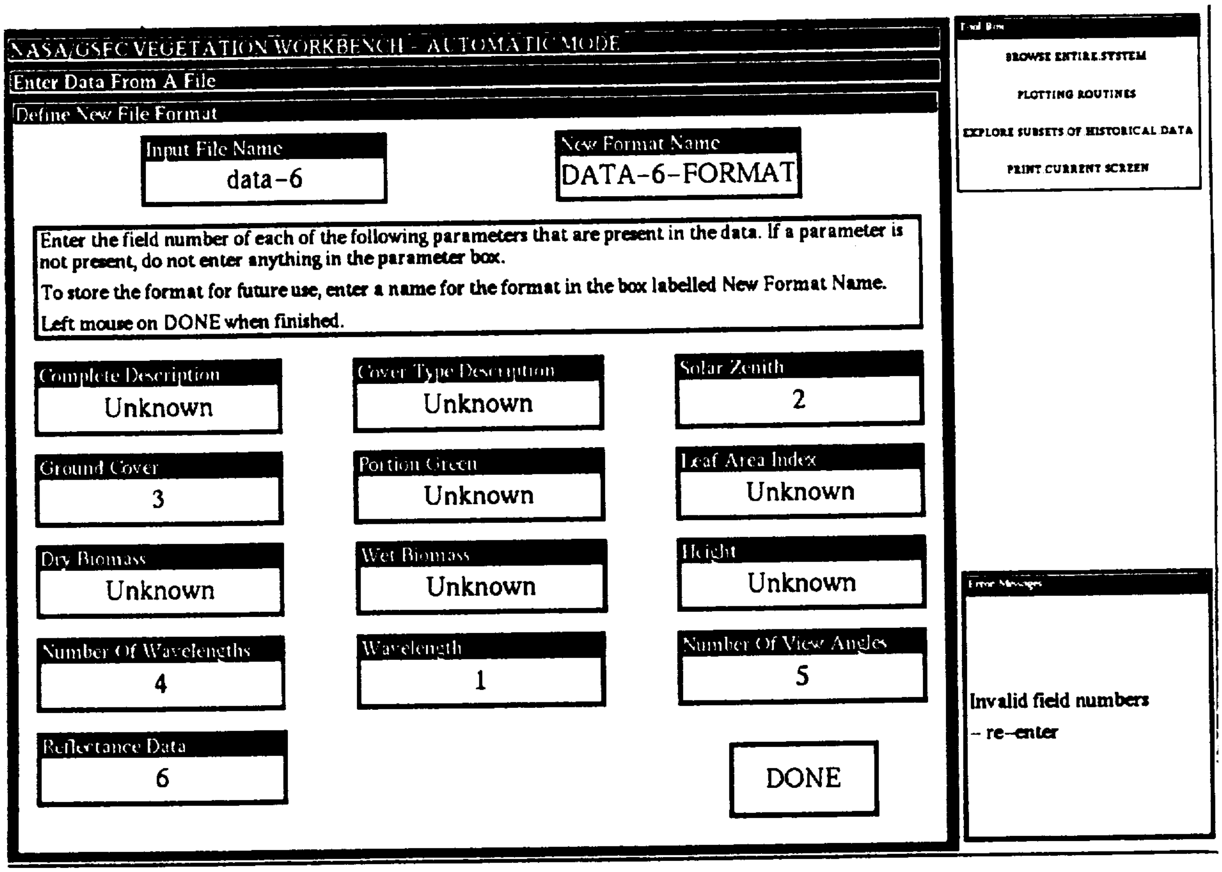


Figure 3-5

Attempting to Define a Format with the Wavelength Field Before the Solar Zenith Field

Finally, the correct field numbers for the file data-6, and the file format name data-6-format were entered. The option DONE was selected. The method for estimating the spectral hemispherical reflectance was activated once again. The data was read from the file, stored in the system and processed correctly. This test demonstrated that the option DEFINE.NEW.FORMAT was operating correctly.

3.7 TEST 7

This test was designed to test the option PREVIOUSLY.DEFINED.FORMAT. The file data-5, and the file format option PREVIOUSLY.DEFINED.FORMAT were selected. The file format name DATA-5-FORMAT was entered in response to the prompt as shown in Figure 3-6. When the method for estimating the spectral hemispherical reflectance was activated using this file and file format, the data was stored and processed correctly.

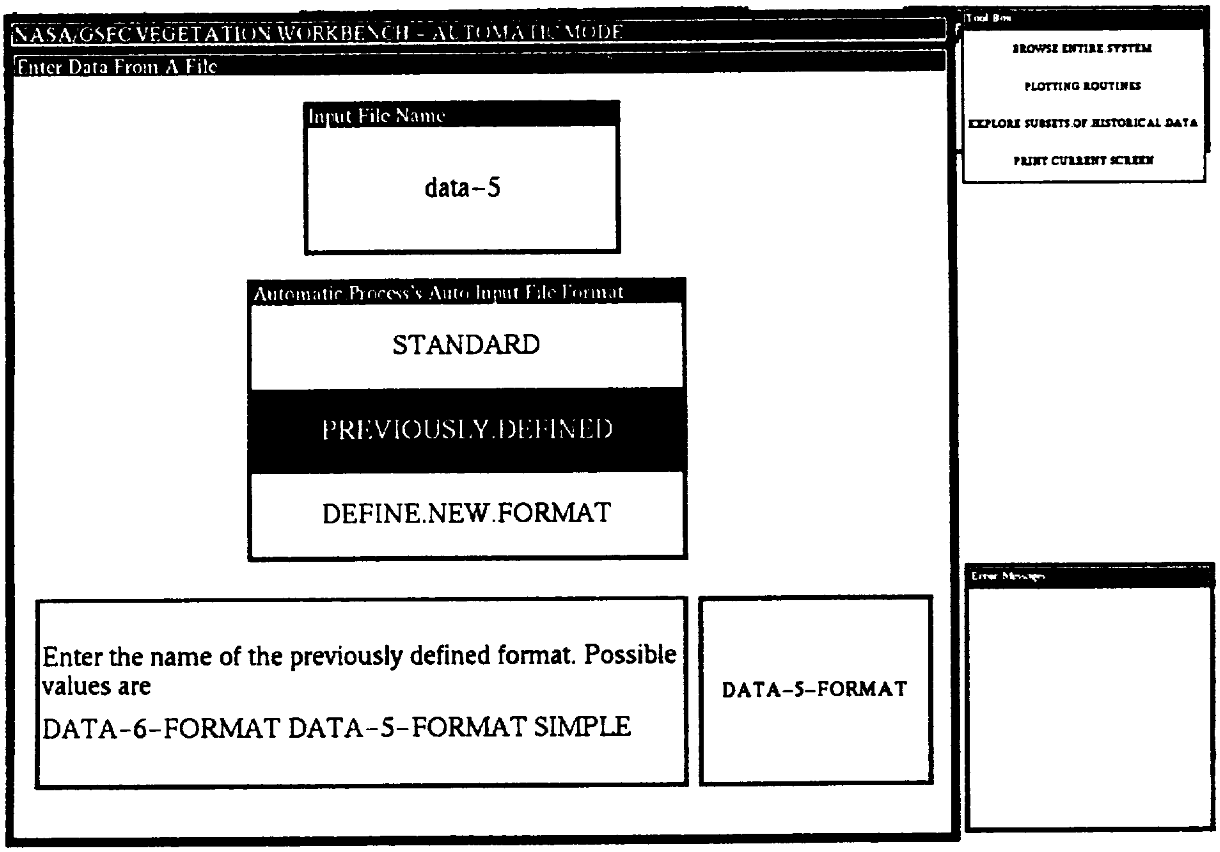


Figure 3-6

Selecting the Previously Defined Format DATA-5-FORMAT

The tester then exited the VEG "Automatic Process" main menu. Because a new file format had been defined, the values in the slot PREVIOUSLY.DEFINED.FORMATS of the unit AUTOMATIC.PROCESS were written to the file format-definitions. Inspection of the file confirmed that the format definitions had been correctly saved. The user then exited KEE.

3.8 TEST 8

The purpose of this test was to test whether a previously defined format definition could be correctly retrieved from the file of format definitions. This test also included a test to confirm that the interface would respond correctly if the user selected a previously defined format that did not exist.

KEE and VEG were loaded again. The VEG "Automatic Mode" was selected. The input data file data-6 was specified and the file format option PREVIOUSLY.DEFINED.FORMAT was selected. The user selected the format NONEXISTENT. This format did not exist so the user was prompted to re-enter the format. The format name DATA-6-FORMAT was then entered. The options SPECTRAL.HEMISPHERICAL.REFLECTANCE and GO were selected. The correct units were created and the data was read and processed correctly. This test confirmed that the option PREVIOUSLY.DEFINED.FORMAT was working correctly.

3.9 TEST 9

The purpose of this test was to determine how the system would react if the wrong format definition was given for a file. The file data-6 and the file format STANDARD were selected. This was the wrong format for the file data-6. An attempt was made to apply the VEG goal SPECTRAL.HEMISPHERICAL.REFLECTANCE to this data. As expected, VEG was not able to process this data. When VEG attempted to read the value 0.3 into the slot SOLAR.ZENITH, an error was detected because the value for the solar zenith must be an integer. The processing of the file was aborted and the system was re-initialized. An error message was displayed as shown in Figure 3-7. This test was successful in that, as expected, VEG aborted the processing of data from a file when the wrong format was specified. The result of this test highlighted the importance of the user selecting the correct format for an input file.

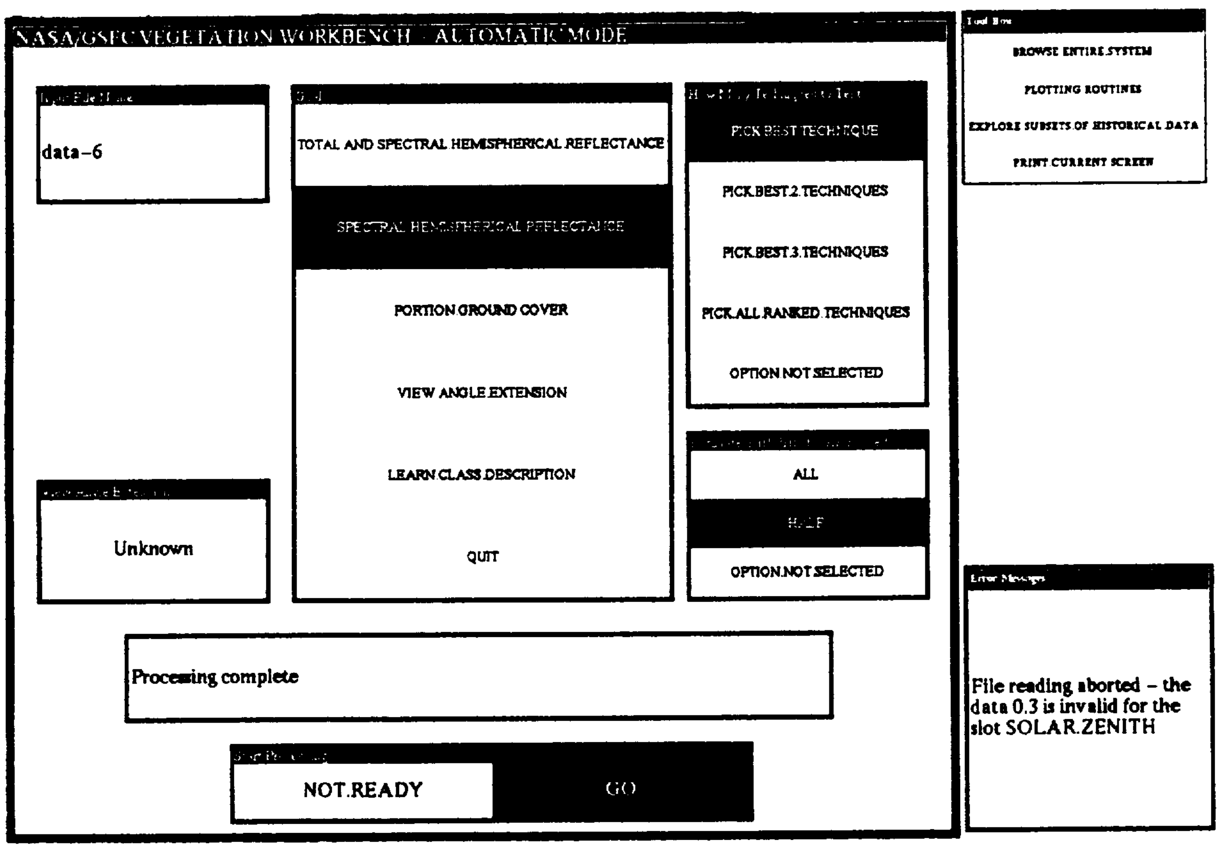


Figure 3-7

Attempting to Process a File Using the Wrong Format

SECTION 4.0

CONCLUSIONS

The interface to a file of unknown cover type data was successfully implemented. The interface allows the user to specify a format for the file. This can be a standard, previously defined or newly defined format. Using this format, the file is read and the cover type data is stored ready for processing in the system.

Using data from six test files, all options were tested. The testing included tests using both valid and invalid data. All the tests were successful, showing that the interface was working correctly. It should be noted that ultimately, system performance is limited by the performance of the Lisp interpreter and the memory management facilities the Lisp system provides.

APPENDIX A

LISP CODE FOR THE INTERFACE TO ENTER DATA FROM A FILE OF UNKNOWN COVER TYPE DATA

```
;;; veg-methods1.lisp
;;;
;;; Code for Task A of the NASA VEG project, 1992
;;;
;;; Written by PAH
;;; Created 6th August 1992
;;; Last Modified 1st September 1992

(in-package 'kee)

;;;------------------------------------------------------------------------
;;;Methods to Enter Data From a File of Possibly Previously Unknown Format
;;;------------------------------------------------------------------------

(defvar *standard-sample-format* '(cover.type.description solar.zenith
leaf.area.index ground.cover portion.green dry.biomass wet.biomass height
number.wavelengths) "The slots from which the sample data should be read, in
the correct order, in the standard format")

(defvar *standard-wavelength-format* '(wavelength number.view.angles
reflectance.data) "The slots from which the wavelength data should be read, in
the correct order, in the standard format")

(defun input-data-from-file (file)
"Controls the input of sample data from a file."
  (with-open-file (str file :direction :input)
    (store-sample-data str
                        (get.value 'automatic.process
                           'auto.input.sample.format)
                        (get.value 'automatic.process
                           'auto.input.wavelength.format))))
```

```
(defun store-sample-data (str sample-format-list wavelength-format-list)
"Stores the data for any number of samples."
 (do ((first-slot (read-file str)(read-file str)))
    ((null first-slot)(format () "End of data file"))
   (let ((new-unit (create.unit
                     (gentemp "FILE-SAMPLE-")
                     'veg 'target.data ())))
    (add.value 'estimate.hemispherical.reflectance 'new.samples new-unit)
    (put.value new-unit (first sample-format-list) first-slot)
    (dolist (slot (rest sample-format-list))
       (let ((data (read-file str)))
        (cond ((null data)
               (abort-reading 'eof)) ;End of file in wrong place
              ((eq data 'unknown) nil)  ;Data unknown - do nothing
              ((eq slot 'number.wavelengths)
               (dotimes (n data)
                 (store-wavelength-data str new-unit
                                        wavelength-format-list)))
              (t (if (valid-data slot data)
                     (put.value new-unit slot data)
                     (abort-reading data slot))))))
    ;If number of wavelengths not specified, assume one wavelength per
    ;sample and process the sample data accordingly
    (when (and wavelength-format-list
               (not (member 'number.wavelengths sample-format-list)))
       (store-wavelength-data str new-unit wavelength-format-list)))))

(defun store-wavelength-data (str parent wavelength-format-list)
"Stores the data for one wavelength."
 (let ((new-unit (create.unit (gentemp "W") 'veg () parent))
       (number-view-angles 0))
  (dolist (slot wavelength-format-list)
   (let ((data (read-file str)))
       (cond ((null data)
              (abort-reading 'eof)) ;End of file in wrong place - abort
            ((eq data 'unknown) nil)  ;Data unknown - do nothing
            ((eq slot 'number.view.angles)
             (if (and (integerp data)
                      (>= data 0)
                      (<= data 100))
                 (setf number-view-angles data)
                 (abort-reading data slot)))
            ((eq slot 'reflectance.data)
             (sort-out-reflectance-data str new-unit number-view-angles
                                        data))
            (t (if (and (numberp data)
                        (>= data 0)
                        (<= data 1.2))
                   (put.value new-unit slot data)
                   (abort-reading data slot))))))))
```

```
(defun sort-out-reflectance-data (str unit num-view-angles data)
"Deduce the format of the reflectance data, construct a list of lists if
necessary and then store it in the reflectance data slot."
  (cond ((and (consp data)        ;Data already is list of lists
              (consp (first data)))
         (if (valid-reflectance-data data)
             (put.value unit 'reflectance.data data)
             (abort-reading data 'reflectance.data)))
        ((consp data)             ;Data is list for one point
         (let ((reflectance-data (list data)))
           (dotimes (ang (- num-view-angles 1))
             (let ((next-point (read-file str)))
                  (if (null next-point)
                      (abort-reading 'eof)
                      (push next-point reflectance-data))))
           (put.value unit 'reflectance.data reflectance-data)))
        (t (let* ((azimuth (read-file str))
                  (reflectance (read-file str))
                  (reflectance-data ;Separate values in data
                   (if (or (null azimuth)(null reflectance))
                       (abort-reading 'eof)
                       `((,data ,azimuth ,reflectance)))))
             (dotimes (ang (- num-view-angles 1))
                  (let ((zenith (read-file str))
                        (azimuth (read-file str))
                        (reflectance (read-file str)))
                    (if (or (null zenith)(null azimuth)(null reflectance))
                        (abort-reading 'eof)
                        (push `(,zenith ,azimuth ,reflectance)
                              reflectance-data))))
             (put.value unit 'reflectance.data reflectance-data)))))

(defun valid-data (slot data)
"Returns t if the data is valid for the slot and nil otherwise."
  (case slot
    (complete.description (stringp data))
    (cover.type.description (stringp data))
    (leaf.area.index (and (numberp data)
                          (>= data 0)
                          (<= data 10)))
    (height (and (numberp data)
                 (>= data 0)
                 (<= data 3000)))
    (number.wavelengths (and (integerp data)
                             (>= data 0)
                             (<= data 10)))
    (solar.zenith (and (integerp data)
                       (>= data 0)
                       (<= data 90)))
    (t (and (>= data 0)
            (<= data 1)))))
```

```
(defun abort-reading (data &optional slot)
"Displays an error message and aborts the reading of the file if invalid data
is encountered or the end of file is encountered in the wrong place."
  (put.value 'methods 'general.message
     (if (eq data 'eof)
         "File reading aborted - end of file encountered prematurely"
         (format ()
           "File reading aborted - the data ~S is invalid for the slot ~S"
           data slot)))
  (dolist (sam (get.values 'estimate.hemispherical.reflectance 'new.samples))
   (dolist (w (unit.children sam 'member))
    (delete.unit w))
   (delete.unit sam))
  (initialize-auto-system)
  (unitmsg 'initialize.system 'initialize.system)
  (throw 'invalid-data nil))

(defun read-file (str)
"Reads the file. Returns the object read, or nil if the end of the file has
been reached."
  (flet ((eof-p (obj)
          "Returns t if the end of the file has been reached and nil
        otherwise"
          (eq obj '*eof*)))
   (let ((obj (read str () '*eof* ())))
    (if (eof-p obj)
        nil
        obj))))

(defun define-new-format ()
"Defines a new format for an input file by setting up the appropriate lists of
slots in the slots auto.input.sample.format and auto.input.wavelength.format of
the unit automatic.process."
  (remove.all.values 'automatic.process 'input.done)
  (remove.all.values 'automatic.process 'complete.description)
  (remove.all.values 'automatic.process 'cover.type.description)
  (remove.all.values 'automatic.process 'dry.biomass)
  (remove.all.values 'automatic.process 'ground.cover)
  (remove.all.values 'automatic.process 'height)
  (remove.all.values 'automatic.process 'leaf.area.index)
  (remove.all.values 'automatic.process 'number.view.angles)
  (remove.all.values 'automatic.process 'number.wavelengths)
  (remove.all.values 'automatic.process 'portion.green)
  (remove.all.values 'automatic.process 'reflectance.data)
  (remove.all.values 'automatic.process 'solar.zenith)
  (remove.all.values 'automatic.process 'wavelength)
  (remove.all.values 'automatic.process 'wet.biomass)
  (remove.all.values 'automatic.process 'new.file.format.name)
  (unitmsg 'viewport-automatic.process.3 'open-panel!)
  (put.value 'automatic.process 'message
    "Enter the field number of each of the following parameters that are present in the data. If a
parameter is not present, do not enter anything in the parameter box. To store the format for future
use, enter a name for the format in the box labelled New Format Name. Left mouse on DONE
when finished."))
```

```
(defun finish-format-definition ()
"Call the function to construct the list of sample and wavelength slots from
the input field numbers."
 (let ((possible-sample-slots
        '(complete.description cover.type.description
          dry.biomass ground.cover
          height leaf.area.index
          number.wavelengths portion.green
          solar.zenith wet.biomass))
       (possible-wavelength-slots
        '(wavelength number.view.angles reflectance.data)))
  (remove.all.values 'automatic.process 'auto.input.sample.format)
  (cond ((and
          (sufficient-fields)
          (process-slots possible-sample-slots 'auto.input.sample.format)
          (process-slots possible-wavelength-slots
                         'auto.input.wavelength.format))
         (let ((new-name
                (get.value 'automatic.process 'new.file.format.name)))
          (when new-name
            (add.value 'automatic.process 'previously.defined.formats
                 `(,new-name ,(get.value 'automatic.process
                                'auto.input.sample.format)
                  ,(get.value 'automatic.process
                    'auto.input.wavelength.format)))
            (put.value 'automatic.process 'new.format.defined.p t)))
         (put.value 'automatic.process 'message "")
         (clear-prompt)
         (unitmsg 'viewport-automatic.process.2 'close-panel!)
         (unitmsg 'viewport-automatic.process.3 'close-panel!))
        (t (put.value 'methods 'general.message
               "Invalid field numbers  - re-enter")))))

(defun process-slots (possible-slots slot-for-storage)
"Construct the list of slots and store it in the appropriate slot of the unit
automatic.process."
 (let ((present-slots ()))
  (dolist (slot possible-slots)
   (let ((slot-value (get.value 'automatic.process slot)))
      (when slot-value
        (push `(,slot ,slot-value) present-slots))))
  (setf present-slots (sort present-slots #'< :key #'second))
  (cond ((valid-field-numbers present-slots)
         (put.value 'automatic.process slot-for-storage
            (mapcar #'first present-slots))
         t)
        (t nil))))

(defun sufficient-fields ()
"Checks to see that the minimum fields are included. These are the solar
zenith, wavelength and reflectance data fields."
 (and (get.value 'automatic.process 'solar.zenith)
      (get.value 'automatic.process 'wavelength)
      (get.value 'automatic.process 'reflectance.data)))
```

```
(defun valid-field-numbers (present-slots)
"Examines the list of slots and field numbers. Checks that the field numbers
are consecutive, with the sample level fields before the wavelength level
fields."
  (let ((first-field (second (first present-slots))))
   (and (or (= first-field 1)
            (= first-field (1+
                            (length (get.value 'automatic.process
                                              'auto.input.sample.format)))))
          (dotimes (n (length present-slots) t)
           (when (/= (second (nth n present-slots))
                     (+ first-field n))
             (return-from valid-field-numbers nil))))))

;;;----------------------------------------------------------------------
;;; Methods to read in previously defined formats from a file and output them
;;; to a file
;;;----------------------------------------------------------------------

(defun read-defined-formats ()
 (when (probe-file "format-definitions")
  (with-open-file (str "format-definitions" :direction :input)
    (put.values 'automatic.process 'previously.defined.formats
         (read-file str)))
   (put.value 'automatic.process 'new.format.defined.p nil)))

(defun rewrite-defined-formats ()
 (with-open-file (str "format-definitions" :direction :output)
   (princ (get.values 'automatic.process 'previously.defined.formats)
          str))
  (put.value 'automatic.process 'new.format.defined.p nil))
```

APPENDIX B

LISTINGS OF FILES USED FOR TESTING THE INTERFACE

LISTINGS OF FILES USED FOR TESTING THE INTERFACE

Listing of the File data-1

```
"Example of dense vegetation canopy"
45 unknown
unknown unknown
unknown unknown unknown
1
0.68
11
((0 0 0.043)(15 180 0.043)(15 0 0.043)(30 180 0.054)(30 0 0.043)(45 180 0.066)(45 0
0.044)(60 180 0.076)(60 0 0.054)(75 180 0.089)(75 0 0.067))
```

Listing of the File data-2

```
"Example of dense vegetation canopy"
45 unknown
unknown unknown
unknown unknown unknown
2
0.68
11
((0 0 0.043)(15 180 0.043)(15 0 0.043)(30 180 0.054)(30 0 0.043)(45 180 0.066)(45 0
0.044)(60 180 0.076)(60 0 0.054)(75 180 0.089)(75 0 0.067))
0.92
1
((0 0 0.5))
"Example of dense vegetation canopy"
45 unknown unknown unknown
unknown unknown unknown
2
0.68
1
((0 0 0.043))
0.92
1
((0 0 0.5))
```

Listing of the File data-3

```
"Example of dense vegetation canopy"
45 3.5
0.4 unknown
unknown unknown unknown
2
0.68
1
(0 0 0.43)
0.92
1
(0 0 0.5)
"Dense forest"
65 9.9 0.9 0.99
unknown unknown 3000
3
0.44
1
(15 45 0.2)
0.65
1
(0 0 0.5)
0.92
7
(0 0 0.6) (15 14 0.65) (30 16 0.57) (45 12 0.54) (60 13 0.62) (75 17 0.58) (90 14 0.63)
```

Listing of the File data-4

```
"Example of dense vegetation canopy"
45 3.5
0.4 unknown
unknown unknown unknown
2
0.68
1
0 0 0.43
0.92
1
0 0 0.5
"Dense forest"
65 9.9 0.9 0.99
unknown unknown 3000
3
0.44
1
15 45 0.2
0.65
1
0 0 0.5
0.92
7
0 0 0.6 15 14 0.65 30 16 0.57 45 12 0.54 60 13 0.62 75 17 0.58 90 14 0.63
```

Listing of the File data-5

```
45 0.3
2
0.68
1
0 0 0.43
0.92
1
0 0 0.5
65 0.71
3
0.44
1
15 45 0.2
0.65
1
0 0 0.5
0.92
7
0 0 0.6 15 14 0.65 30 16 0.57 45 12 0.54 60 13 0.62 75 17 0.58 90 14 0.63
```

Listing of the file data-6

```
45 0.3 0.5 0.6
2
0.68
((0 0 0.43))
0.92
((0 0 0.5))
65 0.71 0.7 0.8
3
0.44
((15 45 0.2))
0.65
((0 0 0.5))
0.92
((0 0 0.6) (15 14 0.65) (30 16 0.57) (45 12 0.54) (60 13 0.62) (75 17 0.58) (90 14 0.63))
```

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle An Expert System Shell for Inferring Vegetation Characteristics - Interface to a File of Unknown Cover Type Data (Task A)		5. Report Date September 1992 6. Performing Organization Code
7. Author(s) P. Ann Harrison		8. Performing Organization Report No. B921015-U-2R01 10. Work Unit No. 462-61-14
9. Performing Organization Name and Address JJM Systems, Inc. One Ivybrook Blvd., Suite 190 Ivyland, PA 18974		11. Contract or Grant No. NAS5-30127 13. Type of Report and Period Covered Task Report for Task A August-September 1992
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration, Washington, D.C. 20546-0001 NASA/Goddard Space Flight Center, Greenbelt, MD 20771		14. Sponsoring Agency Code

15. Supplementary Notes

The Lisp and KEE code for this work is available on a Sun Cartridge Tape.

16. Abstract

VEG is an expert system that infers vegetation characteristics from reflectance data. Various scientists have conducted experiments and produced data files of unknown cover type data on which VEG could operate. In order to enter this data into VEG an interface is required. This interface has been implemented using KEE and Common Lisp. The interface allows the user to specify a format for the file. This can be a standard, previously defined format or newly defined format. The system uses the specified format to read data from a file into VEG. The interface has been tested using both valid and invalid data. The tests confirmed that the interface was operating correctly.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
EXPERT SYSTEM, ARTIFICIAL INTELLIGENCE, REMOTE SENSING	UNCLASSIFIED - UNLIMITED

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	34	

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171

APPENDIX B

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - OUTPUT OF RESULTS TO A FILE (TASK B)

JJM
SYSTEMS INC

B921016-U-2R02

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - OUTPUT OF RESULTS TO A FILE (TASK B)

11 September 1992

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771

Prepared by:

JJM Systems, Inc.
One Ivybrook Boulevard, Suite 190
Ivyland, PA 18974

SYSTEMS INC

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

KEE	Knowledge Engineering Environment
VEG	VEGetation Workbench

SECTION 1.0

INTRODUCTION

The previous version of the VEGetation workbench (VEG) could be operated in two different modes. In the "Research Mode" the results were displayed on the screen, whereas, in the "Automatic Mode," the results were written to a file in a simple format. The purpose of this task was to expand the options available for writing the results to a file. This would enable the results from VEG to be input via a file to other programs for plotting characteristics. It would also be the first step towards entering the results from one run of VEG back into VEG to be used as historical data for future runs.

Task B has been completed. An interface to an output file has been implemented. The interface allows the results from VEG to be written to a file when VEG is run in the "Research Mode" as well as when it is run in the "Automatic Mode." The interface to an output file is available for all the VEG goals except "LEARN.CLASS.DESCRIPTIONS" which has a radically different structure from the other subgoals and will be developed as Tasks C and D of this contract. The interface allows the user to select the output file name, the parameters to be written and the format to be used. A number of standard templates that match commonly used formats have also been provided. The interface is described in detail in this report. The code for the Lisp methods involved is included in Appendix A. A Sun cartridge tape containing these Lisp methods and the current version of VEG including the completed interface to a output file has been delivered to the NASA GSFC technical representative.

SECTION 2.0

DETAILED DESCRIPTION OF THE INTERFACE

VEG can be operated in two different modes. In the "Research Mode," the scientist must separately execute each step in the processing of unknown cover type data. This mode allows the scientist to study the intermediate results in detail. VEG can also be operated in the "Automatic Mode." In this mode, the scientist selects the operations to be carried out. The cover type data is read from a file, processed and the results are written in a simple format to another file without any further intervention from the user. The options available for writing the results to a file have been expanded. These options are available when VEG is operated in the "Research Mode" as well as in the "Automatic Mode."

2.1 SPECIFYING THE FILENAME, PARAMETERS AND FORMAT - RESEARCH MODE

When VEG is operated in the "Research Mode," the results are displayed on the screen as shown in Figure 2-1. This figure shows the results for the goal "VIEW.ANGLE.EXTENSION" and the extension of the reflectance data for SAMPLE 9 to the view angle (45 75). The user can left click on the menu options "NEXT.WAVELENGTH" and "PREVIOUS.WAVELENGTH" to view the results for the different wavelengths.

The options for outputting the results have been expanded. When the user selects "QUIT" (Figure 2-1), the screen shown in Figure 2-2 is displayed. This screen enables the user to indicate whether the results should be written to a file. If the user left clicks on "NO", the screen is cleared, and the main menu for the appropriate VEG subgoal is visible once again. If the user left clicks on "YES", the box labelled "Output File Name" is opened, and the user is prompted to enter the name of the file. Figure 2-3 shows this case. When the user has entered a file name, a check is made to see if the file already exists. If the file already exists, the user is asked whether or not the file should be overwritten, as shown in Figure 2-4. If the user left clicks on "NO", he/she is prompted to enter a new file name, as shown in Figure 2-5. When the user has either entered the name of a file that does not already exist, or consented to overwriting an existing file, the screen shown in Figures 2-2 through 2-5 is closed and the screen shown in Figure 2-6 is opened.

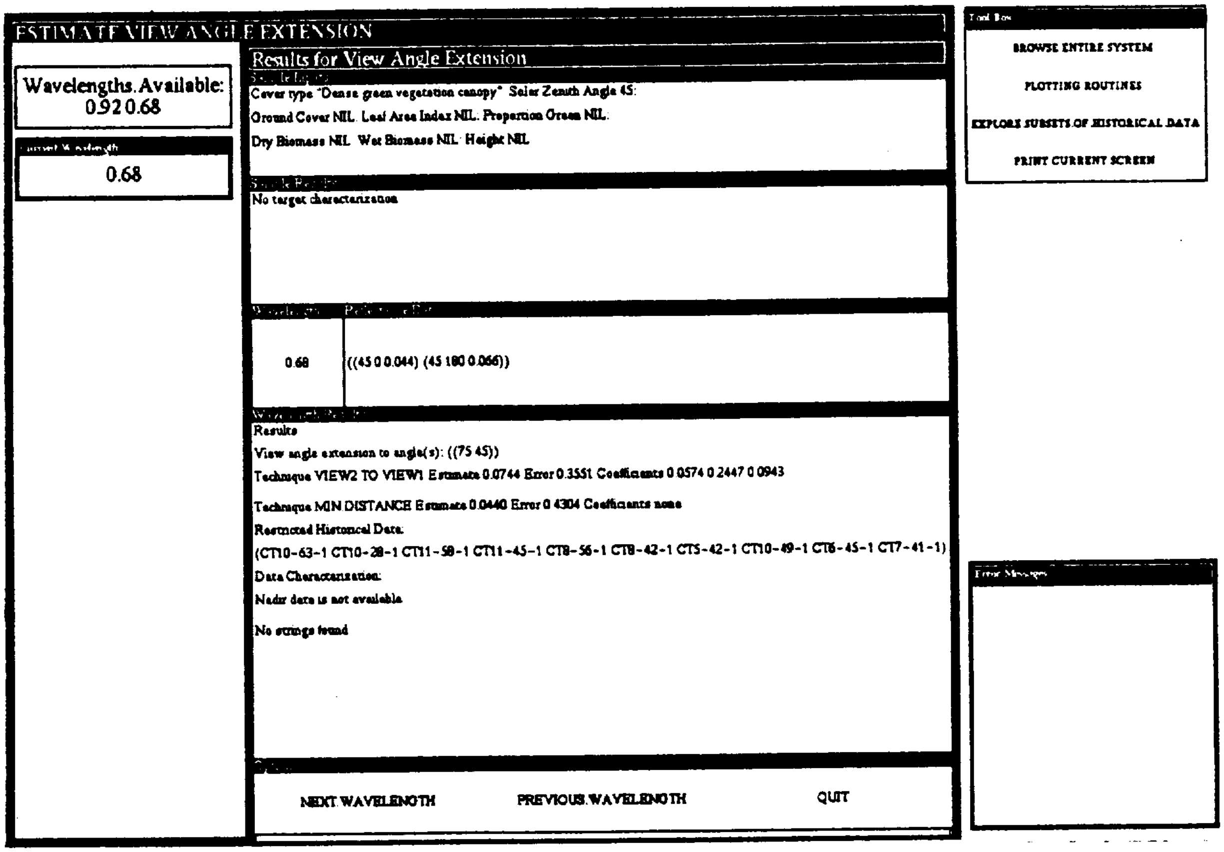


Figure 2-1

Results Displayed on the Screen in the VEG "Research Mode"

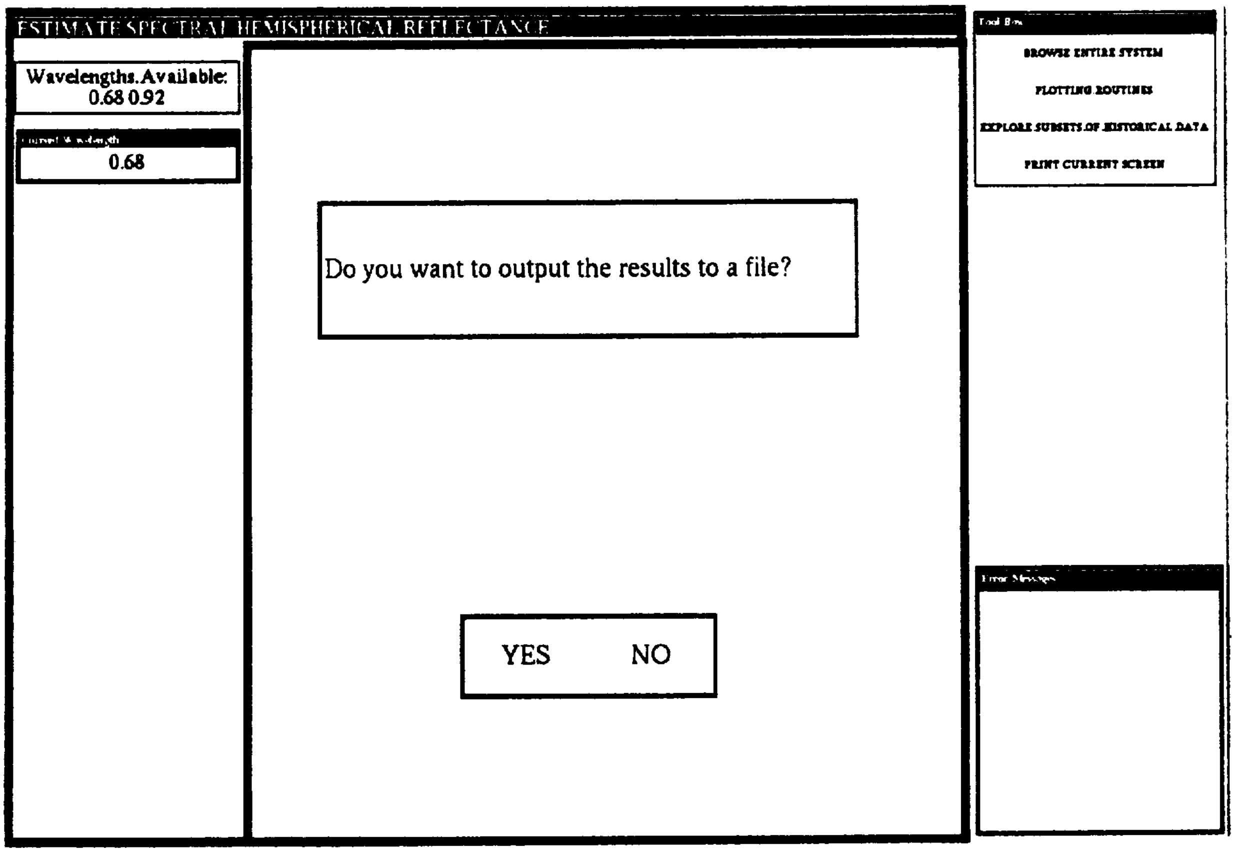


Figure 2-2

The Screen that Enables the User to Specify Whether the Results Should be Written to a File

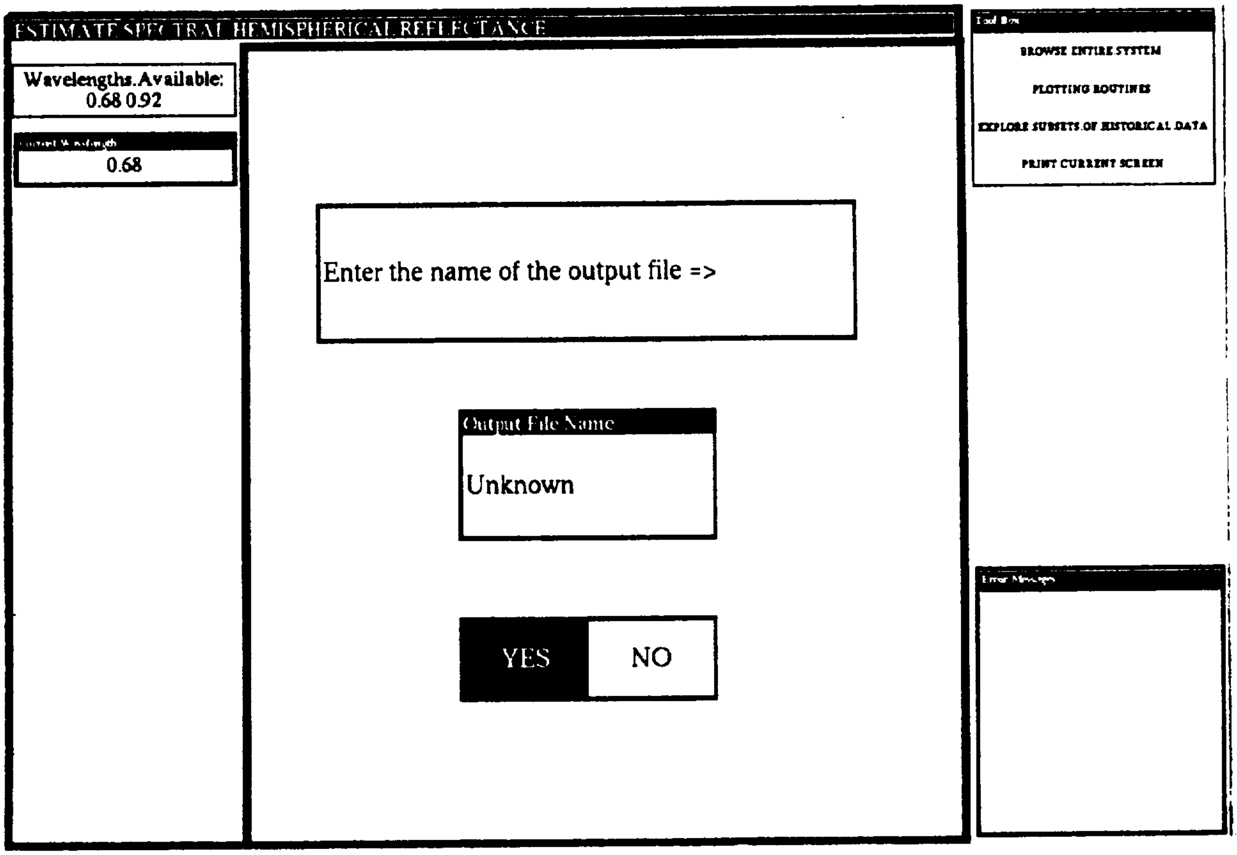


Figure 2-3

The Screen that Enables the User to Name the Output File

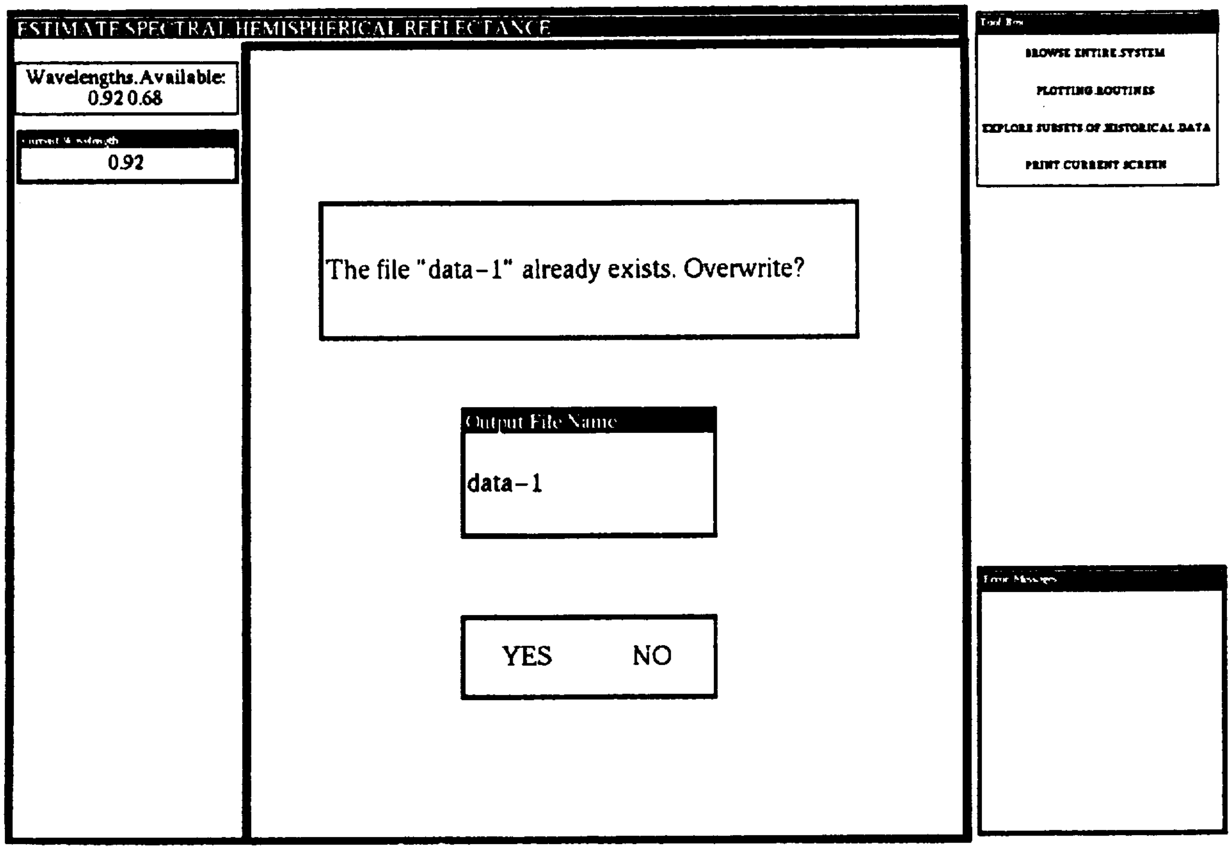


Figure 2-4

Asking Whether a File Should be Overwritten

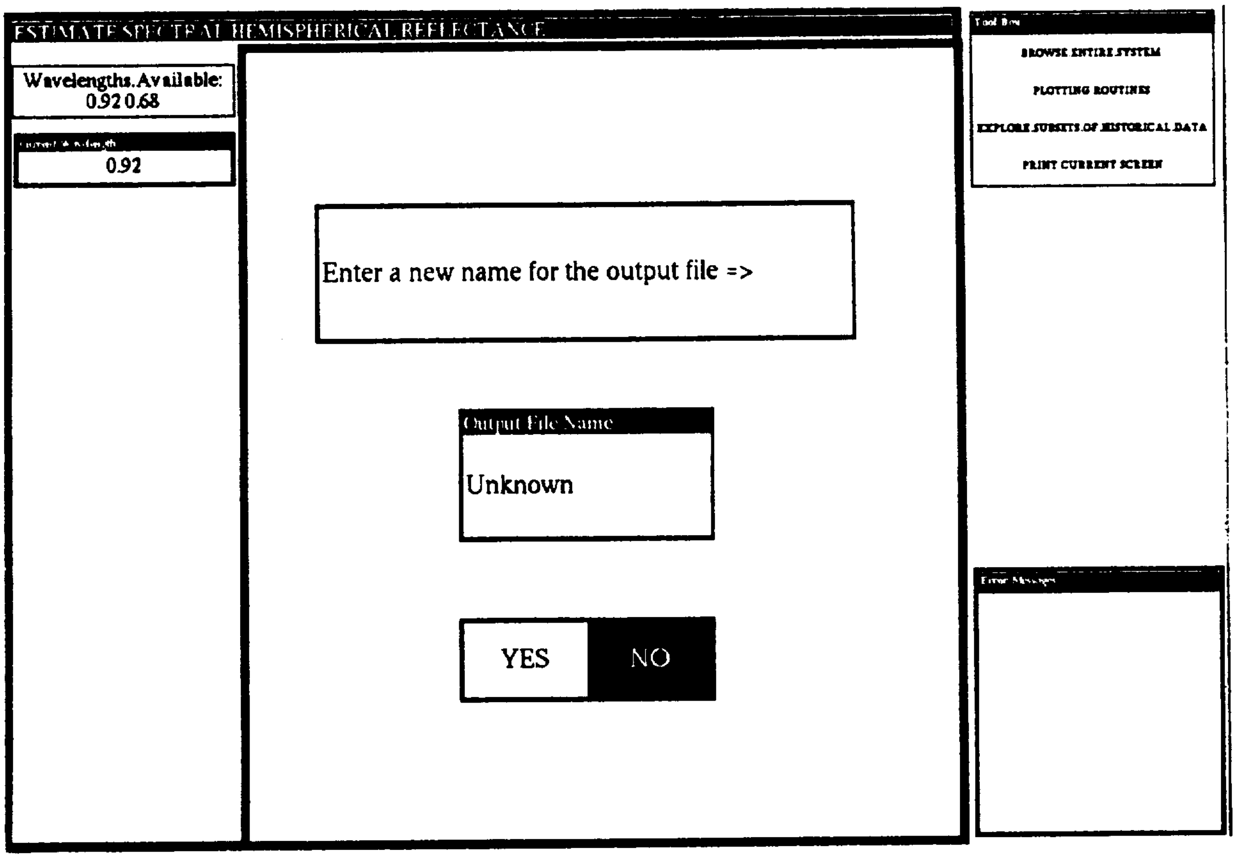


Figure 2-5

Prompting the User for a New File Name

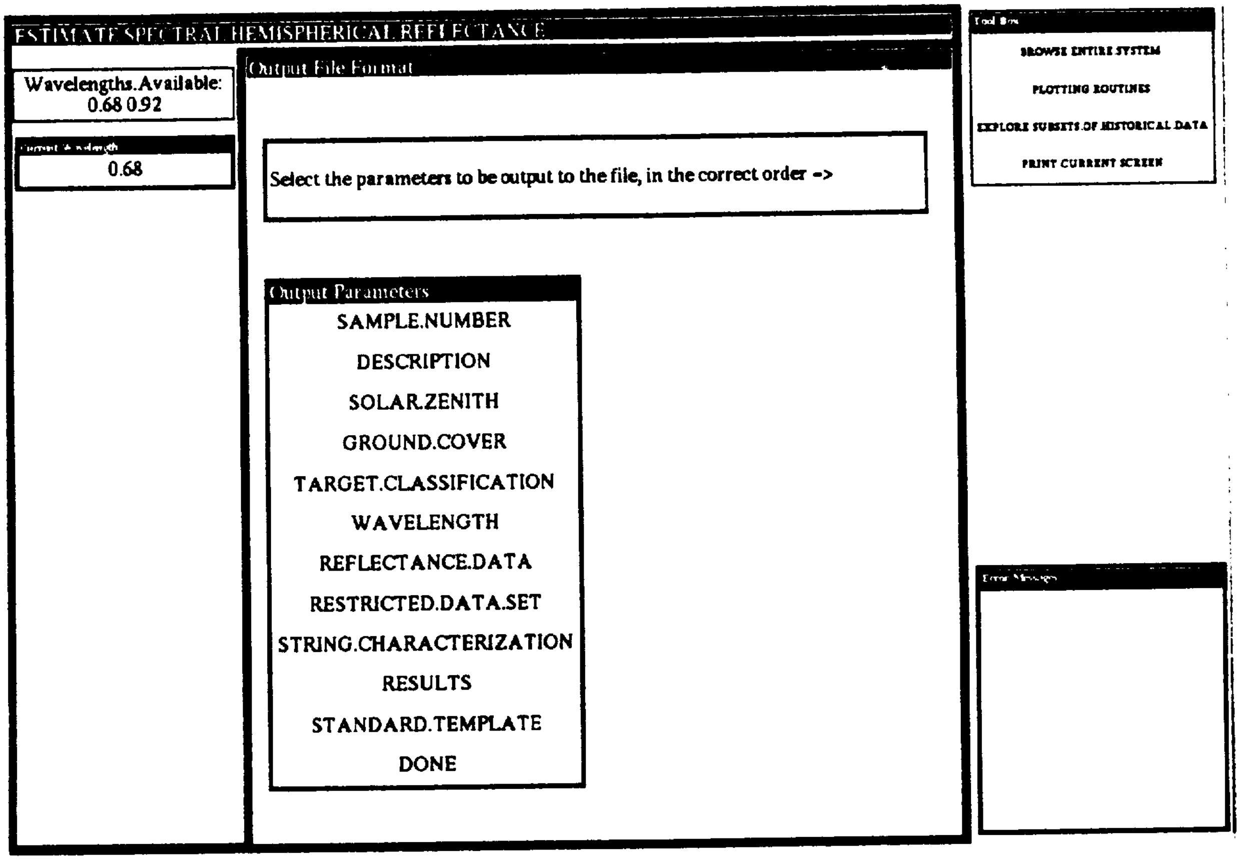


Figure 2-6

The Screen that Enables the User to Specify the File Format

The screen shown in Figure 2-6 enables the user to select the parameters and specify the format in which the data is to be written to the file. Each time the user left clicks on a parameter in the output parameters menu, the name of the parameter is added to the slot FORMAT.LIST in the unit 9.OUTPUT.

The reflectance data can be written to the file in one of several different formats. If the user selects the parameter REFLECTANCE.DATA, the prompt is changed accordingly and a submenu is opened as shown in Figure 2-7. This submenu allows the user to select the format in which the reflectance data is to be written to the file. This can be a list of lists, e.g., ((45 0 0.044)(45 180 0.066)), points as lists, e.g., (45 0 0.044)(45 180 0.066), without parentheses, e.g., 45 0 0.044 45 180 0.066 or reflectance values only, e.g., 0.044 0.066. When the reflectance data submenu opens, the left mouse button functionality of the output parameters menu is disabled. This is to prevent the user selecting another output parameter before choosing the format for the reflectance data. When the user left clicks on a reflectance data format to select it, the value of the format is added to the slot FORMAT.LIST of the unit 9.OUTPUT, the reflectance data format submenu is closed and the left button functionality of the output.parameters menu is restored.

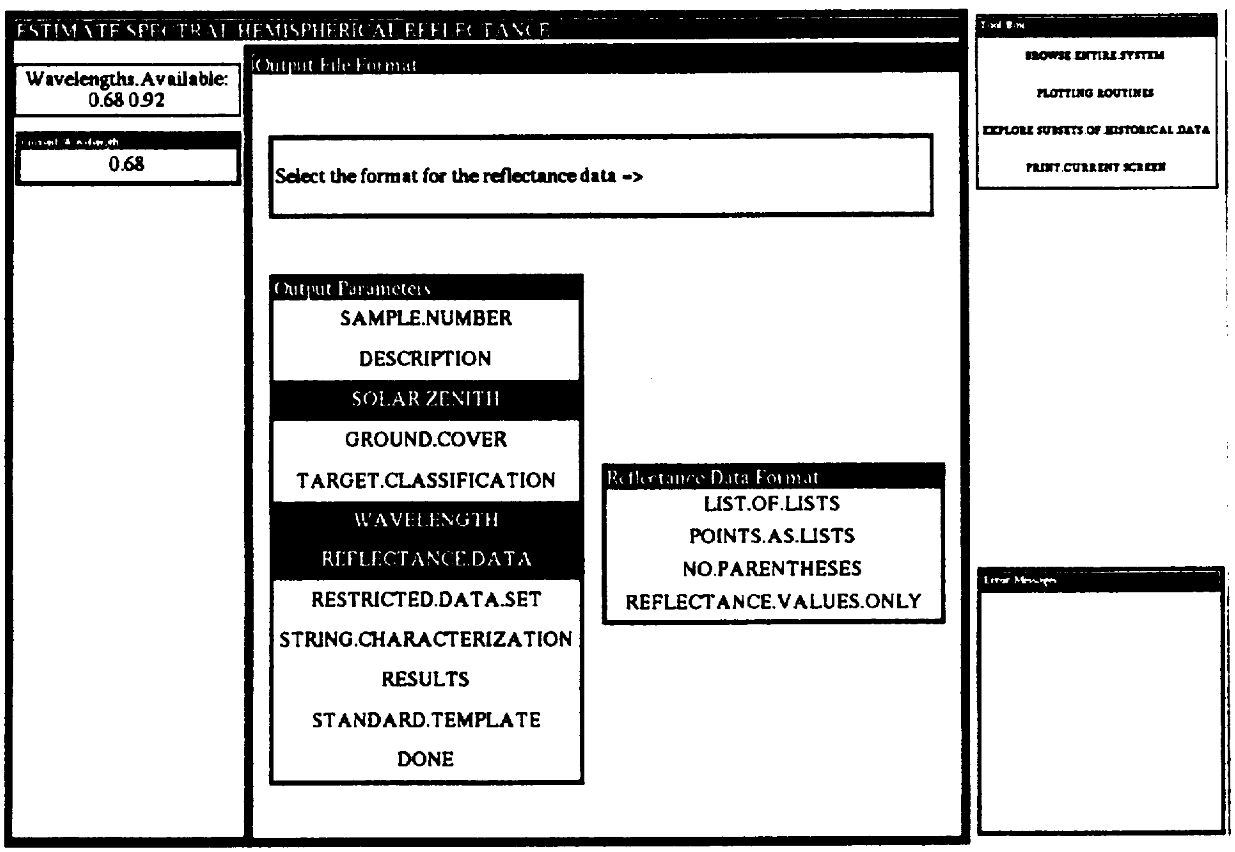


Figure 2-7

The Reflectance Data Format Submenu

If the user selects the parameter RESULTS, the results format submenu is opened as shown in Figure 2-8. This submenu allows the user to select the format for the results. For each technique the results for the goal VIEW.ANGLE.EXTENSION consist of the estimate of the reflectance value at the extended view angle, the error estimate and possibly some coefficients. Several different techniques may have been applied to the sample of unknown cover type data, and the results for each technique will be available. The user can select "ALL RESULT." In this case, the technique name, result, error estimate and coefficients for each technique are written to the file. If the user selects "RESULTS.ERROR.ONLY," the result and error estimate for each technique are written to the file. If the user selects "RESULTS.ONLY," the results for each technique is written to the file. In the case of the goal VIEW.ANGLE.EXTENSION, the result is the reflectance value at the extended view angle as calculated using each technique.

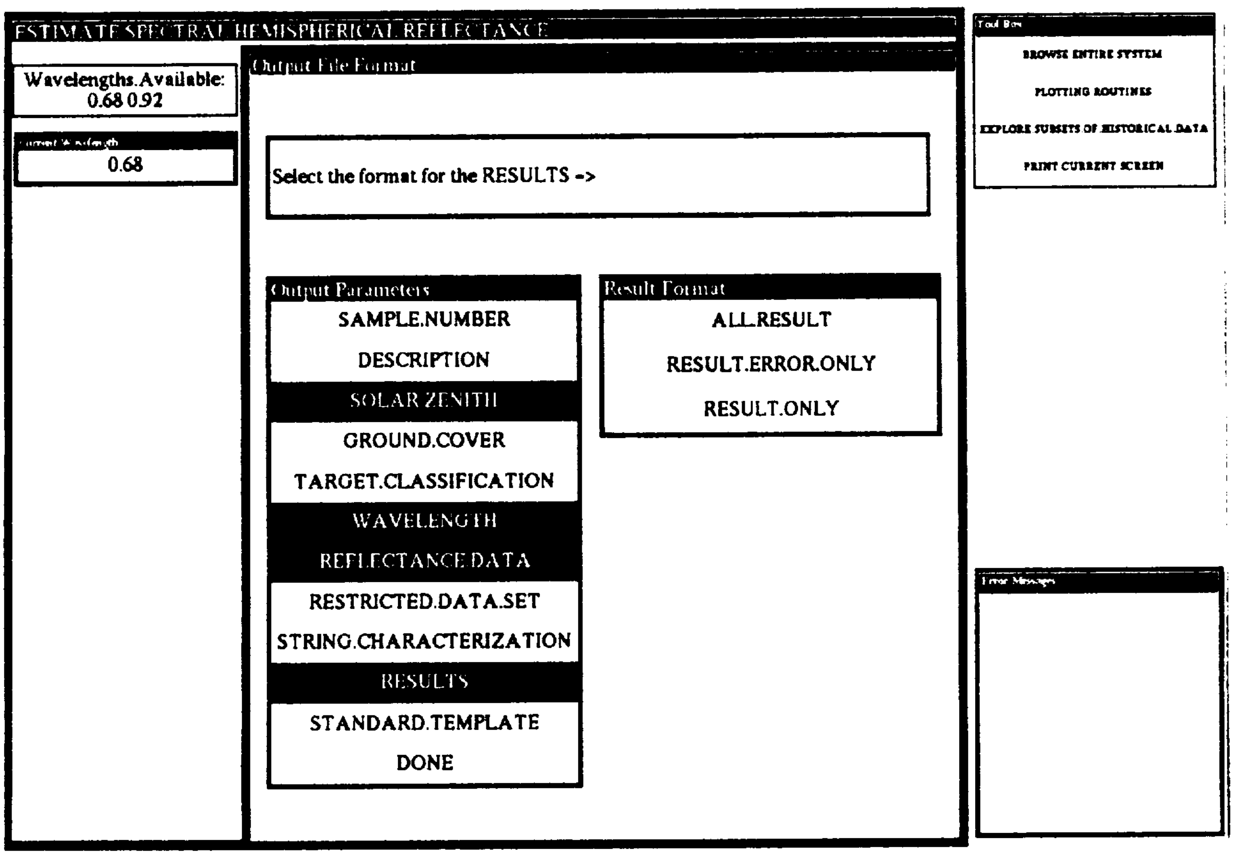


Figure 2-8

The Result Format Submenu

If the user left clicks on a parameter that has already been selected, the parameter is deselected. The parameter is removed from the slot FORMAT.LIST of the unit 9.OUTPUT. If the removed parameter is either reflectance data or results, the format for the parameter is also removed from the slot FORMAT.LIST.

A number of templates have been provided. If the user left clicks on "STANDARD.TEMPLATE" in the output parameters menu, brief descriptions of the templates are displayed and the template number submenu is opened. This submenu is shown in Figure 2-9. This submenu allows the user to select from a number of standard templates. TEMPLATE.1 is the simple format that was previously available in the VEG "Automatic Mode." Simple formats have been set up for TEMPLATE.2 and TEMPLATE.3. When the user selects a template number the screen shown in Figure 2-9 is closed. If TEMPLATE.2 or TEMPLATE.3 has been selected, a list of parameters and formats corresponding to the template is written to the slot FORMAT.LIST of the unit 9.OUTPUT, replacing my previous values in the slot. If TEMPLATE.1 has been selected, the value TEMPLATE.1 is placed in the FORMAT.LIST slot. If VEG is being run in the "Research Mode", the data is then written to the file in the specified format.

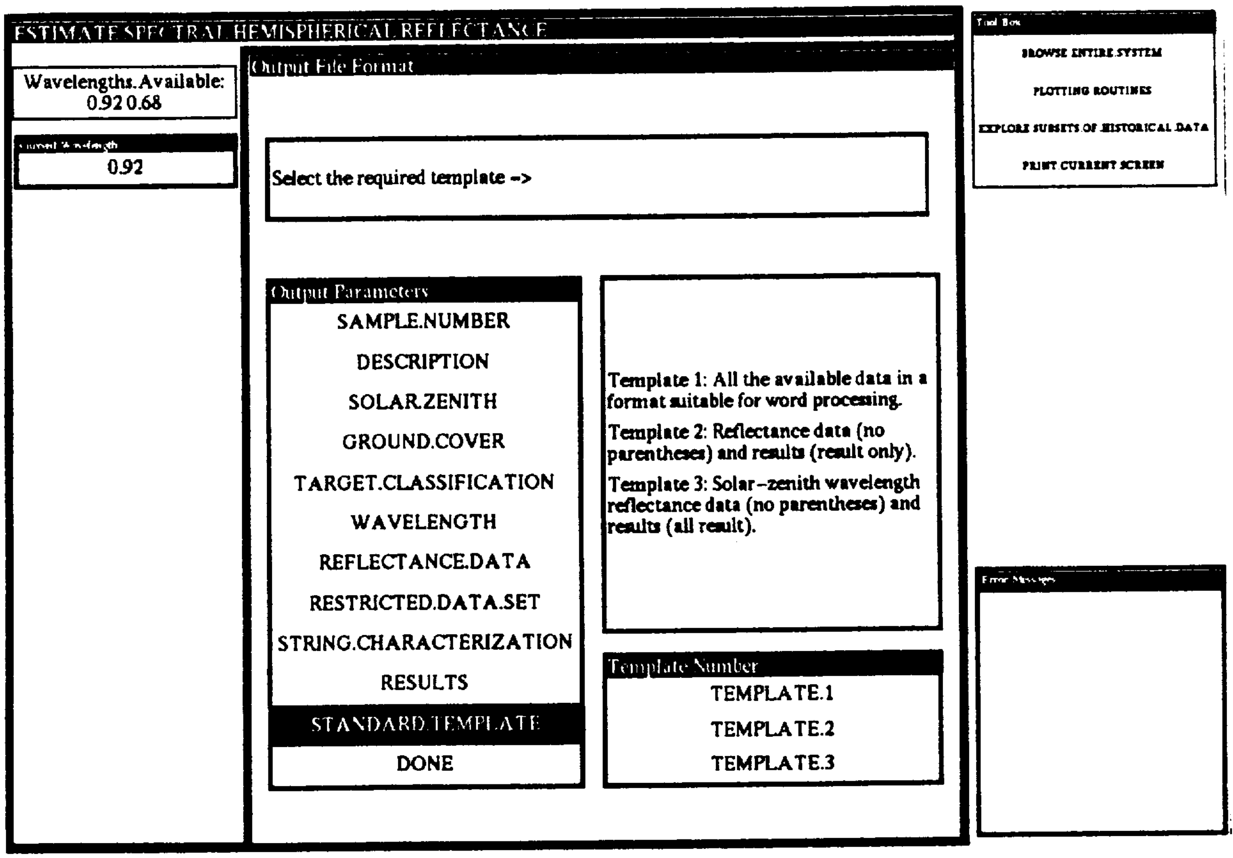


Figure 2-9

The Template Number Submenu

If the user selects the parameters and formats separately (rather than using a standard template) and then left clicks on "DONE," the output file format screen is closed. If VEG is being run in the "Research Mode," the data is then written to the file in the specified format.

2.2 SPECIFYING THE FILENAME, PARAMETERS AND FORMAT - AUTOMATIC MODE

When the user selects the "Automatic Mode," the screen shown in Figure 2-10 is displayed. This screen enables the scientist to specify how the system is to be run, including naming the input file and selecting the VEG goal such as SPECTRAL.HEMISPHERICAL.REFLECTANCE. An additional box labelled "Output File Name," has been added to this screen. When the user enters an output file name, a check is made to see if the file already exists. If the file does exist the screen shown in Figure 2-4 is opened and the user is prompted to indicate whether or not the file should be overwritten, as described in

Section 2.1 for the VEG "Research Mode." When the user has either entered the name of a file that does not already exist, or consented to overwriting an existing file, the output file format screen is opened as shown in Figure 2-11. The operation of this screen was described in Section 2.1. When the user selects a standard template or left clicks on "DONE," this screen is closed. In the "Research Mode," the output file parameters and format are specified after the data has been processed. The data is written to the file immediately after the file format has been specified. When VEG is being run in "Automatic Mode" the output file name, parameters and format are specified before the data is processed. The data for a sample is only written to the file after it has been processed.

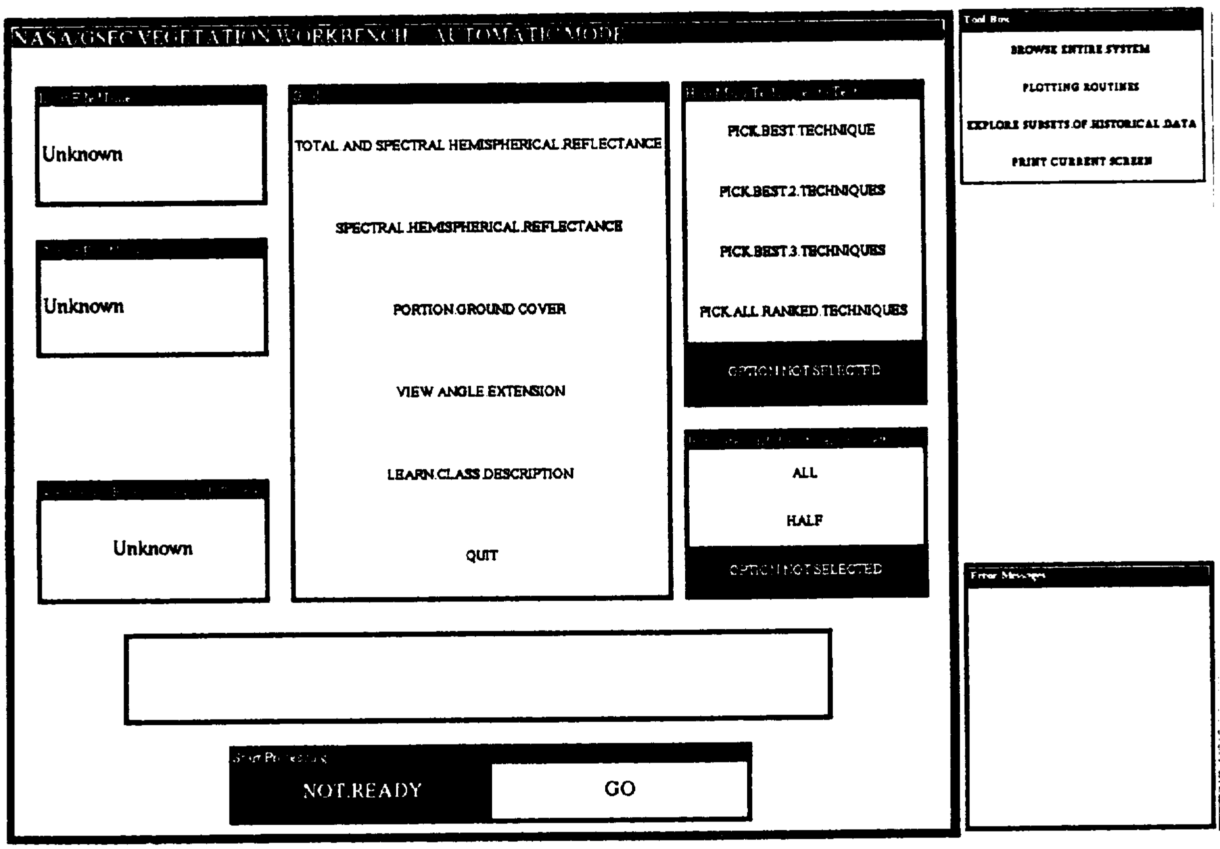


Figure 2-10

The Main Screen in the VEG "Automatic Mode" Including a Box for the Output File Name

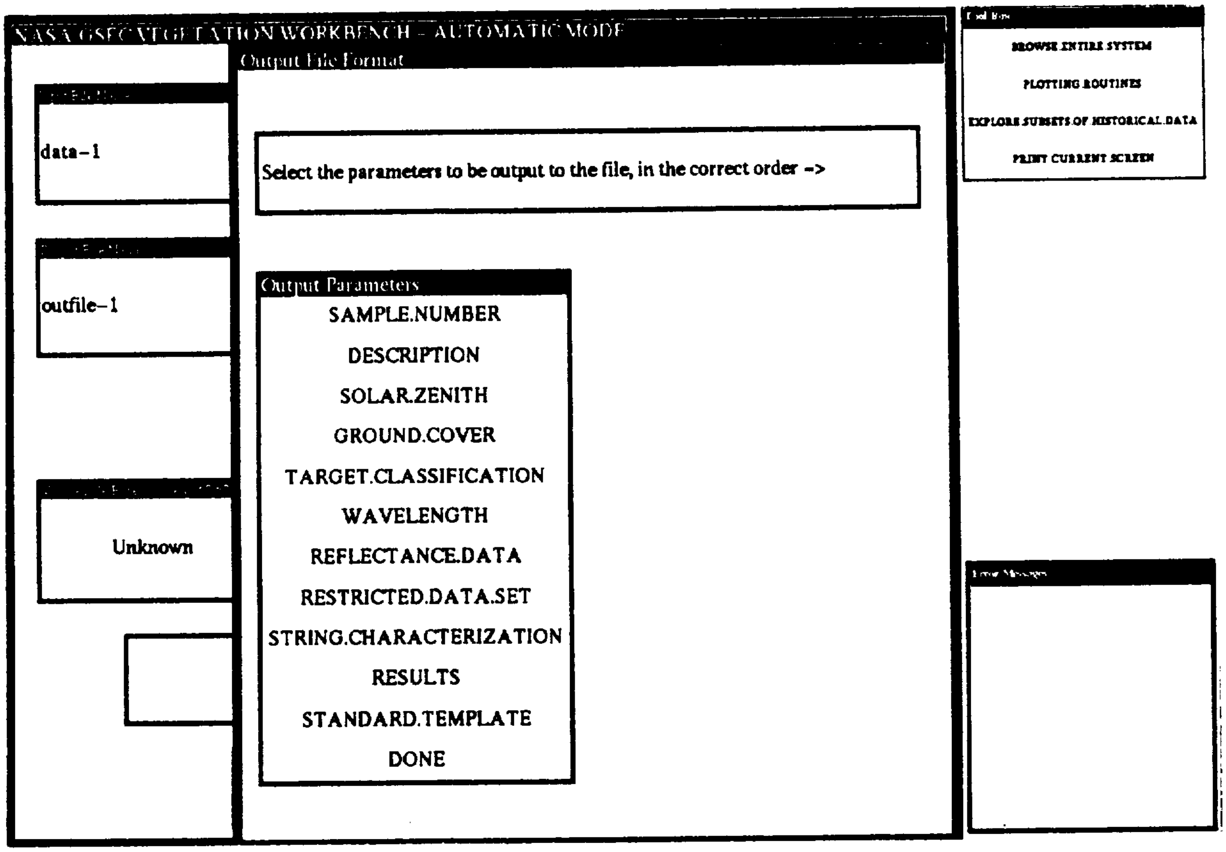


Figure 2-11

The Output File Format Screen in the VEG "Automatic Mode"

2.3 WRITING THE DATA TO THE FILE - RESEARCH MODE

If VEG is being run in the "Research Mode," the results are written to the named output file when the user either selects a template from the template submenu or selects the option "DONE" from the output parameters menu. In order to write the data to the file, the Lisp function WRITE-RESULTS-TO-FILE is called. This function copies the contents of the slot FORMAT.LIST of the unit 9.OUTPUT in reverse order into the local variable FORMAT-LIST and then opens the output file. If the first value in FORMAT-LIST is TEMPLATE.1, the function OUTPUT-DATA-TO-FILE is called and the results are written to the file in the simple format that was provided in the "Automatic Mode" in the previous version of VEG. The code that was developed previously is re-used to produce data in this simple format. If FORMAT-LIST contains anything other than TEMPLATE.1, the Lisp function WRITE-RESULTS-TO-FILE-AUX is called.

The function WRITE-RESULTS-TO-FILE-AUX writes the required parameters to the output file for each wavelength. All data is considered at the wavelength level only. For each wavelength the parameters in FORMAT-LIST are processed. If the parameter REFLECTANCE.DATA or RESULTS is found in FORMAT-LIST, the next value in FORMAT-LIST is examined to determine the format in which the data should be written to the file. If the format for the reflectance data is anything other than a list of lists, the number of view angles is written to the file before the reflectance data. The purpose of this is to facilitate the reading of data from the file into another program.

If the goal TOTAL. AND. SPECTRAL.HEMISPHERICAL.REFLECTANCE has been selected, the total hemispherical reflectance results for the sample are written to the file before the spectral hemispherical reflectance results at each wavelength.

Whitespace is written to the file after each data item to separate the data items. When all the required data has been output, the file is automatically closed.

2.4 WRITING DATA TO THE FILE - AUTOMATIC MODE

When VEG is being run in "Automatic Mode," the user selects the name and format of the input and output files before the data is processed. The user selects the goal to be used, such as SPECTRAL.HEMISPHERICAL.REFLECTANCE and then left clicks on "GO" to start the processing of the data. Before processing the data, the system checks that both an input and an output file name have been specified. If either file name is missing, processing is aborted and an error message is displayed as shown in Figure 2-12. The user must enter the missing file name(s) and then left click on "GO" again to process the data.

The data is processed a sample at a time. When the processing of a sample has been completed, the results for all wavelengths of the sample are written to the output file. The results are written to the file using the function OUTPUT-DATA-TO-FILE if TEMPLATE.1 has been specified. Otherwise the function WRITE-RESULTS-TO-FILE-AUX is used, as described in section 2.3 for the VEG "Research Mode". When all the samples have been processed and all the results have been written to the file, the file is closed.

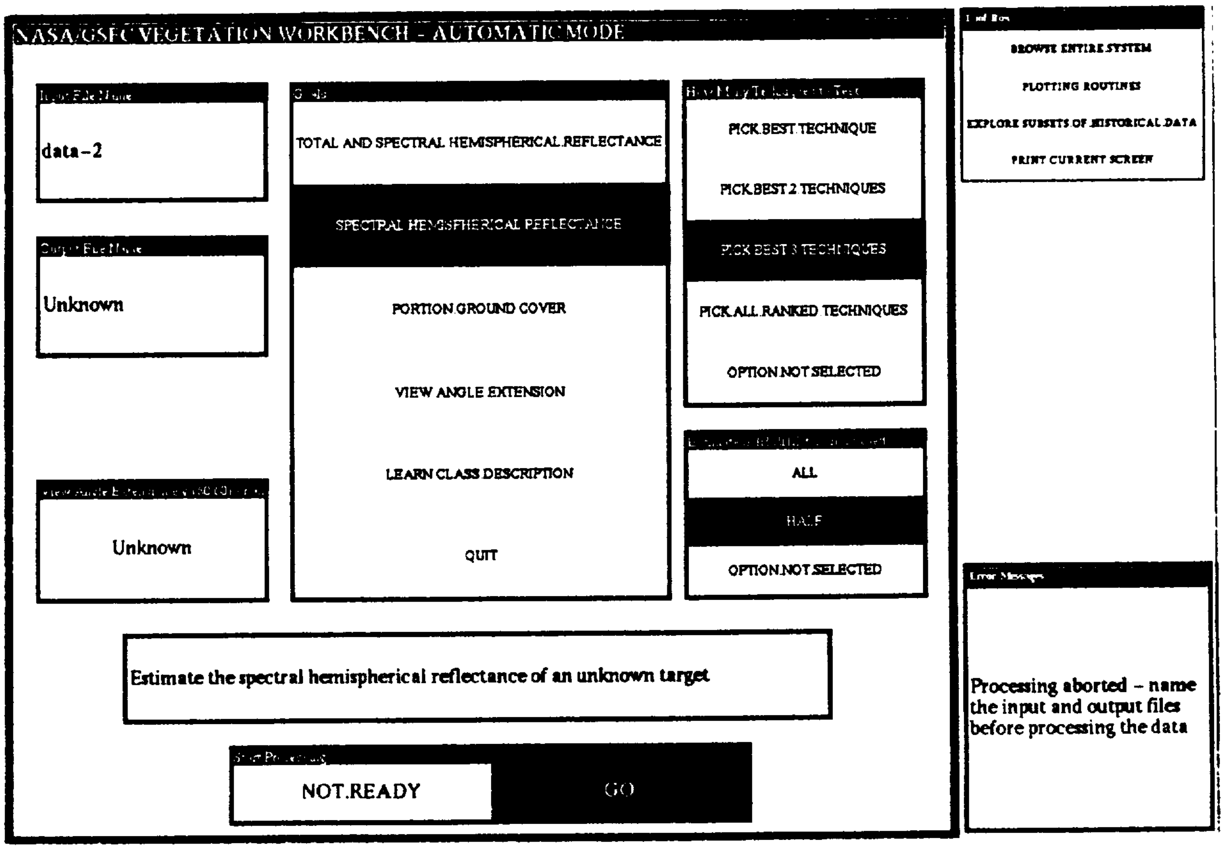


Figure 2-12

Attempting to Process the Data Before Naming the Output File

SECTION 3.0

TESTING THE INTERFACE

Using samples already in VEG and data files developed as part of Task A, all the options in the interface were tested. The tests included running VEG in the "Research Mode" and the "Automatic Mode" and selecting each of the VEG subgoals except LEARN.CLASS.DESCRIPTIONS. Some of the tests gave incorrect results the first time they were carried out. The code was debugged and the tests were repeated until the correct results were obtained in every test. The output files produced by the test runs are listed in Appendix B.

3.1 TESTING THE INTERFACE - RESEARCH MODE

In the first seven tests, VEG was run in "Research Mode" and the subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE was selected. The data was from SAMPLE5 in the VEG example data set. The system selected the restricted data set and the techniques. The best three techniques for each wavelength were used. The techniques were executed using half the historical data for the coefficients and half for the error estimation. Table 3-1 shows the parameters and formats selected for these tests. All the output parameters and all the possible formats for the reflectance data and results were included in these tests. All the tests were successful.

Table 3-1

Parameters and Formats Selected for Tests 1-7

Test Number	Parameter	Format
1	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESULTS	LIST.OF.LISTS ALL.RESULT
2	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESULTS	REFLECTANCE.VALUES.ONLY RESULT.ONLY
3	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESULTS	POINTS.AS.LISTS ALL.RESULT
4	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESULTS	NO.PARENTHESES RESULT.ERROR.ONLY
5	SAMPLE.NUMBER DESCRIPTION SOLAR.ZENITH GROUND.COVER TARGET.CLASSIFICATION WAVELENGTH REFLECTANCE.DATA RESTRICTED.DATA.SET STRING.CHARACTERIZATION RESULTS	LIST.OF.LISTS ALL.RESULT
6	TEMPLATE.1	
7	TEMPLATE.2	

In tests 8 through 11, VEG was again run in "Research Mode," but different VEG subgoals were tested. The data was taken from the VEG example data sets. The system selected the restricted data set and the techniques. The best three techniques were used and the techniques were executed using half the historical data for the coefficients and half for the error estimation. Table 3-2 summarizes tests 8 through 11. All the data was correctly written to the files. The coding of the VEG subgoal PORTION.GROUND.COVER is incomplete. All the techniques for this subgoal currently return the dummy result "1." This result was correctly written to the file in test 10.

The first eleven tests showed that the interface to an output file was working correctly in the "Research Mode" for the VEG subgoals SPECTRAL.HEMISPHERICAL.REFLECTANCE, TOTAL.AND.SPECTRAL.HEMISPHERICAL.REFLECTANCE, PORTION.GROUND.COVER and VIEW.ANGLE.EXTENSION.

Table 3-2

Summary of Tests 8 - 11

Test Number	VEG Subgoal	Data	Parameter	Format
8	TOTAL.AND.SPECTRAL .HEMISPHERICAL. REFLECTANCE	SAMPLE3	TEMPLATE.1	
9	TOTAL.AND.SPECTRAL .HEMISPHERICAL. REFLECTANCE	SAMPLE3	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESULTS	LIST.OF.LISTS ALL.RESULT
10	PORTION.GROUND. COVER	SAMPLE1	TEMPLATE.3	
11	VIEW.ANGLE. EXTENSION to (45 75)	SAMPLE1	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESULTS	LIST.OF.LISTS ALL.RESULT

3.2 TESTING THE INTERFACE - AUTOMATIC MODE

In tests 12 through 17, VEG was run in "Automatic Mode." The input file data-2 was selected. All the VEG subgoals that were previously tested using the "Research Mode" were tested again using the "Automatic Mode." These tests are summarized in Table 3-3. The data was written to the file correctly in all of tests 12 through 17. The tests showed that all the options in the interface to output the results to a file were working correctly in the "Automatic Mode."

Table 3-3

Summary of Tests in the VEG "Automatic Mode"

Test Number	VEG Subgoal	Parameter	Format
12	SPECTRAL.HEMISPHERICAL .REFLECTANCE	TEMPLATE.1	
13	SPECTRAL.HEMISPHERICAL .REFLECTANCE	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESULTS	POINTS.AS.LISTS RESULT.ONLY
14	TOTAL.AND.SPECTRAL. HEMISPHERICAL. REFLECTANCE	TEMPLATE.2	
15	VIEW.ANGLE.EXTENSION to all angles	SAMPLE.NUMBER SOLAR.ZENITH GROUND.COVER WAVELENGTH REFLECTANCE.DATA RESULTS	LIST.OF.LISTS ALL.RESULT
16	VIEW.ANGLE.EXTENSION to (60 75)	TEMPLATE.1	
17	PORTION.GROUND.COVER	SAMPLE.NUMBER SOLAR.ZENITH WAVELENGTH REFLECTANCE.DATA RESTRICTED.DATA.SET RESULTS	LIST.OF.LISTS ALL.RESULT

As task G in the optional second year of the contract, JJM Systems will redesign the data base of historical cover types as a series of flat files that are external to VEG and provide an interface between VEG and these files. The interface that has been developed in Task B will be extended so that the results from VEG can be stored in the data base of historical cover types. As a first step towards this interface tests 18 and 19 were designed to test the output of data from VEG to a file and then the reading of the data from that file for processing again.

In test 18, VEG was run in "Automatic Mode." The input file was data-1 and the output file was test-18. The output parameters solar zenith, wavelength and reflectance data (in list of lists format) were selected. The data was processed for the goal SPECTRAL.HEMISPHERICAL.REFLECTANCE and the required parameters were written to the file. In test 19, the input file was test -18 and a new format was defined to match this file. The output file was test-19, and the same output parameters selected for test 18 were used. The data was processed for the goal SPECTRAL.HEMISPHERICAL.REFLECTANCE and the required data was written to the file. Inspection of the files test-18 and test-19 revealed that they were identical. Tests 18 and 19 successfully demonstrated the writing of data to a file and the subsequent re-reading of data from the file back into VEG. These tests suggest that the extension of the interface form Task B to meet some of the requirements of Task G will be a straight forward process.

SECTION 4.0

CONCLUSIONS

The interface that allows the results from VEG to be written to a file was successfully implemented. This interface is available when VEG is running in either the "Automatic Mode" or the "Research Mode." Results from the subgoals SPECTRAL.HEMISPHERICAL.REFLECTANCE, TOTAL.AND.SPECTRAL.HEMISPHERICAL.REFLECTANCE,VIEW.ANGLE.EXTENSION and PORTION.GROUND.COVER can be written to the file. The user can specify the parameters to be written and the format to be used or can select from a number of standard templates.

The interface was comprehensively tested using both operating modes, all relevant subgoals, all possible formats and all the currently available templates. All the tests were successful, showing that the interface was operating correctly. A simple test to write data to a file and then read it back into VEG was also successful.

The interface has been designed to facilitate expansion of the options available. New file templates can be easily incorporated into VEG as they became available. As part of Task G in the optional second%% year of the contract, the interface will be expanded to allow the results from VEG to be incorporated back into the data base of historical cover types for future runs.

APPENDIX A

LISP CODE FOR THE INTERFACE TO OUTPUT THE RESULTS TO A FILE

```
;;; veg-methods2.lisp
;;;
;;; Code for Task B of the NASA VEG project, 1992
;;;
;;; Written by Ann Harrison
;;; Created 31st August 1992
;;; Last Modified 8th September 1992

(in-package 'kee)

(defun ask-to-output-to-file ()
"Asks the user whether the data should be written to a file. If the answer is
yes, opens the output-to-file interface."
  (put.value '9.output 'message
    "Do you want to output the results to a file?")
  (remove.all.values '9.output 'yes.no)
  (remove.all.values '9.output 'output.file.name)
  (unitmsg 'viewport-9.output.2 'open-panel!))

(defun open-output-to-file-interface ()
"Opens the interface for outputting the results to a file."
  (remove.all.values '9.output 'output.parameters)
  (remove.all.values '9.output 'format.list)
  (put.value '9.output 'message
"Select the parameters to be output to the file, in the correct order =>")
  (unitmsg 'viewport-9.output.3 'open-panel!))

(defun open-output-to-file-results-interface (slot)
"Opens the window that allows the user to select the format for the
results."
  (put.value 'windowpane-output.parameters-of-9.output.4 'mouseleftfn!
    'deactivate-left-mouse)
  (remove.all.values '9.output 'result.format)
  (put.value '9.output 'message (format ()
    "Select the format for the ~S =>" slot))
  (unitmsg 'windowpane-result.format-of-9.output.1 'open!))

(defun open-output-to-file-rd-interface ()
"Opens the window that allows the user to select the format for the
reflectance data."
  (put.value 'windowpane-output.parameters-of-9.output.4 'mouseleftfn!
    'deactivate-left-mouse)
  (remove.all.values '9.output 'reflectance.data.format)
  (put.value '9.output 'message
    "Select the format for the reflectance data =>")
  (unitmsg 'windowpane-reflectance.data.format-of-9.output.2 'open!))
```

```
(defun open-output-to-file-template-interface ()
"Opens the window that allows the user to select the format for the
reflectance data."
  (put.value 'windowpane-output.parameters-of-9.output.4 'mouseleftfn!
    'deactivate-left-mouse)
  (remove.all.values '9.output 'template.number)
  (put.value '9.output 'message
     "Select the required template =>")
  (unitmsg 'windowpane-template.message-of-9.output.1 'open!)
  (unitmsg 'windowpane-template.number-of-9.output.3 'open!))

(defun remove-format (slot)
"Removes the slot name and the format specified for the slot from the slot
format.list of the unit 9.output."
  (let* ((format-list (get.values '9.output 'format.list))
         (slot-pos (position slot format-list))
         (slot-format (nth (1- slot-pos) format-list)))
   (remove.value '9.output 'format.list slot-format)
   (remove.value '9.output 'format.list slot)))

(defun store-template (template)
"Stores the format corresponding to the specified format."
  (put.values '9.output 'format.list
    (case template
        (template.1 '(template.1))
        (template.2 '(result.only results
                      no.parentheses reflectance.data))
        (template.3 '(all.result results
                      no.parentheses reflectance.data
                      wavelength
                      solar.zenith)))))

(defun write-results-to-file ()
"Writes the results to a file in the specified format."
  (let ((format-list (reverse (get.values '9.output 'format.list)))
        (total-hr-p (total.hr.p)))
   (with-open-file (out-str (get.value '9.output 'output.file.name)
                            :direction :output
                            :if-does-not-exist :create)
     (if (eq (first format-list) 'template.1)
         (output-data-to-file (get.value 'estimate.hemispherical.reflectance
                                         'current.sample)
                              out-str
                              total-hr-p)
         (write-results-to-file-aux out-str format-list total-hr-p)))))
```

```
(defun write-results-to-file-aux (out-str format-list total-hr-p)
 (let ((temp nil)
       (thr-results (if (and (member 'results format-list)
                             total-hr-p)
                        (get-simple-thr-results)
                        nil)))
  (dolist (wave (get.values 'estimate.hemispherical.reflectance
                 'current.sample.wavelengths))
   (dolist (parameter format-list)
      (cond ((eq parameter 'done) nil)
            ((and (null temp)
                  (member parameter '(results reflectance.data)))
             (setf temp parameter))
            ((null temp) (write-simple-results-to-file out-str wave
                                                      parameter))
            (t (write-complex-results-to-file out-str wave temp
                                              parameter thr-results)
               (setf temp nil)))))))

(defun write-simple-results-to-file (out-str wave parameter)
 (case parameter
  (sample.number
       (princ (unit.name (first (unit.parents wave 'member))) out-str))
  (description (princ
                (get.value '9.output 'sample.description) out-str))
  (solar.zenith (princ
                 (get.value wave 'solar.zenith) out-str))
  (ground.cover (princ
                 (get.value wave 'ground.cover) out-str))
  (target.classification (princ
        (get.value '9.output 's-results) out-str))
  (wavelength (princ
               (get.value wave 'wavelength) out-str))
  (restricted.data.set (princ
                        (get-unit-names (get.values wave 'r.d.s)) out-str))
  (string.characterization
        (princ (get-characterizations
                (get.value wave 'string.objects)) out-str)))
 (princ " " out-str))

(defun write-complex-results-to-file (out-str wave slot format thr-results)
 (if (eq slot 'results)
    (write-results out-str wave format thr-results)
    (write-reflectance-data out-str wave format)))
```

```
(defun write-results (out-str wave format thr-results)
  (let ((cw-name (unit.name wave))
        (techs (get.values wave 'techniques))
        (results nil))
    (dolist (tech techs)
     (let* ((tech-name (unit.name tech))
            (result-unit (intern (string-append (string cw-name)
                                                "-" (string tech-name)))))
        (setf results
            (push (list tech-name
                        (get.value result-unit 'calc.spectral.hem.refl)
                        (get.value result-unit 'shr.error.estimate)
                        (get-coeff-values result-unit))
                  results))))
    (setf results (append thr-results results))
    (if (null results)
        (princ "nil " out-str)
        (case format
          (all.result
            (dolist (res results)
                (dolist (res-part res)
                  (princ res-part out-str)
                  (princ " " out-str))))
          (result.error.only
                (dolist (res results)
                  (princ (second res) out-str)
                  (princ " " out-str)
                  (princ (third res) out-str)
                  (princ " " out-str)))
          (result.only
                (dolist (res results)
                  (princ (second res) out-str)
                  (princ " " out-str)))))))

(defun get-simple-thr-results ()
"Returns a list of results for total hemispherical techniques.  Each result
consists of the technique name, result, error estimate and n value."
  (let* ((samp
          (get.value 'estimate.hemispherical.reflectance 'current.sample))
         (samp-name (unit.name samp))
         (results nil))
    (dolist (tech (get.values 'estimate.t.hemispherical.reflectance
                              't.techniques)
            results)
      (let* ((tech-name (unit.name tech))
             (result-unit (intern (string-append (string samp-name)
                                                 "-" (string tech-name)))))
        (push `(,tech-name
                ,(get.value result-unit 'calc.total.hem.refl)
                ,(get.value result-unit 'thr.error.estimate)
                ,(get.value result-unit 'n))
              results)))))
```

```
(defun write-reflectance-data (out-str wave format)
 (let ((reflectance-data (get.value wave 'reflectance.data)))
  (case format
   (list.of.lists
        (princ reflectance-data out-str)
        (princ " " out-str))
   (points.as.lists
         (princ (length reflectance-data) out-str)
         (princ " " out-str)
         (dolist (point reflectance-data)
          (princ point out-str)
          (princ " " out-str)))
   (no.parentheses
         (princ (length reflectance-data) out-str)
         (princ " " out-str)
       (dolist (point reflectance-data)
        (dolist (ele point)
         (princ ele out-str)
         (princ " " out-str))))
   (t (princ (length reflectance-data) out-str)
       (princ " " out-str)
       (dolist (point reflectance-data)
        (princ (third point) out-str)
        (princ " " out-str))))))
```

APPENDIX B

LISTINGS OF FILES PRODUCED BY THE TEST RUNS

LISTINGS OF THE FILES PRODUCED BY THE TEST RUNS

Listing of the File test-1

SAMPLE5 71 0.92 ((0 0 0.5)) DIRECT.NADIR 0.5 0.2325 none NADIR 0.5561 0.1879 0.1145 0.8831 SAMPLE5 71 0.68 ((0 0 0.043) (15 222 0.043) (15 42 0.043) (30 220 0.054) (30 45 0.043) (45 225 0.066) (45 43 0.044) (60 230 0.076) (60 48 0.054) (75 225 0.089) (20 0 0.05) (40 5 0.06) (60 355 0.07) (20 92 0.055) (40 85 0.06) (65 90 0.07)) NORMAN 0.0611 0.0529 none NORMAN.PLUS 0.0642 0.1009 -0.0101 1.2157 2FULL.1HALF.STRINGS 0.0579 0.1592 -2.3446 -3.5426 7.3860

Listing of the File test-2

SAMPLE5 71 0.92 1 0.5 0.5 0.5561 SAMPLE5 71 0.68 16 0.043 0.043 0.043 0.054 0.043 0.066 0.044 0.076 0.054 0.089 0.05 0.06 0.07 0.055 0.06 0.07 0.0611 0.0642 0.0579

Listing of the File test-3

SAMPLE5 71 0.92 1 (0 0 0.5) DIRECT.NADIR 0.5 0.2325 none NADIR 0.5561 0.1879 0.1145 0.8831 SAMPLE5 71 0.68 16 (0 0 0.043) (15 222 0.043) (15 42 0.043) (30 220 0.054) (30 45 0.043) (45 225 0.066) (45 43 0.044) (60 230 0.076) (60 48 0.054) (75 225 0.089) (20 0 0.05) (40 5 0.06) (60 355 0.07) (20 92 0.055) (40 85 0.06) (65 90 0.07) NORMAN 0.0611 0.0529 none NORMAN.PLUS 0.0642 0.1009 -0.0101 1.2157 2FULL.1HALF.STRINGS 0.0579 0.1592 -2.3446 -3.5426 7.3860

Listing of the File test-4

SAMPLE5 71 0.92 1 0 0 0.5 0.5 0.2325 0.5561 0.1879 SAMPLE5 71 0.68 16 0 0 0.043 15 222 0.043 15 42 0.043 30 220 0.054 30 45 0.043 45 225 0.066 45 43 0.044 60 230 0.076 60 48 0.054 75 225 0.089 20 0 0.05 40 5 0.06 60 355 0.07 20 92 0.055 40 85 0.06 65 90 0.07 0.0611 0.0529 0.0642 0.1009 0.0579 0.1592

Listing of the File test-5

SAMPLE5 Cover type "example of dense vegetation canopy": Solar Zenith Angle 71:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL 71 0.927025782688766 Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270 0.92 ((0 0 0.5)) (CT10-49-2 CT5-59-2 CT8-56-2 CT6-63-2 CT6-79-2 CT7-59-2 CT7-74-2 CT10-63-2 CT8-70-2 CT10-76-2) No strings found DIRECT.NADIR 0.5 0.2325 none NADIR 0.5561 0.1879 0.1145 0.8831
SAMPLE5 Cover type "example of dense vegetation canopy": Solar Zenith Angle 71:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL 71 0.927025782688766 Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270 0.68 ((0 0 0.043) (15 222 0.043) (15 42 0.043) (30 220 0.054) (30 45 0.043) (45 225 0.066) (45 43 0.044) (60 230 0.076) (60 48 0.054) (75 225 0.089) (20 0 0.05) (40 5 0.06) (60 355 0.07) (20 92 0.055) (40 85 0.06) (65 90 0.07)) (CT10-49-1 CT5-59-1 CT8-56-1 CT6-63-1 CT6-79-1 CT7-59-1 CT7-74-1 CT10-63-1 CT8-70-1 CT10-76-1) Strings:
COMPLETE HALF-string with 0 degrees azimuth
COMPLETE FULL-string with 42 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth NORMAN 0.0611 0.0529 none NORMAN.PLUS 0.0642 0.1009 -0.0101 1.2157 2FULL.1HALF.STRINGS 0.0579 0.1592 -2.3446 -3.5426 7.3860

Listing of the File test-6

Results for sample SAMPLE5
Sample input data:
Cover type "example of dense vegetation canopy": Solar Zenith Angle 71:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270

Wavelength 0.92
Reflectance data ((0 0 0.5))
Results:
Technique NADIR Estimate 0.5561 Error 0.1879 Coefficients 0.1145 0.8831
Technique DIRECT.NADIR Estimate 0.5000 Error 0.2325 Coefficients none
Restricted Historical Data:
(CT10-49-2 CT5-59-2 CT8-56-2 CT6-63-2 CT6-79-2 CT7-59-2 CT7-74-2 CT10-63-2 CT8-70-2 CT10-76-2)
Data Characterization:
Nadir data is available
No strings found
Wavelength 0.68
Reflectance data ((0 0 0.043) (15 222 0.043) (15 42 0.043) (30 220 0.054) (30 45 0.043) (45 225 0.066) (45 43 0.044) (60 230 0.076) (60 48 0.054) (75 225 0.089) (20 0 0.05) (40 5 0.06) (60 355 0.07) (20 92 0.055) (40 85 0.06) (65 90 0.07))
Results:
Technique 2FULL.1HALF.STRINGS Estimate 0.0579 Error 0.1592 Coefficients -2.3446 -3.5426 7.3860
Technique NORMAN.PLUS Estimate 0.0642 Error 0.1009 Coefficients -0.0101 1.2157
Technique NORMAN Estimate 0.0611 Error 0.0529 Coefficients none
Restricted Historical Data:
(CT10-49-1 CT5-59-1 CT8-56-1 CT6-63-1 CT6-79-1 CT7-59-1 CT7-74-1 CT10-63-1 CT8-70-1 CT10-76-1)
Data Characterization:
Nadir data is available
Strings:
COMPLETE HALF-string with 0 degrees azimuth
COMPLETE FULL-string with 42 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth

Listing of the File test-7

1 0 0 0.5 0.5 0.5561 16 0 0 0.043 15 222 0.043 15 42 0.043 30 220 0.054 30 45 0.043 45 225 0.066 45 43 0.044 60 230 0.076 60 48 0.054 75 225 0.089 20 0 0.05 40 5 0.06 60 355 0.07 20 92 0.055 40 85 0.06 65 90 0.07 0.0611 0.0642 0.0579

Listing of the File test-8

Results for sample SAMPLE5
Sample input data:
Cover type "example of dense vegetation canopy": Solar Zenith Angle 71:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270 Total hemispherical reflectance results:-
Technique KIMES Estimate 0.3213 Error not available n 0

Total hemispherical reflectance results:-
Technique KIMES Estimate 0.3213 Error not available n 0

Wavelength 0.92
Reflectance data ((0 0 0.5))
Results:
Technique NADIR Estimate 0.5815 Error 0.1559 Coefficients 0.0861 0.9907
Technique DIRECT.NADIR Estimate 0.5000 Error 0.2325 Coefficients none
Restricted Historical Data:
(CT10-76-2 CT8-70-2 CT10-63-2 CT7-74-2 CT7-59-2 CT6-79-2 CT6-63-2 CT8-56-2 CT5-59-2 CT10-49-2)
Data Characterization:
Nadir data is available
No strings found
Wavelength 0.68
Reflectance data ((0 0 0.043) (15 222 0.043) (15 42 0.043) (30 220 0.054) (30 45 0.043) (45 225 0.066) (45 43 0.044) (60 230 0.076) (60 48 0.054) (75 225 0.089) (20 0 0.05) (40 5 0.06) (60 355 0.07) (20 92 0.055) (40 85 0.06) (65 90 0.07))
Results:
Technique 2FULL.1HALF.STRINGS Estimate 0.0617 Error 0.0795 Coefficients -0.8361 -10.0132 13.2145
Technique NORMAN.PLUS Estimate 0.0588 Error 0.0900 Coefficients 0.0073 0.8420
Technique NORMAN Estimate 0.0611 Error 0.0529 Coefficients none
Restricted Historical Data:
(CT10-76-1 CT8-70-1 CT10-63-1 CT7-74-1 CT7-59-1 CT6-79-1 CT6-63-1 CT8-56-1 CT5-59-1 CT10-49-1)
Data Characterization:
Nadir data is available
Strings:
COMPLETE HALF-string with 0 degrees azimuth
COMPLETE FULL-string with 42 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth

Listing of the File test-9

SAMPLE5 71 0.92 ((0 0 0.5)) KIMES 0.3213 not available 0 DIRECT.NADIR 0.5 0.2325 none NADIR 0.5815 0.1559 0.0861 0.9907 SAMPLE5 71 0.68 ((0 0 0.043) (15 222 0.043) (15 42 0.043) (30 220 0.054) (30 45 0.043) (45 225 0.066) (45 43 0.044) (60 230 0.076) (60 48 0.054) (75 225 0.089) (20 0 0.05) (40 5 0.06) (60 355 0.07) (20 92 0.055) (40 85 0.06) (65 90 0.07)) KIMES 0.3213 not available 0 NORMAN 0.0611 0.0529 none NORMAN.PLUS 0.0588 0.09 0.0073 0.8420 2FULL.1HALF.STRINGS 0.0617 0.0795 -0.8361 -10.0132 13.2145

Listing of the File test-10

45 0.68 11 0 0 0.043 15 180 0.043 15 0 0.043 30 180 0.054 30 0 0.043 45 180 0.066 45 0 0.044 60 180 0.076 60 0 0.054 75 180 0.089 75 0 0.067 2.WAVE.NEAR.NADIR 1.0 0.2227 none NEW.TECH.1 1.0 0.2227 none NEW.TECH.2 1.0 0.2227 none 45 0.92 1 0 0 0.5 1.WAVE.NEAR.NADIR 1.0 0.2227 none NEW.TECH.1 1.0 0.2227 none NEW.TECH.2 1.0 0.2227 none

Listing of the File test-11

SAMPLE7 45 0.68 ((45 0 0.044) (45 180 0.066) (60 0 0.054) (60 180 0.076)) MIN.DISTANCE 0.044 0.2361 none MULTIPLE.VIEW.TO.VIEW1 0.0472 0.1181 0.0071 0.7608 SAMPLE7 45 0.92 ((0 0 0.31)) MIN.DISTANCE 0.31 0.1928 none VIEW1.TO.VIEW1 0.3126 0.1868 -0.0260 1.0923

Listing of the File test-12

Results for sample FILE-SAMPLE-1212
Sample input data:
Cover type "Example of dense vegetation canopy": Solar Zenith Angle 45:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270

Wavelength 0.68
Reflectance data ((0 0 0.043))
Results:
Technique NADIR Estimate 0.0522 Error 0.2224 Coefficients 0.0171 0.8159
Technique DIRECT.NADIR Estimate 0.0430 Error 0.2973 Coefficients none
Restricted Historical Data:
(CT6-63-1 CT7-59-1 CT10-63-1 CT10-28-1 CT5-42-1 CT8-56-1 CT6-45-1 CT7-41-1 CT8-42-1 CT10-49-1)
Data Characterization:
Nadir data is available
No strings found
Wavelength 0.92
Reflectance data ((0 0 0.5))
Results:
Technique NADIR Estimate 0.5354 Error 0.2542 Coefficients 0.1526 0.7656
Technique DIRECT.NADIR Estimate 0.5000 Error 0.1702 Coefficients none
Restricted Historical Data:
(CT6-63-2 CT7-59-2 CT10-63-2 CT10-28-2 CT5-42-2 CT8-56-2 CT6-45-2 CT7-41-2 CT8-42-2 CT10-49-2)
Data Characterization:
Nadir data is available
No strings found

Results for sample FILE-SAMPLE-1209
Sample input data:
Cover type "Example of dense vegetation canopy": Solar Zenith Angle 45:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270

Wavelength 0.68
Reflectance data ((0 0 0.043) (15 180 0.043) (15 0 0.043) (30 180 0.054) (30 0 0.043) (45 180 0.066) (45 0 0.044) (60 180 0.076) (60 0 0.054) (75 180 0.089) (75 0 0.067))
Results:
Technique 1FULL.STRING Estimate 0.0514 Error 0.1035 Coefficients 0.9008
Technique NORMAN.PLUS Estimate 0.0513 Error 0.1363 Coefficients -0.0087 1.0420
Technique NORMAN Estimate 0.0576 Error 0.1788 Coefficients none
Restricted Historical Data:
(CT6-63-1 CT7-59-1 CT10-63-1 CT10-28-1 CT5-42-1 CT8-56-1 CT6-45-1 CT7-41-1 CT8-42-1 CT10-49-1)
Data Characterization:
Nadir data is available
Strings:
COMPLETE FULL-string with 0 degrees azimuth
Wavelength 0.92
Reflectance data ((0 0 0.5))
Results:
Technique NADIR Estimate 0.5354 Error 0.2542 Coefficients 0.1526 0.7656
Technique DIRECT.NADIR Estimate 0.5000 Error 0.1702 Coefficients none
Restricted Historical Data:
(CT6-63-2 CT7-59-2 CT10-63-2 CT10-28-2 CT5-42-2 CT8-56-2 CT6-45-2 CT7-41-2 CT8-42-2 CT10-49-2)
Data Characterization:
Nadir data is available
No strings found

Listing of the File test-13

FILE-SAMPLE-1334 45 0.68 1 (0 0 0.043) 0.043 0.0522 FILE-SAMPLE-1334 45 0.92 1 (0 0 0.5) 0.5 0.5354 FILE-SAMPLE-1331 45 0.68 11 (0 0 0.043) (15 180 0.043) (15 0 0.043) (30 180 0.054) (30 0 0.043) (45 180 0.066) (45 0 0.044) (60 180 0.076) (60 0 0.054) (75 180 0.089) (75 0 0.067) 0.0576 0.0513 0.0514 FILE-SAMPLE-1331 45 0.92 1 (0 0 0.5) 0.5 0.5354

Listing of the File test-14

1 0 0 0.043 0.2955 0.043 0.069 1 0 0 0.5 0.2955 0.5 0.522 11 0 0 0.043 15 180 0.043 15 0 0.043 30 180 0.054 30 0 0.043 45 180 0.066 45 0 0.044 60 180 0.076 60 0 0.054 75 180 0.089 75 0 0.067 0.288 0.0576 0.054 0.0543 1 0 0 0.5 0.288 0.5 0.522

Listing of the File test-15

FILE-SAMPLE-1578 45 0.927025782688766 0.68 ((0 0 0.043)) (CT6-63-1 CT7-59-1 CT10-63-1 CT10-28-1 CT5-42-1 CT8-56-1 CT6-45-1 CT7-41-1 CT8-42-1 CT10-49-1) nil FILE-SAMPLE-1578 45 0.927025782688766 0.92 ((0 0 0.5)) (CT6-63-2 CT7-59-2 CT10-63-2 CT10-28-2 CT5-42-2 CT8-56-2 CT6-45-2 CT7-41-2 CT8-42-2 CT10-49-2) nil FILE-SAMPLE-1575 45 0.927025782688766 0.68 ((0 0 0.043) (15 180 0.043) (15 0 0.043) (30 180 0.054) (30 0 0.043) (45 180 0.066) (45 0 0.044) (60 180 0.076) (60 0 0.054) (75 180 0.089) (75 0 0.067)) (CT6-63-1 CT7-59-1 CT10-63-1 CT10-28-1 CT5-42-1 CT8-56-1 CT6-45-1 CT7-41-1 CT8-42-1 CT10-49-1) MULTIPLE.VIEW.TO.MULTIPLE.VIEW ((0 0 0.0423) (15 0 0.0411) (30 0 0.0428) (45 0 0.0473) (60 0 0.0547) (75 0 0.0649) (15 45 0.0419) (30 45 0.0443) (45 45 0.0496) (60 45 0.0578) (75 45 0.0688) (15 90 0.0437) (30 90 0.048) (45 90 0.0551) (60 90 0.0651) (75 90 0.0779) (15 135 0.0455) (30 135 0.0517) (45 135 0.0606) (60 135 0.0725) (75 135 0.0871) (15 180 0.0463) (30 180 0.0532) (45 180 0.0629) (60 180 0.0755) (75 180 0.0909) (15 225 0.0455) (30 225 0.0517) (45 225 0.0606) (60 225 0.0725) (75 225 0.0871) (15 270 0.0437) (30 270 0.048) (45 270 0.0551) (60 270 0.0651) (75 270 0.0779) (15 315 0.0419) (30 315 0.0443) (45 315 0.0496) (60 315 0.0578) (75 315 0.0688)) 0.1682 none FILE-SAMPLE-1575 45 0.927025782688766 0.92 ((0 0 0.5)) (CT6-63-2 CT7-59-2 CT10-63-2 CT10-28-2 CT5-42-2 CT8-56-2 CT6-45-2 CT7-41-2 CT8-42-2 CT10-49-2) nil

Listing of the File test-16

Results for sample FILE-SAMPLE-1642
Sample input data:
Cover type "Example of dense vegetation canopy": Solar Zenith Angle 45:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
No target characterization

Wavelength 0.68
Reflectance data ((0 0 0.043) (15 180 0.043) (15 0 0.043) (30 180 0.054) (30 0 0.043) (45 180 0.066) (45 0 0.044) (60 180 0.076) (60 0 0.054) (75 180 0.089) (75 0 0.067))
Results:
View angle extension to angle(s): ((60 75)) Technique MULTIPLE.VIEW.TO.VIEW1 Estimate 0.0575 Error 0.0884 Coefficients 0.0030 0.8739
Technique MIN.DISTANCE Estimate 0.0430 Error 0.3574 Coefficients none
Restricted Historical Data:
(CT7-23-1 CT7-41-1 CT7-59-1 CT9-46-1 CT9-23-1 CT5-28-1 CT5-42-1 CT5-59-1 CT11-58-1 CT11-45-1)
Data Characterization:
Nadir data is available
Strings:
COMPLETE FULL-string with 0 degrees azimuth

Listing of the File test-17

FILE-SAMPLE-1688 45 0.68 ((0 0 0.043)) (CT6-63-1 CT7-59-1 CT10-63-1 CT10-28-1 CT5-42-1 CT8-56-1 CT6-45-1 CT7-41-1 CT8-42-1 CT10-49-1) 1.WAVE.NEAR.NADIR 1.0 0.2227 none NEW.TECH.1 1.0 0.2227 none NEW.TECH.2 1.0 0.2227 none FILE-SAMPLE-1688 45 0.92 ((0 0 0.5)) (CT6-63-2 CT7-59-2 CT10-63-2 CT10-28-2 CT5-42-2 CT8-56-2 CT6-45-2 CT7-41-2 CT8-42-2 CT10-49-2) 1.WAVE.NEAR.NADIR 1.0 0.2227 none NEW.TECH.1 1.0 0.2227 none NEW.TECH.2 1.0 0.2227 none FILE-SAMPLE-1685 45 0.68 ((0 0 0.043) (15 180 0.043) (15 0 0.043) (30 180 0.054) (30 0 0.043) (45 180 0.066) (45 0 0.044) (60 180 0.076) (60 0 0.054) (75 180 0.089) (75 0 0.067)) (CT6-63-1 CT7-59-1 CT10-63-1 CT10-28-1 CT5-42-1 CT8-56-1 CT6-45-1 CT7-41-1 CT8-42-1 CT10-49-1) 2.WAVE.NEAR.NADIR 1.0 0.2227 none NEW.TECH.1 1.0 0.2227 none NEW.TECH.2 1.0 0.2227 none FILE-SAMPLE-1685 45 0.92 ((0 0 0.5)) (CT6-63-2 CT7-59-2 CT10-63-2 CT10-28-2 CT5-42-2 CT8-56-2 CT6-45-2 CT7-41-2 CT8-42-2 CT10-49-2) 1.WAVE.NEAR.NADIR 1.0 0.2227 none NEW.TECH.1 1.0 0.2227 none NEW.TECH.2 1.0 0.2227 none

Listing of the File test-18

45 0.68 ((0 0 0.043) (15 180 0.043) (15 0 0.043) (30 180 0.054) (30 0 0.043) (45 180 0.066) (45 0 0.044) (60 180 0.076) (60 0 0.054) (75 180 0.089) (75 0 0.067))

Listing of the File test-19

45 0.68 ((0 0 0.043) (15 180 0.043) (15 0 0.043) (30 180 0.054) (30 0 0.043) (45 180 0.066) (45 0 0.044) (60 180 0.076) (60 0 0.054) (75 180 0.089) (75 0 0.067))

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

An Expert System Shell for Inferring Vegetation Characteristics - Output of Results to a File (Task B)

5. Report Date

September 1992

6. Performing Organization Code

7. Author(s)

P. Ann Harrison

8. Performing Organization Report No.

B921016-U-2R02

10. Work Unit No.

462-61-14

9. Performing Organization Name and Address

JJM Systems, Inc.
One Ivybrook Blvd., Suite 190
Ivyland, PA 18974

11. Contract or Grant No.

NAS5-30127

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration,
Washington D.C. 20546-0001
NASA/Goddard Space Flight Center, Greenbelt, MD 20771

13. Type of Report and Period Covered

Task Report for Task B
August-September 1992

14. Sponsoring Agency Code

15. Supplementary Notes

The Lisp and KEE code for this work is available on a Sun Cartridge Tape.

16. Abstract

VEG is an expert system that infers vegatation characteristics from reflectance data. The results from VEG can be displayed on the screen or output to a file. A new interface that expands the options available for writing the results to a file has been implemented using KEE and Common Lisp. It allows the user to specify the file name, parameters to be output and format to be used. A number of standard file format templates are also provided. The interface has been comprehensively tested. The tests confirmed that it was operating correctly.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
EXPERT SYSTEM, ARTIFICIAL INTELLIGENCE, REMOTE SENSING	UNCLASSIFIED - UNLIMITED

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	41	

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171

APPENDIX C

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - THE LEARNING SYSTEM (TASKS C AND D)

SYSTEMS INC

B921014-U-2R03

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - THE LEARNING SYSTEM (TASKS C AND D)

26 September 1992

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771

Prepared by:

JJM Systems, Inc.
One Ivybrook Boulevard, Suite 190
Ivyland, PA 18974

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ACRONYMS

KEE	Knowledge Engineering Environment
VEG	VEGetation Workbench

SECTION 1.0

INTRODUCTION

The VEGetation Workbench (VEG) is an expert system that infers vegetation characteristics from spectral reflectance data. The first generation expert system has been implemented using the Knowledge Engineering Environment (KEE) by Intellicorp. VEG is described in references 1 and 2. VEG contains more than one thousand KEE units. Some units hold samples of reflectance data that are used for testing and demonstrating the system. Other units contain the methods and rules used for processing the data and the graphics required for the interface. When VEG is used to infer vegetation characteristics, additional units are created to hold both the intermediate and final results of processing the data.

VEG includes a data base of historical cover type data. This consists of data for 18 different cover types and, for each cover type, this includes data collected at several different solar zenith angles and wavelengths. In the first generation version of VEG, this data base was used to estimate the error term in calculations made by VEG. For example, when the spectral hemispherical reflectance of an unknown cover type was being calculated, the historical cover type data that best matched the unknown cover type was identified and interpolated and extrapolated to match the unknown sample. Techniques were applied to the unknown cover type data to estimate the spectral hemispherical reflectance. The same techniques were also applied to the historical cover type data. The spectral hemispherical reflectance of the historical cover type data was available in the data base so the error in the calculation could be calculated. The error obtained by applying the techniques to the historical cover type data gave an estimate of the error term involved in applying the techniques to the unknown cover type data.

This report describes the implementation of a learning system that uses the data base of historical cover type data to learn class descriptions of classes of cover types. These classes can include broad classes such as soil or vegetation or more specific classes such as forest, grass and wheat. The classes can also include subclasses based on continuous parameters such as 0-30% ground cover, 31-70% ground cover and 71-100% ground cover. The learning system is designed to handle any combination of directional view angles such as (0 0), (30 50), (45 60), (10 135), (40 225) where the first value in each pair is the zenith angle and the second value is the relative azimuth angle. The learning system uses sets of positive and negative examples from the data base of historical cover types to find the most important features that uniquely distinguish each class. For example, the learning system found that for solar zenith 45°, wavelength 0.68 µm and the view angles (15 182), (75 90), (0 0) and (35 45), the class 0-30% ground cover was best distinguished by the following hypotheses:-

```
((FIRST-MAX (75 90)) NIL)
((FIRST-MAX (15 182)) T)
((GREATER-THAN (15 182) (75 90)) T)
```

The first hypothesis says that the maximum reflectance value is not at the view angle (75 90). The second hypothesis says that the maximum reflectance value is at the view angle (15 182). The third hypothesis says that the reflectance at the view angle (15 182) is greater than the reflectance at the view angle (75 90). These hypotheses describe the class 0-30% ground cover for the solar zenith angle, view angles and wavelength specified. In a typical run of the learning system, class descriptions for several alternative classes such as the classes 0-30%, 31-70% and 71-100% ground cover are learned from the data base of historical cover types. These class descriptions are then used to classify an unknown cover type by finding the class that best matches the unknown cover type data.

VEG contains more than one thousand units (objects). In order to conserve memory, the learning system was developed as a separate layer sitting on top of VEG. It is only loaded when needed. VEG does not need the learning system to achieve its various goals. The learning system is fully integrated with VEG and uses some VEG objects in its processing.

Tasks C and D have been completed. The learning system has been implemented and fully integrated into the first generation version of VEG. The learning system is described in detail in this report. A Sun cartridge tape containing KEE and Common Lisp code for the learning system has been delivered to the NASA GSFC technical representative.

SECTION 2.0

OVERVIEW OF THE LEARNING SYSTEM

The learning system has been implemented as a separate knowledge base from VEG. The learning system is loaded only when it is used. It is invoked by selecting the option LEARN.CLASS.DESCRIPTION from either the VEG "Research Mode" or the VEG "Automatic Mode" top level menu. Only when one of these options has been selected, is the learning system loaded. If either of these options is subsequently selected again, the learning system knowledge base is not re-loaded. The learning system is not stand-alone. It requires that VEG be loaded first.

The organization of KEE units in the learning system knowledge base is shown in Figure 2-1. This figure shows the general organization of the learning system. The learning system is composed of three basic components: data bases, learning methods and learning rules. The units whose names end in .AV are ActiveValue units. They are attached to slots in other units and they contain methods that are activated when values are added to or removed from the slots to which they are attached. The use of specific ActiveValue units will be discussed below. The rule class LEARNING.RULES contains rules which determine which hypotheses to test for each training class.

Several units such as ENTER.LEARNING.DATA and OPTION.3 are positioned in the hierarchy of KEE units in the learning system as members of the class LEARNING.METHODS. This organization of units is for convenience only. The member units of the class LEARNING.METHODS are grouped together because they have similar uses. However, they are not strictly members of the class LEARNING.METHODS, because they do not inherit any slots or slot values from the class unit LEARNING.METHODS. The member units of the class LEARNING.METHODS contain slots required by the different methods involved in processing the learning data. For example, the slots CLASS.PARAMETER, DIRECTIONAL. VIEW.ANGLES, SOLAR.ZENITH, VALUE and WAVELENGTH in the unit ENTER. LEARNING.DATA are used to hold data entered via the interface. The slots in the unit ENTER.LEARNING.DATA are shown in Figure 2-2.

The unit LEARNING.METHODS contains no member slots that are inherited by its member units. However, this unit contains a number of own slots as listed in Figure 2-3. Some of these slots are used to ensure that the methods in the learning system are executed in the correct sequence. For example, at the beginning of a run, the slot DONE.ENTER.LEARNING.DATA.P has the value NIL. After data has been entered into the learning system, the value of this slot is changed to T. The next method in the processing of the learning data checks the value of this slot and only proceeds if the slot has the value T, indicating that the necessary data has been entered.

When a training class has been defined, a subclass of the unit TRAINING.DATABASES is set up. Data defining the class is stored in the new training problem unit. The positive and negative training sets for the class are subsequently set up as a hierarchy of units which are subclasses of the training problem unit. This is discussed in detail in Section 3. The positive and negative training set units inherit slots from the unit TRAINING.DATABASES. This unit also contains own slots which contain data that is common to all the training problems. For example, the slot MINIMUM.SET.SIZE holds the minimum acceptable training set size. Figure 2-4 shows the slots in the unit TRAINING.DATABASES.

ENTER.LEARNING.DATA.CLASS.PARAMETER.AV

ENTER.LEARNING.DATA.DVA.AV

ENTER.LEARNING.DATA.MENU.AV

ENTER.LEARNING.DATA.SOLAR.ZENITH.AV

ENTER.LEARNING.DATA.VALUE.AV

ENTER.LEARNING.DATA.VIEW.ANGLE.DATA.AV

ENTER.LEARNING.DATA.WAVELENGTH.AV

KEEPICTURE.INSTANCES

LEARNING.OPTION.1.MENU.AV

LEARNING.OPTION.2.MENU.AV

LEARNING.OPTION.3.MENU.AV

- LEARNING.SYSTEM
  - LEARNING.METHODS
    - ENTER.LEARNING.DATA
    - INITIALIZE.LEARNING.SYSTEM
    - OPTION.1
    - OPTION.2
    - OPTION.3
    - OUTPUT.LEARNING.RESULTS
  - LEARNING.RULES
    - LR1
    - LR2
    - LR3
    - LR4
  - TRAINING.DATABASES

LEARNING.SYSTEM.MENU.AV

LEARNING.SYSTEM.OPTIONS.AV

OUTPUT.LEARNING.RESULTS.MENU.AV

OUTPUT.PAR.AV

OUTPUT.TEMPLATE.NUMBER.AV

TRACE.FILE.NAME.AV

TRACE.FILE.Y.N.AV

B921014G1

Figure 2-1

The Organization of KEE Units in the Learning System

CLASS
CLASS.PARAMETER
DIRECTIONAL.VIEW.ANGLES
MENU
MESSAGE
POSSIBLE.COVER.TYPES
POSSIBLE.DESCRIPTIONS
SOLAR.ZENITH
VALUE
VIEW.ANGLE.DATA
VIEW.ANGLE.DATA.MESSAGE
WAVELENGTH

B921014G1

Figure 2-2

The Slots in the Unit ENTER.LEARNING.DATA

DONE.CLASSIFY.COVER.TYPES.P
DONE.CLASSIFY.SAMPLE.P
DONE.ENTER.LEARNING.DATA.P
DONE.LEARN.CLASS.DESCRIPTIONS.P
MESSAGE
OPTION.NUMBER
SEARCH.DEPTH
SEARCH.OVER
TRACE.FILE
YES.NO

B921014G3

Figure 2-3

Own Slots in the Unit LEARNING.METHODS

Member Slots :

BEST.CLASS
BEST.SCORE
CLASS
CORRECT.MATCHING.COVER.TYPES
COVER.TYPE
HYPOTHESES
INCORRECT.MATCHING.COVER.TYPES
NEG.TRAINING.SET
POS.TRAINING.SET
PREVIOUS.BEST.SCORE
REFLECTANCE.DATA
SAMPLE.SCORE
SCORES
SOLAR.ZENITH
VIEW.ANGLE.DATA
VIEW.ANGLE.DATA.MESSAGE
WAVELENGTH

Own Slots :

CURRENT.CLASSES
MAXIMUM.SET.SIZE
MINIMUM.SET.SIZE
NUM.SCORES
PERFORMANCE.SCORE

B921014G4

Figure 2-4

Slots in the Unit TRAINING.DATABASES

The KEE system contains a graphics package called "ActiveImages". This package was used to build the interface for VEG. Using this package, a comprehensive interface was built for the learning system, and this was fully integrated into the existing VEG system. The interface allows the scientist to run VEG and select options at all stages of the run by clicking the mouse over the appropriate menu option. The interface allows a scientist with no knowledge of KEE, Common Lisp or the detailed structure of VEG to use the system with ease. The only time that the scientist needs to use the keyboard during a run is when he/she enters new data manually. All other operations are controlled by the mouse. The interface allows the scientist to focus on the data and the functions performed by VEG. It abstracts away most of the underlying detailed complexity of the VEG system.

In the VEG "Research Mode," the learning system is invoked by selecting the option LEARN.CLASS.DESCRIPTION from the top level menu, as shown in Figure 2-5. Mousing on the option SELECT.OPTION causes the learning system to be loaded (if it has not already been loaded). The learning system main menu then appears on the screen. This menu is shown in Figure 2-6. If the user mouses on one of the option numbers, a brief description of the option appears in the box below the option menu. If the user then mouses on SELECT.OPTION, the option is selected and the menu for the selected option replaces the learning system main menu on the screen. The operation in the VEG "Research Mode" of each of the options in the learning system is discussed in detail in the next section. The operation of the learning system in the VEG "Automatic Mode" is discussed in Section 4.

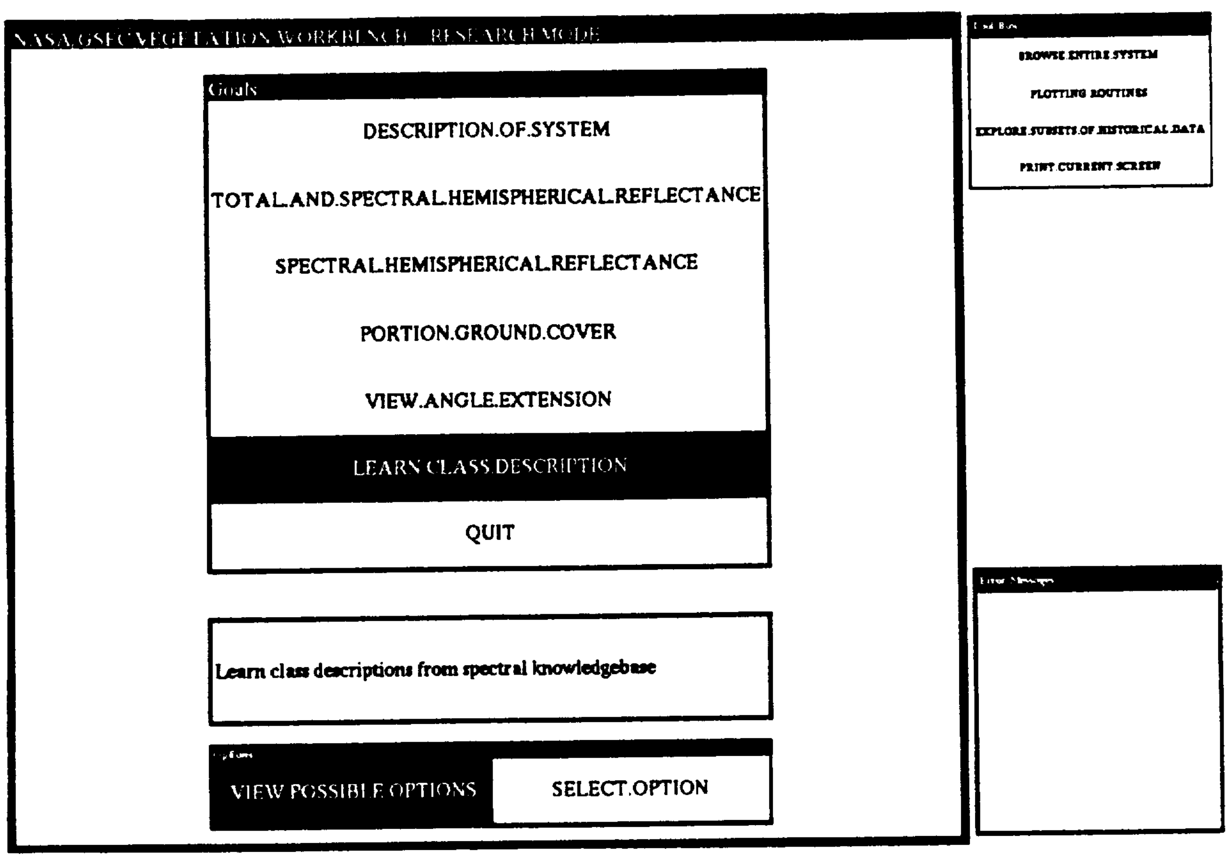


Figure 2-5

VEG "Research Mode" Top Level Menu

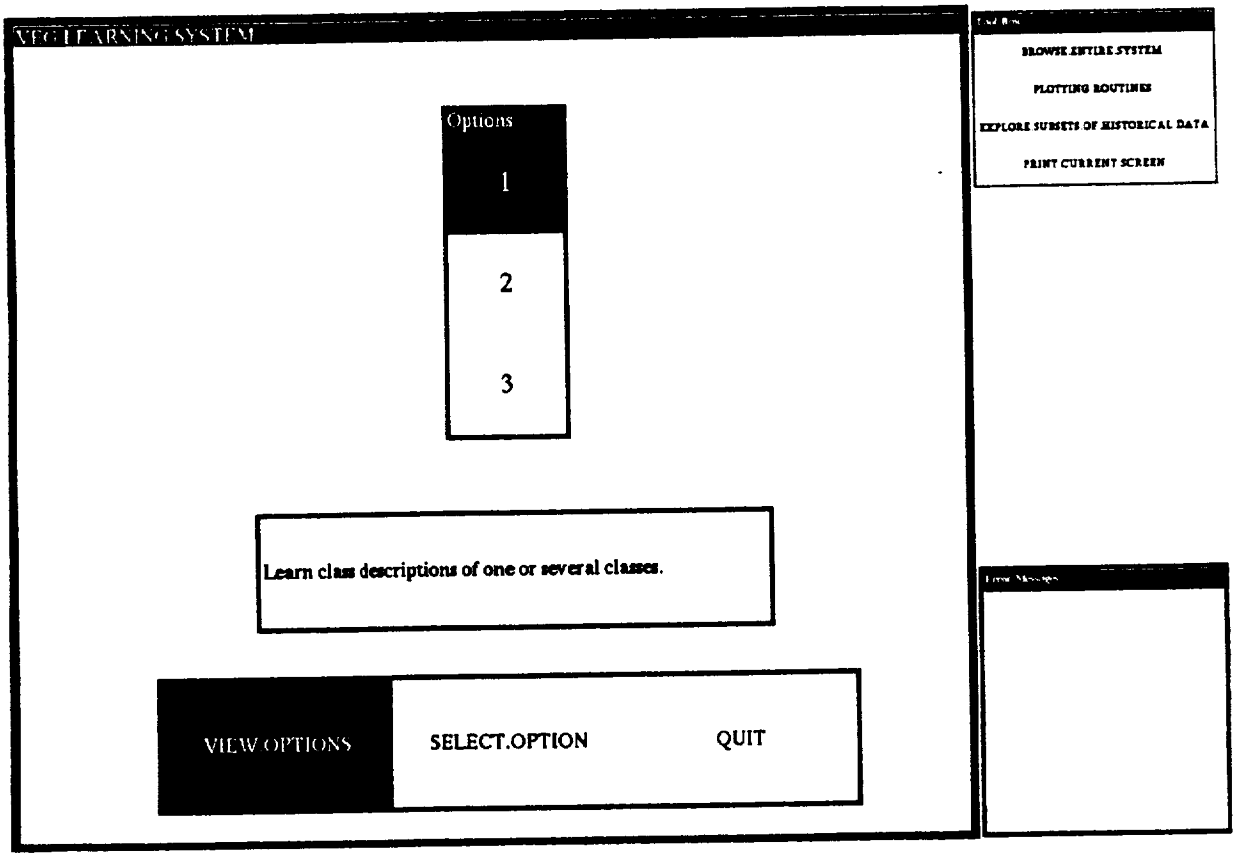


Figure 2-6

The Learning System Main Menu

SECTION 3.0

DETAILED DESCRIPTION OF THE LEARNING SYSTEM IN THE VEG "RESEARCH MODE"

In the VEG "Research Mode," the learning system presents the user with three different options. In option 1, the system learns class descriptions for one or more classes. In option 2, the system learns class descriptions for one or more classes and then uses the learned classes to classify an unknown sample. Option 3 allows the user to test the system's classification performance. In this option, the system learns class descriptions for one or more classes and then classifies the appropriate samples in the data base. The percentage of correctly classified samples is then used to summarize the degree of classification accuracy achieved by the learning system. All three options in the learning system are described in detail in this section.

3.1 OPTION 1

In option 1, the user enters data to define one or more training problems. The system then learns the class descriptions for the training problems. Finally, the results are output on the screen and the user has the option of writing the results to a file. Figure 3-1 shows the menu for Option 1. The user selects each step by mousing on the appropriate option in the learning system option 1 menu.

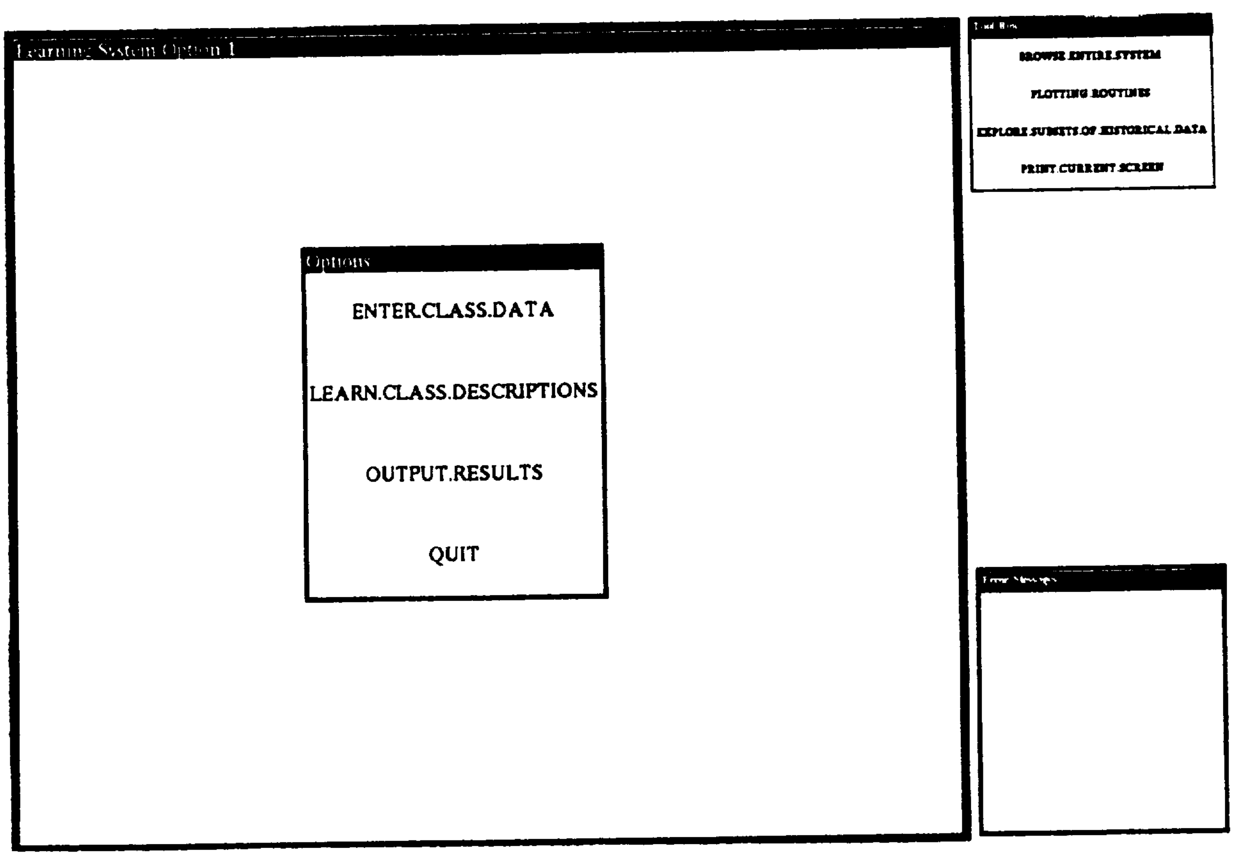


Figure 3-1

The Menu for Option 1

If the user attempts to select the menu options in the wrong order, an error message (lower right) is displayed in the Error Message box. Figure 3-2 illustrates the case when the user attempted to output the results before the system had learned the class descriptions. The steps involved in Option 1 are described in detail in this subsection.

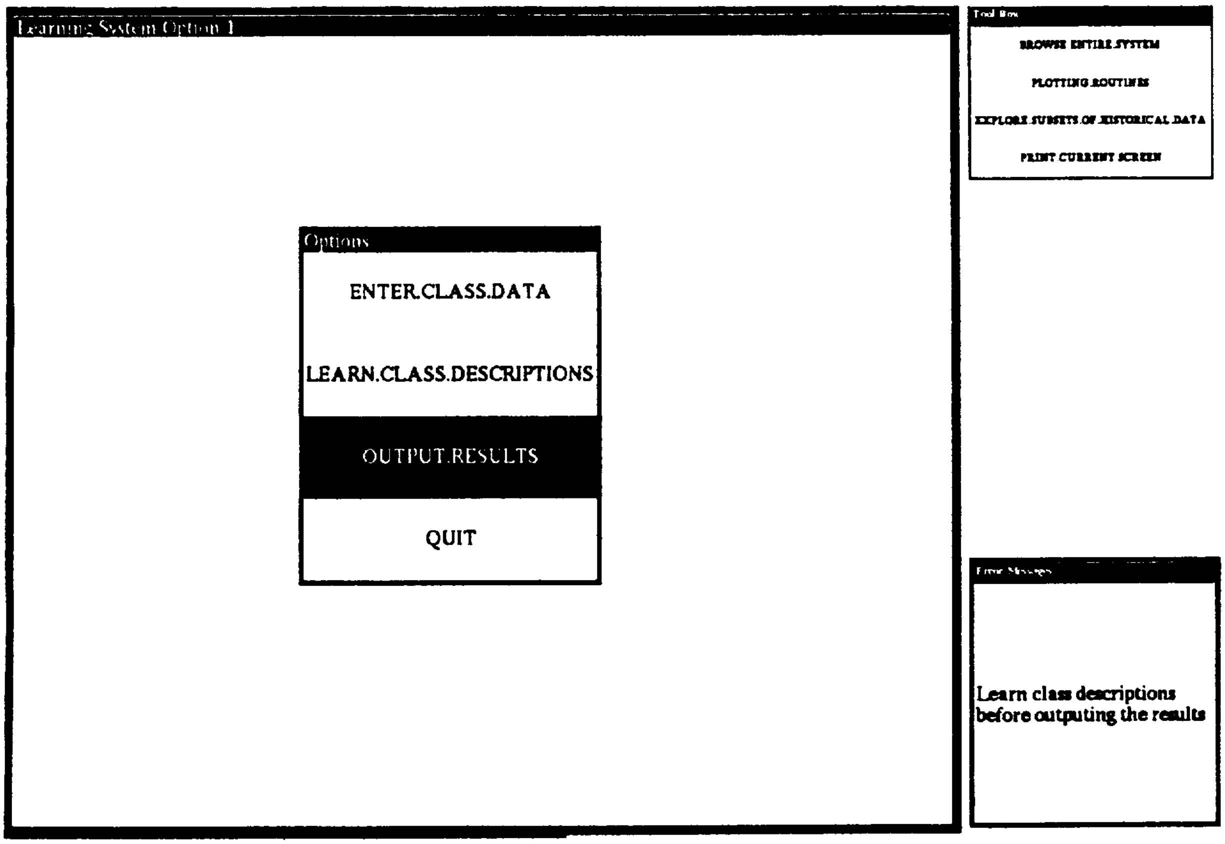


Figure 3-2

An Error Message is Displayed if Steps are Selected Out of Sequence

3.1.1 Enter Learning Data

When the user selects the option ENTER.CLASS.DATA from the option 1 menu, any previously entered training problems are deleted, and the "Enter Learning Data" interface is opened. This interface allows the user to enter data to define the training problems that are to be investigated. It is shown in Figure 3-3. The data is stored temporarily in slots in the unit ENTER.LEARNING.DATA that was shown in Figure 2-2. When the user enters a value for the solar zenith, the value is stored in the SOLAR.ZENITH slot of this unit. The ActiveValue unit ENTER.LEARNING.DATA.SOLAR.ZENITH.AV is attached to the slot. When the value of the SOLAR.ZENITH slot is changed, a Lisp function in the ActiveValue unit is executed. This function checks that the solar zenith is between 0 and 90. If the value is out of range, an error message is displayed and the value is not retained.

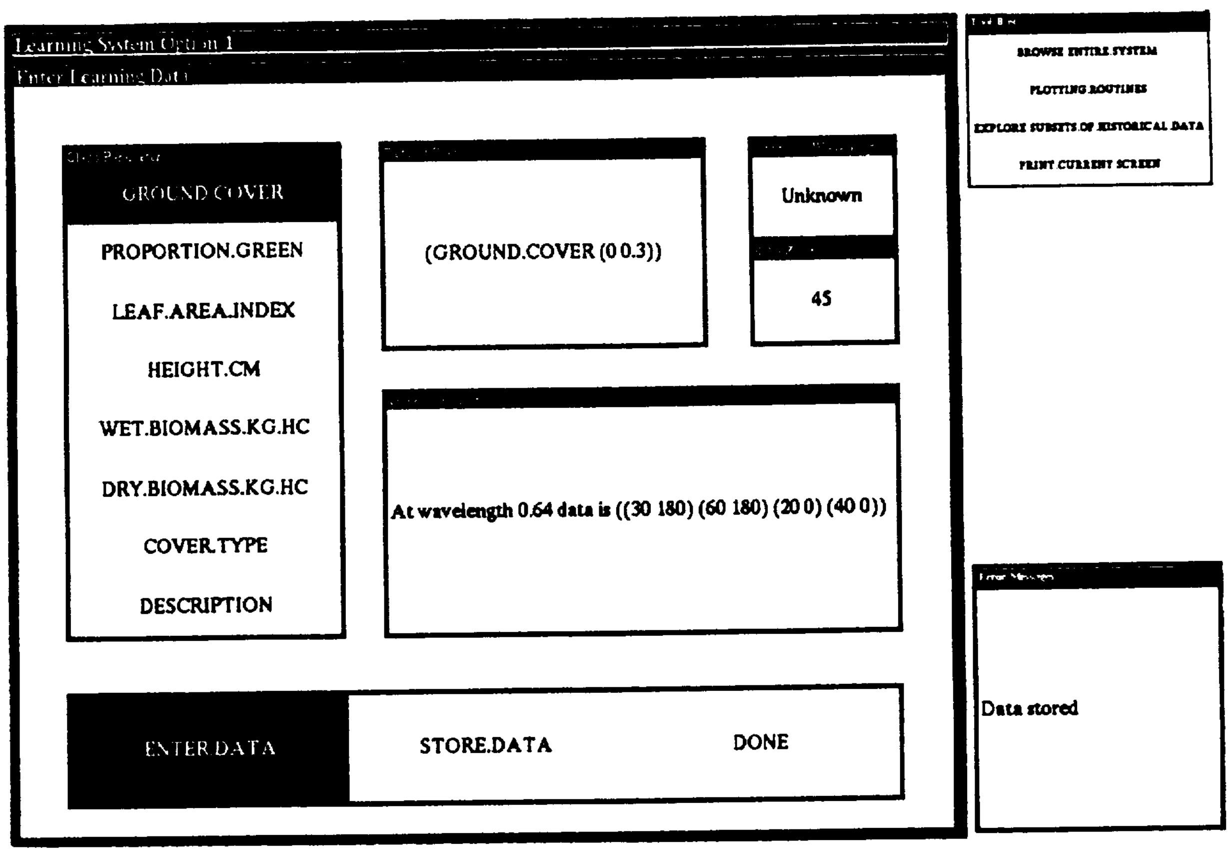


Figure 3-3

The Enter Learning Data Interface

The learning system allows the user to define a training problem consisting of more than one wavelength, with a different set of view angles at each wavelength. If the user enters a value into the box labelled "Additional Wavelength" in the interface, this value is stored and the Lisp function in the unit ENTER.LEARNING.DATA.WAVELENGTH.AV is executed. This function checks that the value for the wavelength is in range. If it is in range, an additional window is opened. This window prompts the user to enter the directional view angles for the new wavelength. The window is shown in Figure 3-4. The directional view angles are stored in the slot DIRECTIONAL.VIEW.ANGLES of the unit ENTER.LEARNING.DATA. An ActiveValue is attached to the slot in order to validate the view angle data. Once valid view angle data has been entered, the window is closed and a list such as (0.64 ((0 0)(30 180)(60 180))) is constructed from the data. This list means that at wavelength 0.64 µm, the directional view angles were (0 0), (30 180) and (60 180). This list is stored in the slot VIEW.ANGLE.DATA. A message is also constructed from this data, stored in the slot VIEW.ANGLE.DATA.MESSAGE and displayed in the box labelled "View Angle Data" in the interface, shown in Figure 3-3.

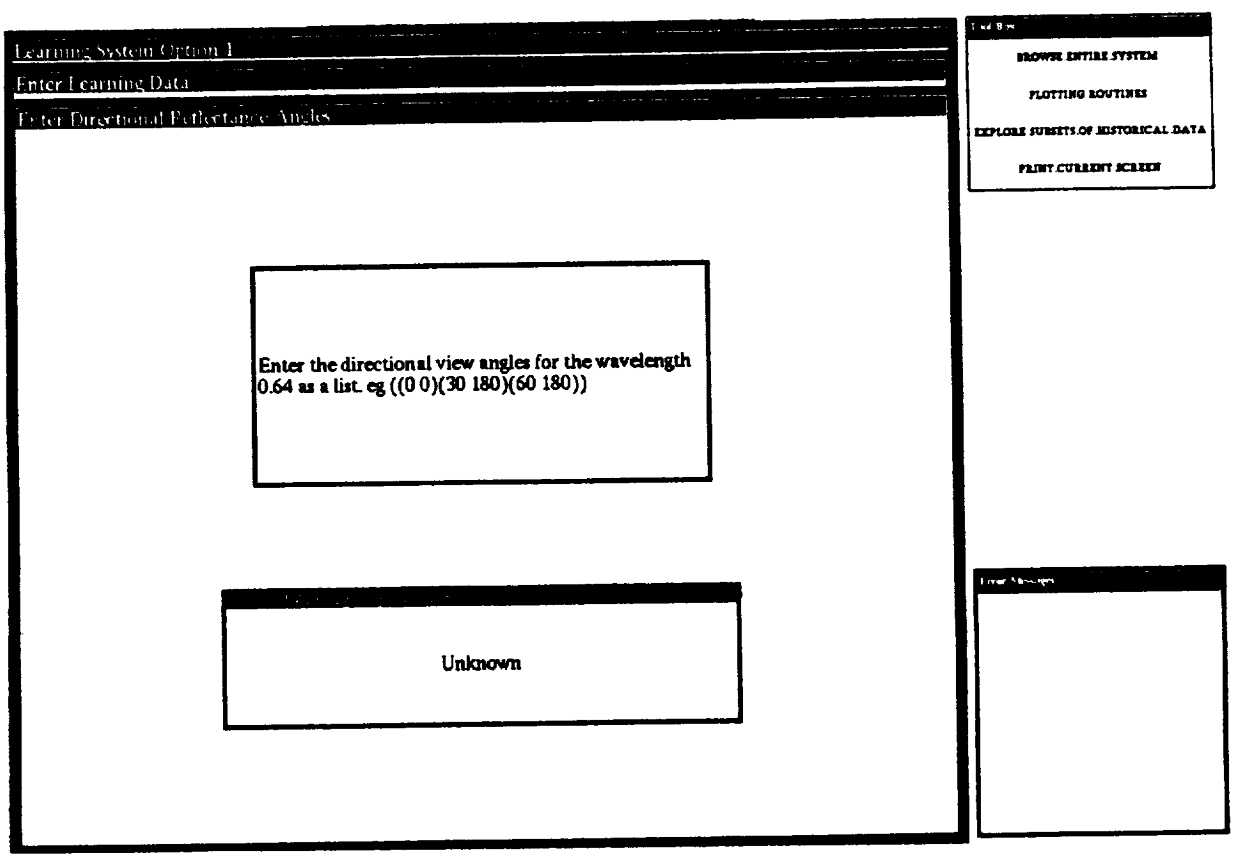


Figure 3-4

Window Through which Directional View Angles are Entered

In order to define the class whose description is to be learned, the user first selects a parameter by mousing on the Class Parameter menu. The window shown in Figure 3-5 then opens. This window prompts the user to define the class. In the case of a continuous parameter such as GROUND.COVER, the window prompts the user to enter the maximum and minimum values for the class as a list and informs the user of the range of possible input values. In the case of a discrete parameter such as DESCRIPTION, the screen displays the possible values of the parameter and prompts the user to enter the value for the parameter in the class. Figure 3-6 shows an example of this case. For example, if the parameter is DESCRIPTION, the class might be FOREST. An ActiveValue once again checks the validity of the entered data and prompts the user to enter the data again if it is invalid. Once valid data has been entered, the window is closed. A list such as (GROUND.COVER (0 0.3)) is constructed, stored in the slot CLASS and displayed on the screen as shown in Figure 3-3. This example represents the class of 0-30% ground cover. Additional class parameters can then be defined if necessary. For example, a class might be defined as forest with 70-100% ground cover.

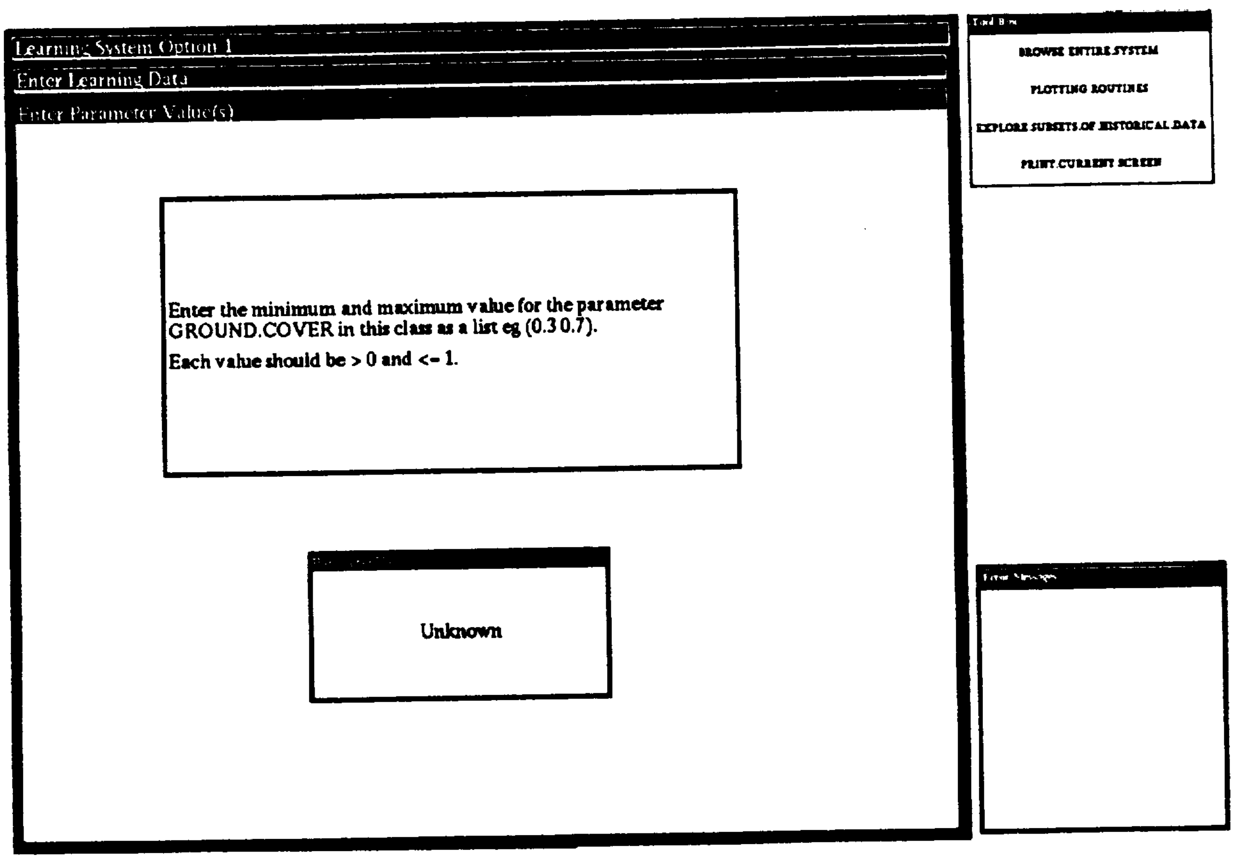


Figure 3-5

Defining a Ground Cover Class

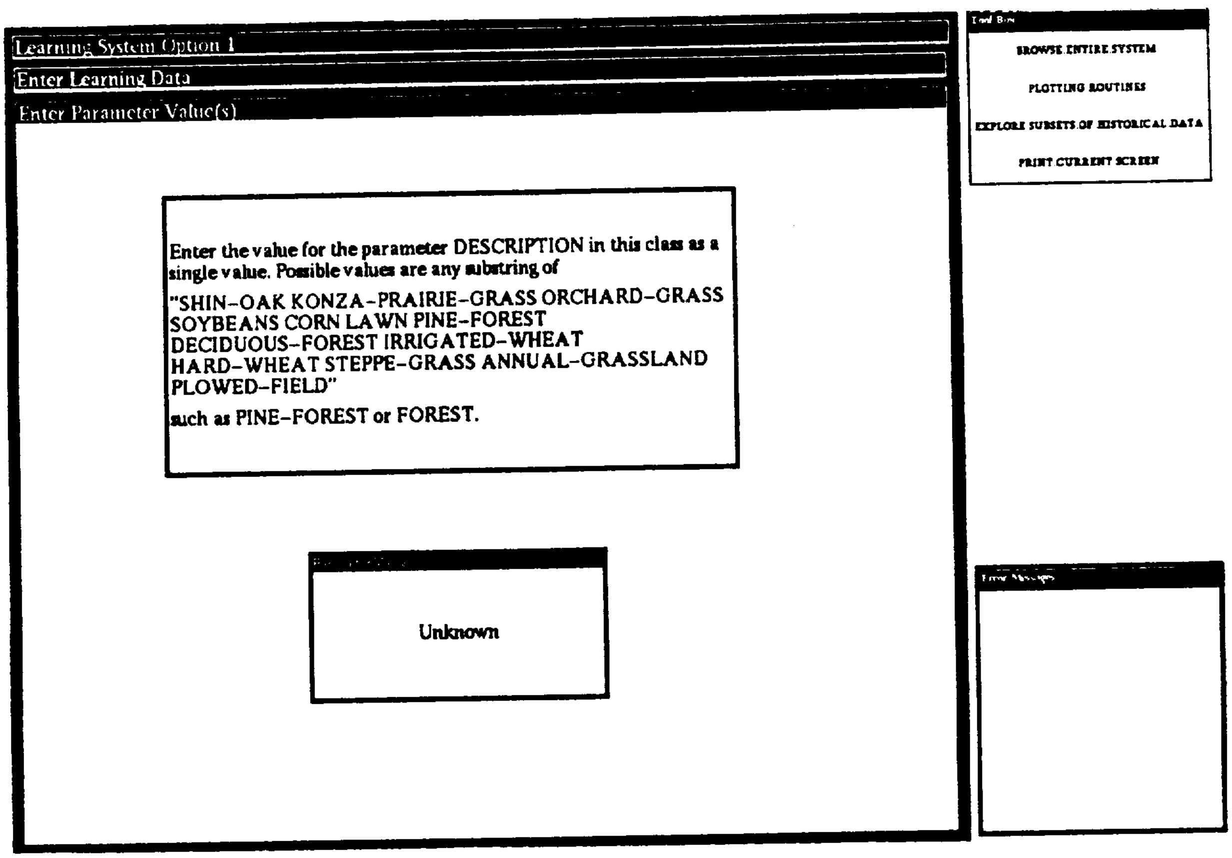


Figure 3-6

Defining a Class for a Discrete Description

When all the data for a training problem has been entered, the user can mouse on STORE.DATA. The system checks to make sure that a complete set of data has been entered. If the data is incomplete, the user is prompted to enter the missing data. If the data is complete, a new unit is created as a subclass of the unit TRAINING.DATABASES. The solar zenith, wavelengths and associated view angles and class definition for the problem are copied into the new training problem unit. The user can then select DONE if all training problems have been entered, or he/she can enter data for additional classes such as 31-70% ground cover and select the option STORE.DATA to store the data for each additional training problem. When the user selects the option "DONE," the "Enter Learning Data Interface" is closed and the menu for Option 1 appears on the screen again.

3.1.2 Learn Class Descriptions

The second step in option 1 is for the system to learn the class descriptions for the classes that were defined in the previous step. Learning the class descriptions involves several steps. First, the system uses the data base of historical cover types to set up the positive and negative training sets for each problem. Rules are run to determine the set of possible hypotheses for each problem. Next, the hypotheses are tested on the training sets to determine the discrimination score for each hypothesis. The scores are sorted in order to determine the best discriminating hypothesis for each training problem. Finally, compound hypotheses containing two or more hypotheses are constructed. These are tested in order to determine the best discriminating compound hypotheses for each training problem. The steps involved in learning class descriptions are described in detail in this subsection.

The process of learning class descriptions can take several minutes, especially when the training problem has a large number of view angles. For this reason, the message "Learning class descriptions..." is displayed at the beginning of the process. When the process has ended, this message is replaced by the message "Finished learning class description".

The first step in learning the class descriptions is to generate the training sets. The system searches the data base of historical cover types in the VEG knowledgebase and finds the cover types that best match the training problem. A cover type matches the training problem if it has data at all the wavelengths specified in the training problem, if its solar zenith is close to the training problem solar zenith and if it has a value for every parameter specified in the class definition. For example, a cover type that has no value in its ground cover slot cannot be included in either of the training sets for the class 0-30% ground cover. Once a matching cover type has been identified, the values in the slots for each parameter in the class definition are examined. If the cover type data fits the class definition, the name of the cover type is stored in the slot POS.TRAINING.SET of the training problem unit. Otherwise, it is stored in the slot NEG.TRAINING.SET. In the first search through the data base, each matching cover type whose solar zenith is within 10% of the training problem solar zenith is identified and added to the appropriate training set slot. If insufficient cover types have been found for the training sets, the search is then repeated. In the second search, matching cover types whose solar zenith is within 20% of the training problem solar zenith are identified. The process of increasing the bounds on the solar zenith and searching through the data base is continued until either the positive or negative training set exceeds the maximum permissible size, both training sets exceed the minimum permissible size or the bounds have increased to $\pm$ 100%. If, when the search ends, either training set is found to be empty, a message is displayed on the screen and the process of learning class descriptions is stopped. Figure 3-7 show this case.

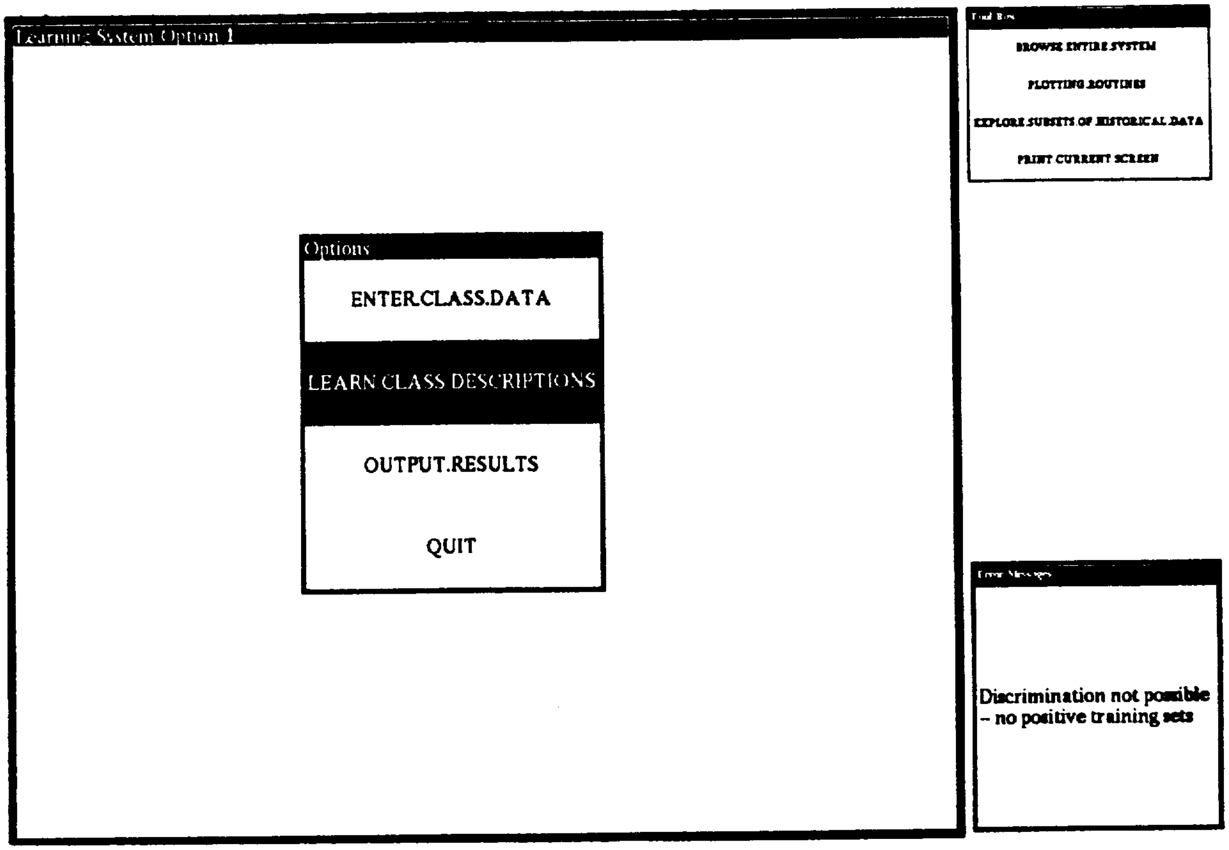
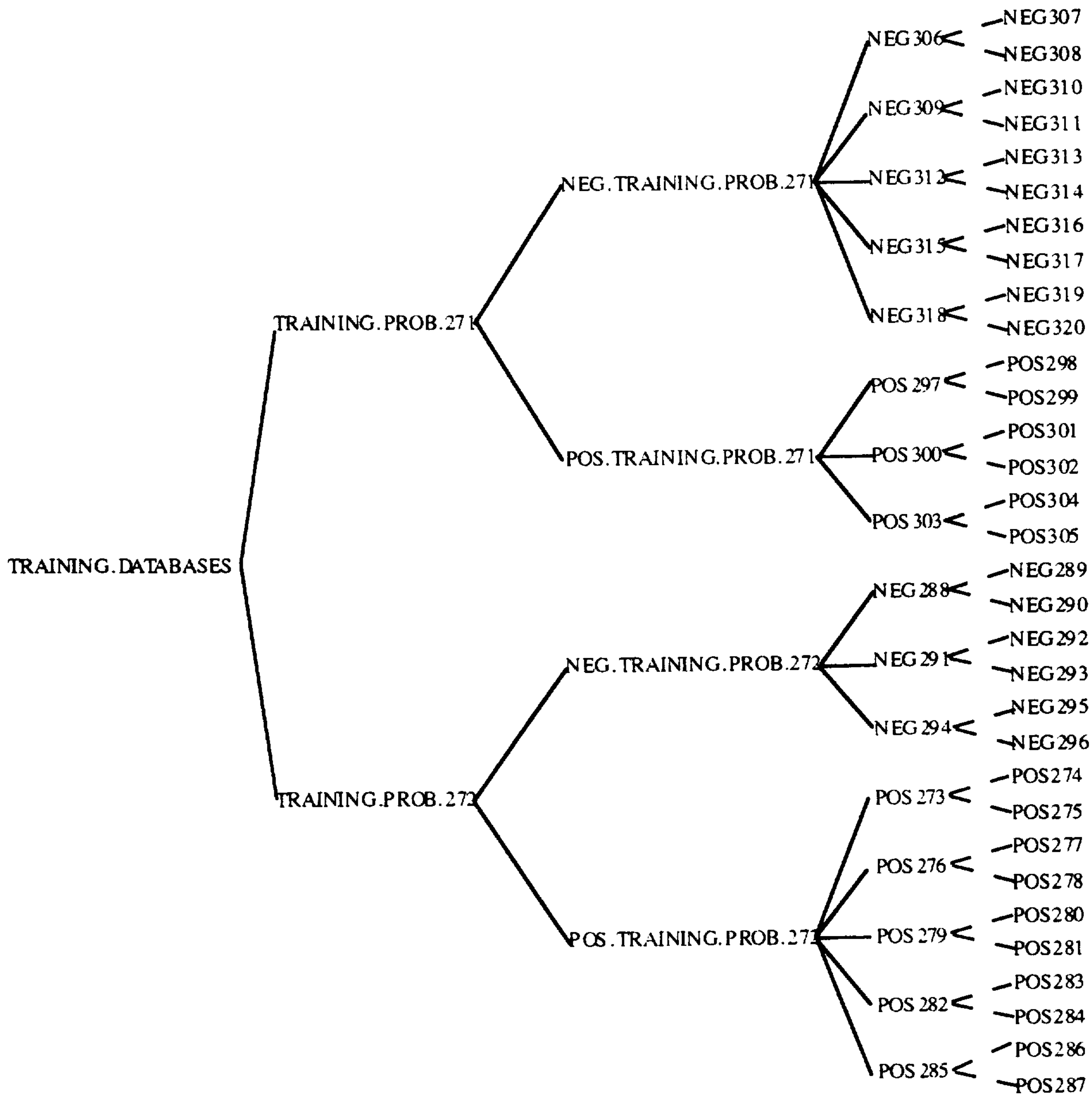


Figure 3-7

Learning Class Descriptions is Terminated if Either Training Set is Empty

Once the cover types that will make up the positive and negative training sets for each training problem have been identified, a hierarchy of units is created to hold the training set data. Figure 3-8 illustrates the hierarchy of units in the training sets for two training problems named TRAINING.PROB.271 and TRAINING.PROB.272. In this figure, the size of the training sets has been reduced so that the entire hierarchy will fit on the page. The learning system is usually run with a minimum training set size of 8 units. The class units NEG.TRAINING.PROB.271 and POS.TRAINING.PROB.271 are the parent units for the negative and positive training sets respectively for TRAINING.PROB.271. A subclass (such as the unit POS297) of the unit POS.TRAINING.PROB.271 is created corresponding to each cover type in the positive training class of the training problem and the name of the cover type is stored in this unit. In the problem illustrated in Figure 3-8, each training problem contains a set of view angles at two different wavelengths. Two member units, such as POS298 and POS299, are created for each subclass unit -- one corresponding to each wavelength in the training problem. The raw reflectance data from the cover type data at the appropriate wavelength is interpolated and extrapolated to match the view angles in the training problem at each wavelength. This is then stored in the appropriate member unit. The wavelength is also stored in this unit so that, later in the processing, the system can determine which data corresponds to each wavelength.



B921014G5

Figure 3-8

The Hierarchy of KEE Units in the Training Sets for Two Training Problems

Once the training sets have been set up, rules in the rule class LEARNING.RULES are run in order to determine the set of possible hypotheses that can be constructed for the data in each training set. The rules are shown in Figure 3-9. The left-hand side of each rule tests the view angle data. If the rule fires, the appropriate lisp function is called. Each lisp function generates hypotheses which are added to the HYPOTHESES slot of the training problem unit.

The rule LR.1 fires if the view angle data at a particular wavelength contains at least two view angles. The right-hand side of this rule calls the lisp function TRY-DIRECTION-RELATIONSHIPS which generates direction relationships for every possible pair of view angles in the data and adds these to the HYPOTHESES slot of the training problem unit. An example of a direction relationship that might be generated by this function is

(GREATER-THAN 0.64 (60 180) (30 180))

This relationship represents the hypothesis that at wavelength 0.64 μm, the reflectance at the view angle(60 180) is greater than the reflectance at view angle (30 180).

If the view angle data at a particular wavelength contains at least three view angles, the rule LR.2 fires and the lisp function TRY-MAX-MIN-RELATIONSHIPS is called. This function generates relationships such as (FIRST-MAX 0.64 (60 180)) and (FIRST-MIN 0.64 (60 180)) for each angle in the view angle data and adds them to the HYPOTHESES slot of the training problem unit. The relationship (FIRST-MIN 0.64 (60 180)) represents the hypothesis that at wavelength 0.64 μm the minimum value of the reflectance is at the view angle (60 180).

The rule LR.3 fires if the view angle data at a wavelength contains at least four view angles. The right-hand side of this rule calls the function TRY-SECOND-MAX-MIN-RELATIONSHIPS which generates relationships such as (SECOND-MAX 0.64 (60 180)) and (SECOND-MIN 0.64 (60 180)) for every angle in the view angle data. These relationships are also added to the HYPOTHESES slot of the training problem unit. The relationship (SECOND-MAX 0.64 (60 180)) represents the hypothesis that the second highest reflectance value in the data is at the view angle (60 180).

The left-hand side of rule LR.4 calls the lisp function FULL-STRING-DATA-IN-PLANE. This function returns T if the view angle data consists entirely of one full string in one azimuthal plane and NIL otherwise. Note that if the data contains more than one full string, the function returns NIL. If this rule fires, the lisp function TRY-BACKSCATTER>FORWARDSCATTER-RELATIONSHIP is called. This function adds a relationship such as (BACKSCATTER>FORWARDSCATTER 0.64) to the HYPOTHESES slot of the training problem unit. This relationship represents the hypothesis that at wavelength 0.64 μm the average reflectance value in the backscatter data is greater than the average reflectance value in the forwardscatter data.

When the forward chaining of the rules has been completed, the set of all possible separate hypotheses for each training problem has been stored in the HYPOTHESES slot of the training problem unit. If the training problem contains data at more than one wavelength, the set of hypotheses may contain hypotheses at each wavelength. The current system does not permit a single hypothesis to refer to more than one wavelength, but the system has been designed to facilitate the addition of this capability at a later date.

```
(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (> (LENGTH (SECOND ?D))
            1))
   THEN
   (LISP (TRY-DIRECTION-RELATIONSHIPS ?TRAINING.PROB ?D)))

RULE : LR1

(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (>= (LENGTH (SECOND ?D))
             3))
   THEN
   (LISP (TRY-MAX-MIN-RELATIONSHIPS ?TRAINING.PROB ?D)))

RULE : LR2

(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (>= (LENGTH (SECOND ?D))
             4))
   THEN
   (LISP (TRY-MAX-MIN-RELATIONSHIPS ?TRAINING.PROB ?D)))

RULE : LR3

(IF (SUBCLASS.OF ?TRAINING.PROB
                 TRAINING.DATABASES)
   (THE VIEW.ANGLE.DATA OF ALL ?TRAINING.PROB IS ?D)
   (LISP (FULL-STRING-DATA-IN-PLANE (SECOND ?D)))
   THEN
   (LISP
    (TRY-BACKSCATTER>FORWARDSCATTER-RELATIONSHIP ?TRAINING.PROB ?D)))

RULE : LR4
```

B921014 G6

Figure 3-9

The Rules in the Learning System

At the end of the run, the set of the best single hypotheses and the set of the best compound hypotheses for each training problem are displayed on the screen. The scientist can choose to run the system for compound hypotheses combining up to five single hypotheses (level 5). The scientist might also choose to trace the testing process. The trace provides a file consisting of the set of best compound hypotheses at each level. The purpose of this is to allow the scientist to look at predictive improvements with various combinations from the set of single hypotheses.

When the set of all possible hypotheses for each training problem has been generated, the user is asked whether the hypothesis testing should be traced. The screen shown in Figure 3-10 is displayed. If the user left clicks on "YES", the screen shown in Figure 3-11 is displayed, prompting the user to enter the name of the trace file. If the named file already exists, the user is asked to confirm that the file can be overwritten, or enter a new file name. The screen shown in Figures 3-10 and 3-11 is then closed.

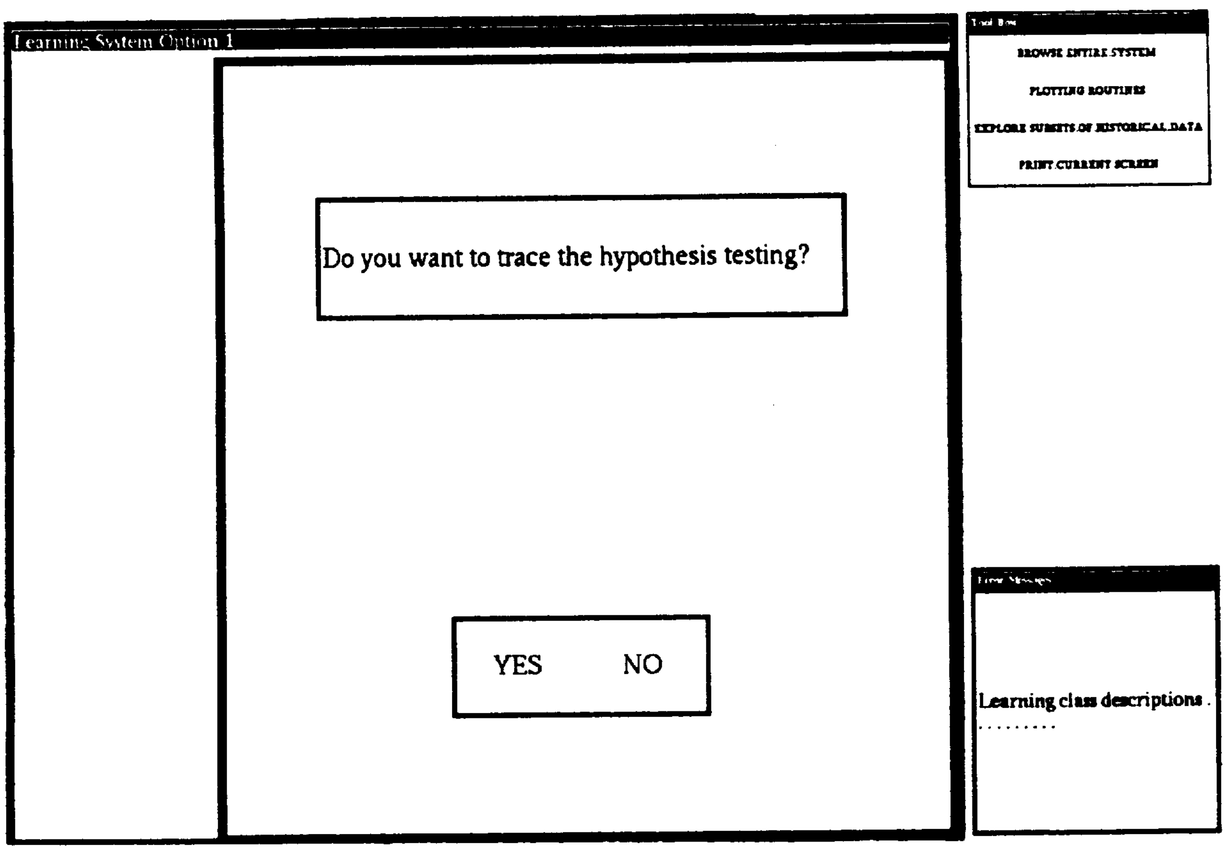


Figure 3-10

The Screen that Asks the User whether the Hypothesis Testing Should be Traced

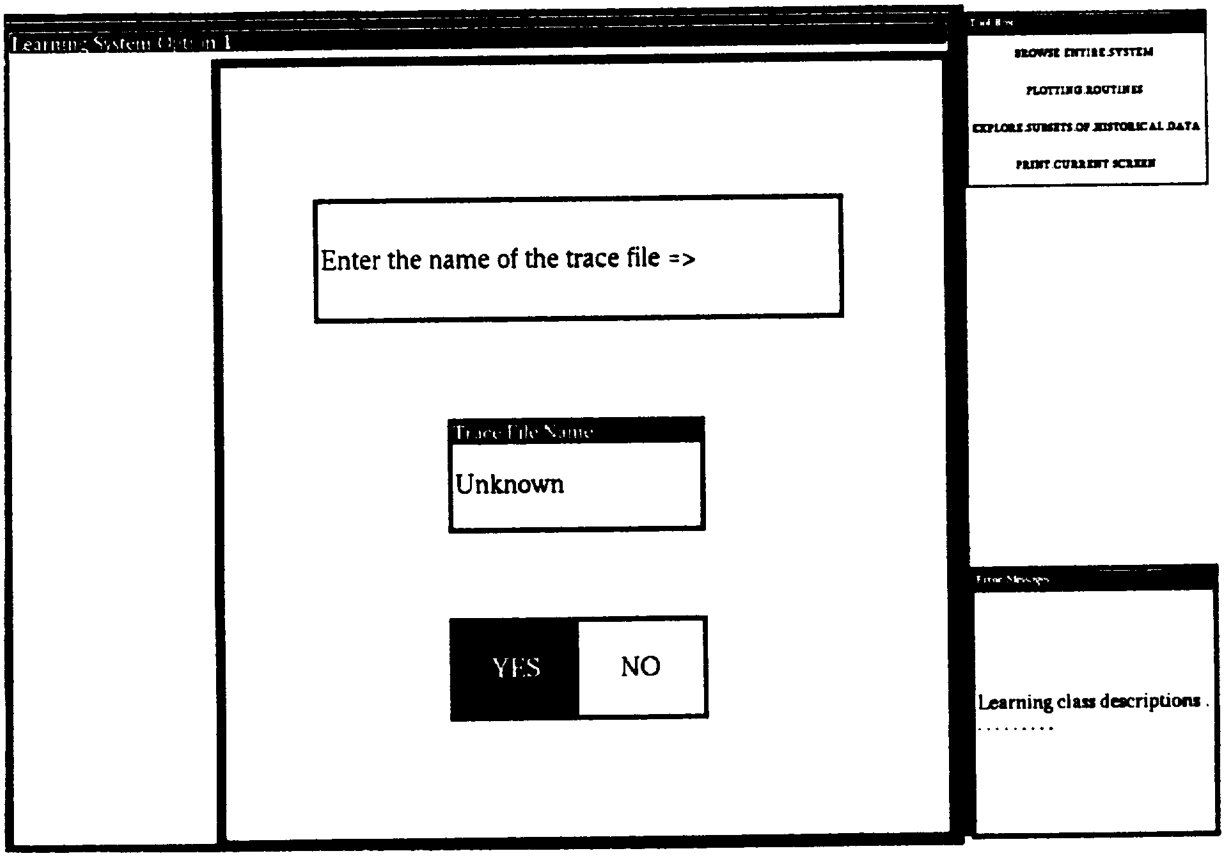


Figure 3-11

Entering the Name of the Trace File

The next step in learning the class descriptions is to determine the discrimination score for each separate hypothesis. Each hypothesis such as (GREATER-THAN 0.64 (60 180)(30 180)) is tested on each sample in the positive and negative training sets. The sample score is 1 if the hypothesis is true and 0 otherwise. The discrimination score is calculated as

$$\left[\frac{1}{p}\sum_{i=1}^{p} S_i\right] - \left[\frac{1}{n}\sum_{j=1}^{n} S_j\right] \tag{1}$$

where each sample score is S, S_i is the ith positive sample score, S_j is the jth negative sample score, p is the number of samples in the positive training set and n is the number of samples in the negative training set. Thus a discrimination score of 1 for a hypothesis represents the case where the hypothesis is true for all samples in the positive training set and false for all samples in the negative training set, i.e., 1-0=0. This represents perfect discrimination. A score of 0 is the break even point where there is no effective discrimination between the positive and negative training sets, e.g. 0-0=0 or 0.5-0.5=0 or 1-1=0. A score of less than zero for a hypothesis represents the case where the hypothesis is true for more samples in the negative training set than in the positive training set. In this case, the converse of the hypothesis would yield a positive discrimination score. For each hypothesis such as (GREATER-THAN 0.64 (60 180)(30 180)) two separate scores are calculated. The order of the elements is re-ordered and two scores such as

(((((GREATER-THAN (60 180)(30 180)) T) 0.64)) 0.4) (2)

and

(((((GREATER-THAN (60 180)(30 180)) NIL) 0.64)) -0.4) (3)

are reported. In this example, the score (((((GREATER-THAN (60 180)(30 180)) T) 0.64)) 0.4) means that the hypothesis that the reflectance at angle (60 180) is greater than the reflectance at angle (30 180) for the wavelength 0.64 μm produced a discrimination score of 0.4. The discrimination score in (2) is calculated directly by testing the hypothesis (GREATER-THAN 0.64 (60 180)(30 180)) on all the data in the positive and negative training sets. The discrimination score in (3), -0.4, is calculated as minus one multiplied by the discrimination score in (2). Scores such as (2) and (3) are calculated for each hypothesis and then pushed onto a list. This list is then put into the HYPOTHESES slot of the training problem unit, replacing the previous values in the slot. The list is also sorted by the discrimination score and the set of best scores is stored in the SCORES slot of the training problem unit.

The number of scores to be stored per wavelength is stored in the NUM.SCORES slot of the unit TRAINING.DATABASES as shown in Figure 2-4. If this slot has the value 5 and the number of wavelengths in the training problem is 2, then the best ten scores are stored in the SCORES slot of the training problem. This set of scores contains the best scores for the problem and may contain any number from zero to ten scores for each wavelength. Figure 3-12 shows an example of the value in the SCORES slot of a training problem. In this case, nine scores are for wavelength 0.64 μm and one score is for wavelength 0.82 μm. If a trace file has been named, the contents of the SCORES slot of each training problem are written to the trace file.

```
(((((GREATER-THAN  (20 0)  (40 0) T)   0. 64)) 0.5476)
(((((FIRST-MIN  (40 0)   T)  0.64))  0.5476)
(((((GREATER-THAN  (45 135) (75 135))  T) 0.82))  0.5119)
(((((FIRST-MIN  (20 0))  NIL)  0.64)) 0.4762)
(((((SECOND-MIN  (20 0)) T)  0.64))  0.4762)
(((((GREATER-THAN  (30 180)  (60 180)) T)  0.64))  0.3571)
(((((FIRST-MAX  (30 180)) T)  0.64))  0.3571)
(((((FIRST-MAX  (60 180))  NIL) 0.64)) 0.3571)
(((((SECOND-MAX  (60 180)) T)  0.64))  0.3571)
(((((SECOND-MIN  (40 0))  NIL)  0.64))  0.3333)
```

B921014G8

Figure 3-12

The Values in the SCORES Slot of a Training Problem

At this stage in the processing of the learning data, the HYPOTHESES slot of each training problem contains all the possible separate hypotheses for the problem. For each hypothesis the slot holds the discrimination score and also the positive and negative training set scores as given by the first and second terms respectively in equation (1). The SCORES slot of each training problem contains the set of the best single hypotheses for the training problem. The next step in the learning of class descriptions is to construct compound hypotheses. A compound hypothesis is composed of the combination of two or more individual hypotheses. The idea is that the interactions between various individual hypotheses may account for more variance (be more predictive) than any individual hypothesis. All the hypotheses contained in the HYPOTHESES slot of a training problem are considered as potential parts of compound hypotheses, and not just the set of the best single hypotheses stored in the SCORES slot.

Before compound hypotheses are constructed, the set of hypotheses for each training problem is reduced by removing from the HYPOTHESES slot any hypothesis that could not be combined with another hypothesis to form a compound hypothesis with a discrimination score better than the current best score. A compound hypothesis is (T) if all the separate hypotheses in the compound hypothesis are (T). Thus a compound hypothesis cannot have a discrimination score that is greater than the minimum of its components' positive training set scores. No hypothesis whose positive training set score is less than or equal to the current best score can form part of a compound hypothesis whose discrimination score is better than the current best score. For this reason, every hypothesis whose positive training set score is less than or equal to the current best score for the problem is removed from the HYPOTHESES slot.

For example, consider the hypotheses listed in Table 3-1. The best single hypothesis is hypothesis D which has a discrimination score of 0.85. If hypothesis B is combined with another hypothesis, the positive training set score for the resulting compound hypothesis cannot be greater than 0.84. Thus, whatever the negative score, the overall discrimination score for a compound hypothesis containing hypothesis B cannot be greater than 0.84. Consequently, hypothesis B cannot form part of a compound hypothesis that has a discrimination score of greater than 0.85. Hypothesis B is removed from the HYPOTHESES slot. Even though the discrimination scores of hypothesis A and hypothesis C are relatively low, the combination of these hypotheses could have a discrimination score greater than 0.85. For example, if all positive training set units that scored (T) for hypothesis A also scored (T) for hypothesis C, and no negative training set unit that scored (T) for hypothesis A also scored (T) for hypothesis C, the discrimination score of the compound hypotheses formed from hypothesis A and hypothesis C would be 0.9 - 0.0 = 0.9. The biggest benefit from combining these hypotheses together would be to reduce the negative score. Hypotheses A and C are retained in the HYPOTHESES slot even though their discrimination scores are relatively low because they could form part of a high scoring compound hypothesis.

Table 3-1

Examples of Hypothesis Scores

Hypothesis	Positive Training Set Score	Negative Training Set Score	Discrimination Score
A	0.9	0.3	0.6
B	0.84	0.01	0.83
C	0.95	0.6	0.35
D	0.89	0.04	0.85

The content of the HYPOTHESES slot is reduced further. If a hypothesis scores one for both the positive and negative training sets, it does not discriminate at all and it cannot contribute towards improved discrimination in a compound hypotheses so it is removed from the HYPOTHESES slot. If a hypothesis scores zero for the negative training set, combining it with other hypotheses cannot reduce the negative training set score and thus increase the overall discrimination score. Hypotheses that score zero for the negative training set are also removed from the HYPOTHESES slot. At the end of this step, the HYPOTHESES slot of each training problem contains only those hypotheses that could potentially be combined with other hypotheses to form a compound hypothesis with a discrimination score greater than the current best score for the problem. It should be noted that compound hypotheses can be constructed from any of the remaining hypotheses in the HYPOTHESES slot of the training problem.

The HYPOTHESES slot of a training problem may contain in excess of fifty hypotheses, even after it has been reduced. The number of possible compound hypotheses for some training problems is immense. The problem of dealing with such a large number of potential compound hypotheses was the subject of much effort. Several alternative strategies were experimented with before a successful solution to the problem was found. The first attempt was to implement a breadth-first search. Compound hypotheses that had been investigated were stored on an explored list. Each time a compound hypothesis was investigated, all possible combinations of the hypothesis and other hypotheses from the HYPOTHESES slot were constructed and stored on an unexplored list. Checks were made to prevent duplication of compound hypotheses on the unexplored list and to prevent the same hypothesis being investigated more than once. This involved sorting all the separate hypotheses within a compound hypothesis into a standard order so that comparisons could be made. This strategy was rejected because it was slow and the system frequently crashed because it ran out of memory. The second attempt was to implement a heuristic search with a depth bound. This strategy was also very limiting since, in many cases, the system ran out of memory even before the search of compound hypotheses consisting of only two separate hypotheses was completed.

It was decided to try a completely different approach. The new approach was a "Generate and Test" approach. This strategy involved testing some compound hypotheses that the previous version would have recognized as not being possible solutions. However, the new approach had a much reduced memory requirement compared with the previous version because explored and unexplored lists were not kept. All possible compound hypotheses were generated systematically in a way that made duplication impossible. This also greatly reduced the processing time because the separate hypotheses did not need to be sorted within a compound hypothesis and no checks for duplication needed to be made. This approach was successful. Tests have shown that learning class descriptions can take as long as twenty three hours, but no test has failed because of memory problems. The user has the option of interrupting the learning at any time and using the intermediate results. The implementation of this strategy will now be described in detail.

After the number of hypotheses in the HYPOTHESES slot of each training problem has been reduced, a button labelled "INTERRUPT" is displayed on the screen as shown in Figure 3-13. The user can left click on this box at any time during the processing of the compound hypotheses in order to interrupt the processing. If the processing is interrupted, the best results obtained up to the time of interruption are retained for later use.

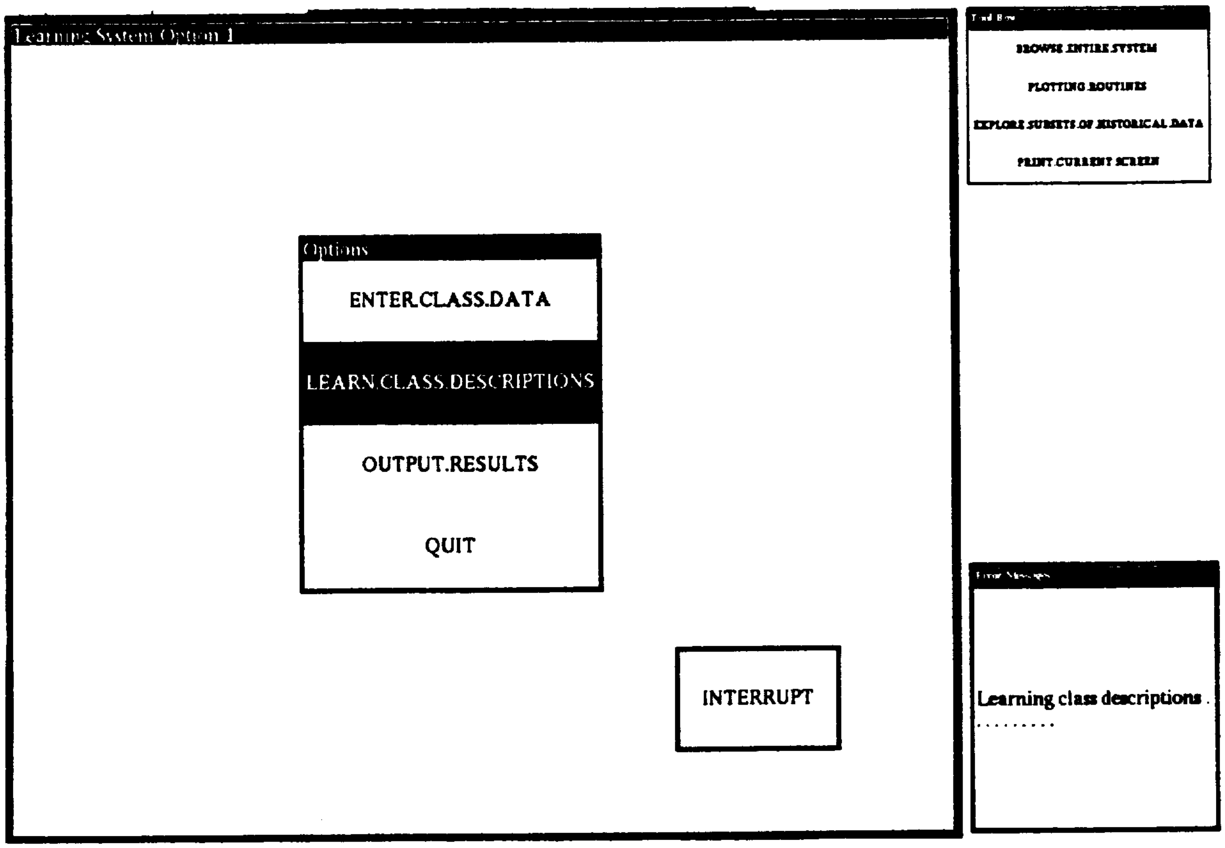


Figure 3-13

The INTERRUPT Button

For each training problem, the best single hypothesis score is stored in the slot PREVIOUS.BEST.SCORE of the training problem unit. Unless the best score for the problem is 1.0, the learning system next constructs and calculates the discrimination scores for all the hypotheses at level 2 (i.e. compound hypotheses each containing two separate hypotheses). Each compound hypothesis is generated and its discrimination score is immediately calculated. If the score is better than the previous best discrimination score for the problem, the hypothesis and score are placed in the BEST.SCORE slot of the problem replacing the previous values in the slot. If the score equals the best discrimination score for the problem and is better than the best discrimination score for single hypotheses for the problem, the compound hypothesis and score are added to the BEST.SCORE slot of the problem. If the best discrimination score for a single hypothesis is equaled by a compound hypothesis, the compound hypothesis and score are not added to the BEST.SCORE slot. Once the level 2 search has been completed, the contents of the BEST.SCORE slot of each training problem are written to the trace file if tracing has been selected. Then each training problem is considered again in turn. A check is made to determine whether a 10% improvement in the best discrimination score for the problem was achieved by searching at level 2 compared with considering only the single hypotheses. If this level of improvement has occurred, the level 3 hypotheses for the problem are constructed and processed. If a 10% improvement has not occurred, no further compound hypotheses are constructed or tested for that training problem. Constructing and testing hypotheses a level at a time continues until insufficient improvement is achieved or the user interrupts the processing by left clicking on the "INTERRUPT" button. The user can also control the level of compound hypothesis testing by setting a depth bound. Depth in this context refers to the number of hypotheses in the combination. This bound is stored in the slot SEARCH.DEPTH of the unit LEARNING.METHODS. If a trace file is in use, the time that the learning ended is written to the trace file and the file is closed. The message "Finished learning class descriptions" is displayed on the screen.

This step completes the learning of class descriptions. At the end of this step the SCORES slot of each training problem unit contains the best single hypotheses that discriminate the training class and the BEST.SCORE slot of each training problem unit contains the set of best compound hypotheses that discriminate the training class.

3.1.3 Output Results

The final step in learning system Option 1 is to output the results. When the user selects this step, the screen shown in Figure 3-14 is displayed. The solar zenith angle, wavelength(s) with associated view angles and the class definition for the first class to be reported are displayed in appropriately labelled boxes. In a text image labelled "Results" the names of the cover types in the positive and negative training sets are displayed, along with the best discrimination hypothesis scores for the class. If more than one training problem has been investigated, the user can view the next or the previous class by mousing on the appropriate options on the menu at the bottom of the screen.

As Task B of the current contract, an interface was implemented. This interface enabled the results of VEG to be written to a file. The interface worked for all the VEG subgoals except LEARN.CLASS.DESCRIPTIONS. This interface has been extended to include the results from the learning system.

When the user selects "QUIT" from the menu shown in Figure 3-14, another screen is opened. The user is asked whether or not the results should be written to a file. Left clicking on "NO" returns the user to the option 1 menu. Otherwise, the user is prompted to enter the name of the output file. A check is made to see if the file exists. If the file already exists, the user has the option of entering a new file name or overwriting the existing file. When an acceptable file name has been entered, the screen shown in Figure 3-15 is opened. This screen enables the user to select the parameters to be written to the file or select a template for a standard format. Each time the user left clicks on a parameter in the "Output Parameters" menu, the parameter is added to the slot FORMAT.LIST of the unit OUTPUT.LEARNING.RESULTS. When the user left clicks on "DONE", the data is written to the file and the option 1 menu is once again displayed. Instead of selecting the output parameters separately, the user can select a standard template. When the user left clicks on "STANDARD.TEMPLATE", the screen shown in Figure 3-16 is opened. Left clicking on the required template selects it. As soon as a template has been selected, the data is written to the file and the user is returned to the option 1 menu. Selecting "QUIT" from this menu returns the user to the main menu for the learning system.

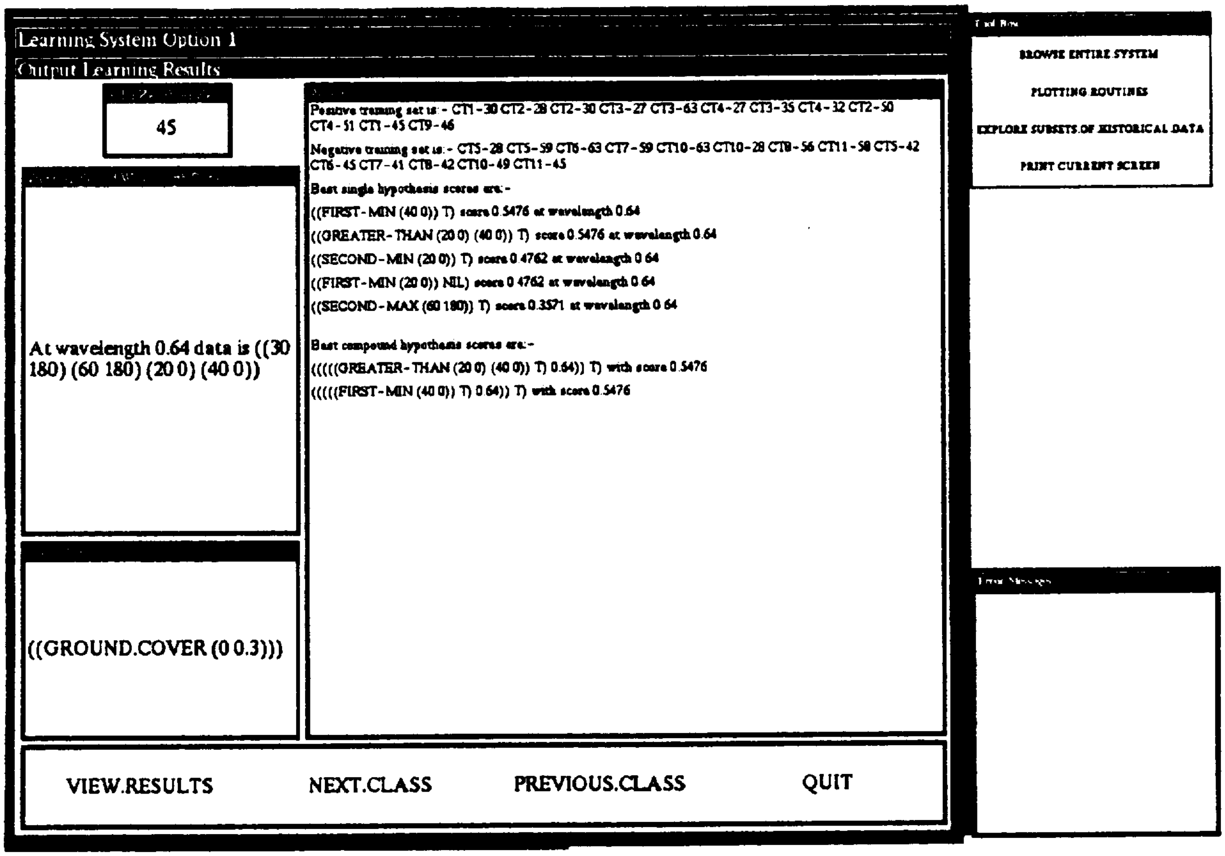


Figure 3-14

The Output Screen for Option 1 of the Learning System

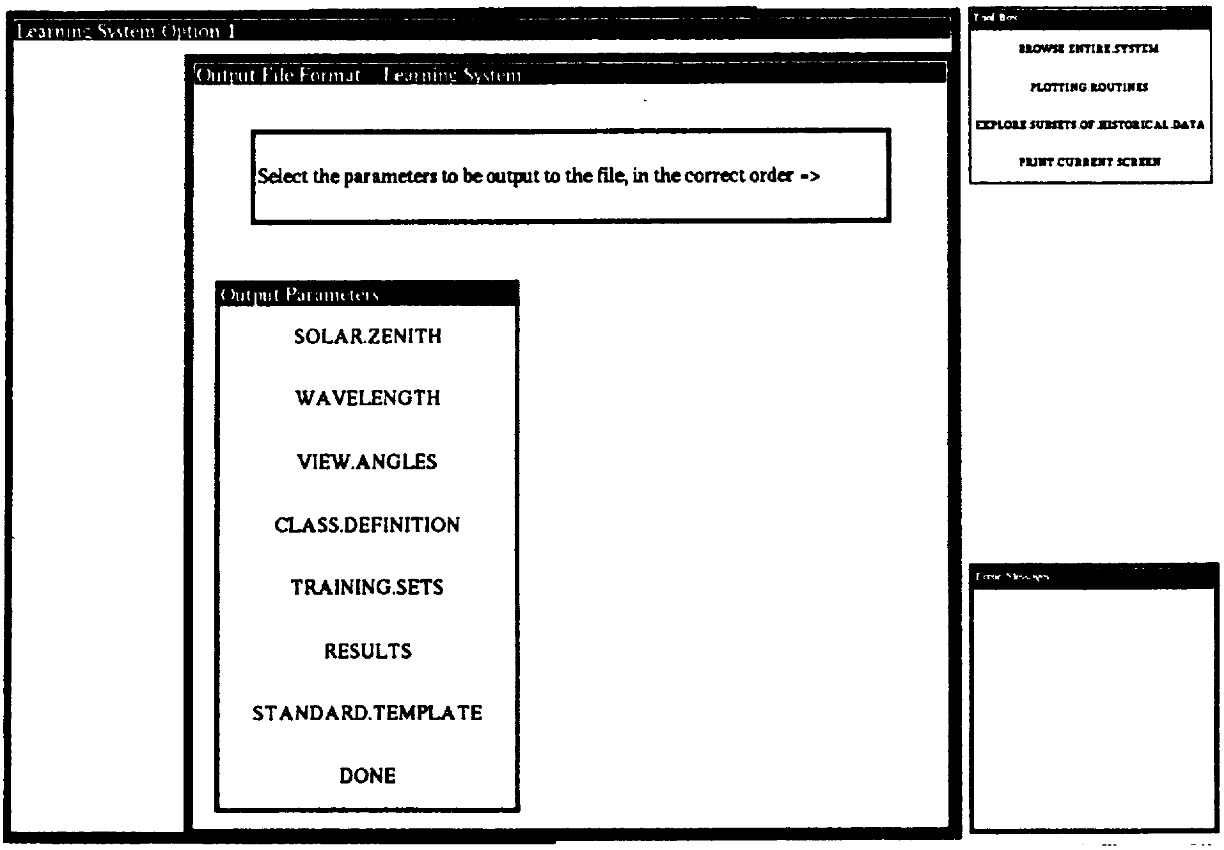


Figure 3-15

The Output File Format Menu for the Learning System

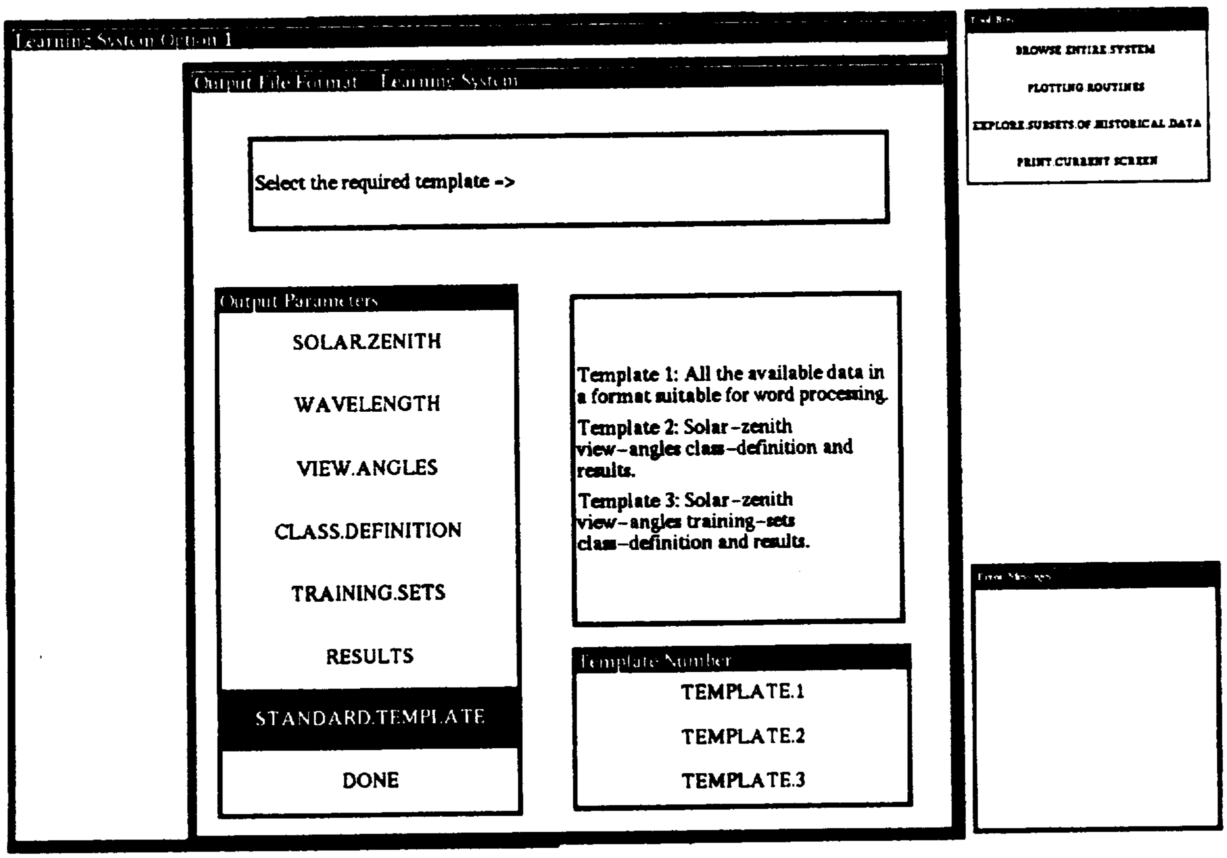


Figure 3-16

The Screen that Enables the User to Select a Standard Template

3.2 OPTION 2

Referring back to Figure 2-6, option 2 is used to have the system learn class descriptions for one or more classes and then use the classes to classify an unknown sample. In this option, the user either enters a new sample or selects a sample from the set of examples already stored in VEG. The user then enters one or more training classes. This class data is used together with the solar zenith, wavelength(s) and view angles from the sample to define the training problem(s). The system learns the class descriptions for the training problem(s) as in option 1. The sample is then classified to determine the degree to which it matches each of the defined classes, and to identify the class that it best fits. Figure 3-17 shows the menu for option 2. The steps involved in option 2 are described in detail in this subsection.

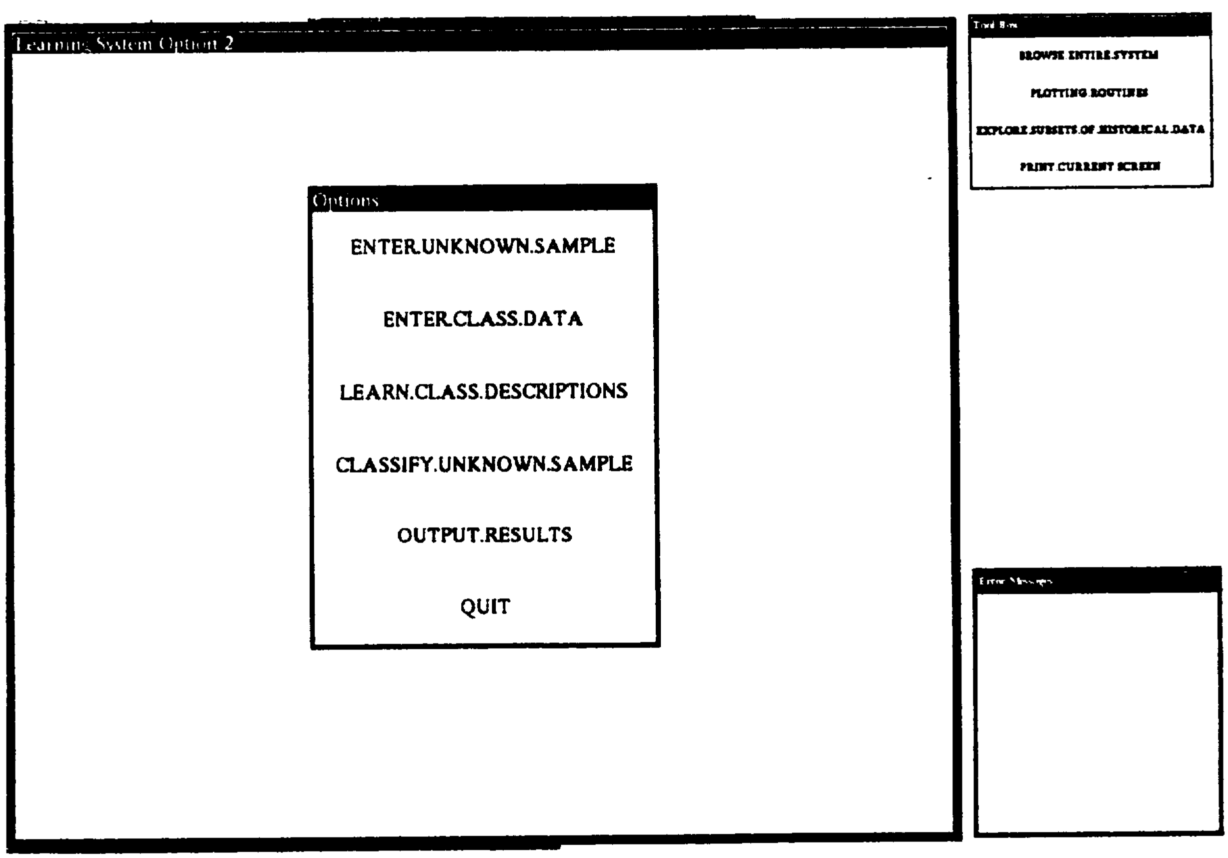


Figure 3-17

The Menu For Option 2

3.2.1 Enter Unknown Sample

When the user selects the option ENTER.UNKNOWN.SAMPLE from the option 2 menu, the Enter Data Interface that was originally developed for the Estimate Spectral Hemispherical Reflectance subgoal of VEG is opened. The first screen allows the user to choose between entering original data and selecting an example data set already stored in VEG. If the user selects the first of these options, the screen shown in Figure 3-18 is displayed. This screen enables the user to enter and store the data for the original sample that is to be investigated. If the user elects to select an example data set, a different screen is displayed. Descriptions of the available sets of sample data are displayed on this screen and the user can mouse on the sample name to select the required sample. The names of the units containing the newly entered or selected sample data are stored in slots of the unit ESTIMATE.HEMISPHERICAL.REFLECTANCE. It should be noted that this option uses objects from the underlying VEG layer.

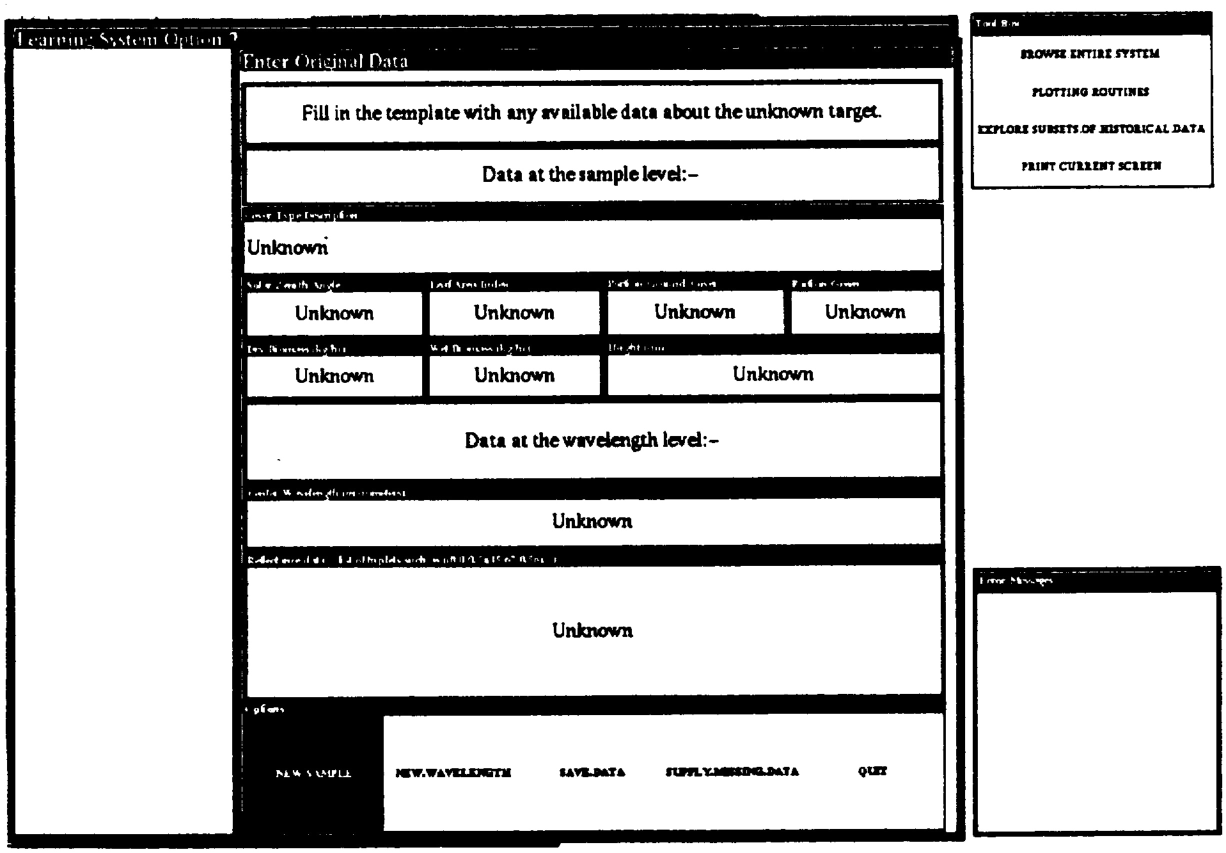


Figure 3-18

The Screen Through Which Original Sample Data is Entered

3.2.2 Enter Learning Data

When the user selects ENTER.LEARNING.DATA from the option 2 menu, the Lisp function MATCH-CLASSES-TO-SAMPLE is called. This function checks that the step ENTER.UNKNOWN.SAMPLE has been completed. It then deletes any previously entered training problems and initializes the values in the slots of the unit ENTER.LEARNING.DATA (Figure 2-2). The values of the directional view angles, solar zenith and wavelength from the sample are copied into the appropriate slots of the unit ENTER.LEARNING.DATA so that the training problem matches the unknown sample that is being classified. The mouse left functions of the boxes labelled "Additional Wavelength", "Solar Zenith" and "View Angle Data" in the Enter Learning Data interface are disabled to prevent the user changing any of the values that have been set to match the unknown sample. The Enter Learning Data interface is then opened. As shown in Figure 3-19, the solar zenith and view angle data are displayed. In order to complete the definition of the training problems, the user must then enter the class parameters and values and store the data as in option 1 (described in section 3.1.1).

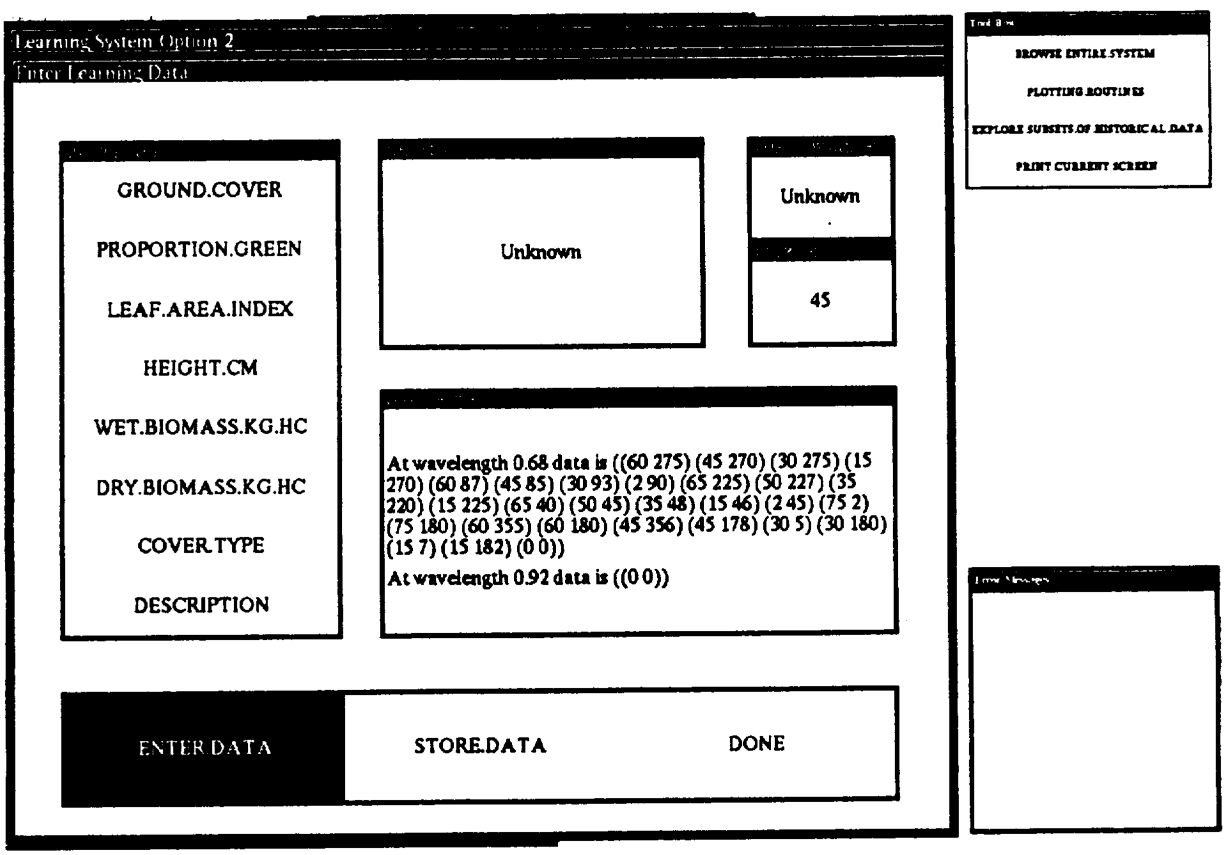


Figure 3-19

The Enter Learning Data Interface For Option 2

3.2.3 Learn Class Descriptions

The step LEARN.CLASS.DESCRIPTIONS in option 2 is identical to the same step in option 1. This step was described in detail in section 3.1.2.

3.2.4 Classify Unknown Sample

The class descriptions are used to classify the unknown sample by testing to determine which class has the most evidence that the sample belongs to that class. The relative score of evidence that is computed to determine whether a sample belongs to a particular class is calculated as :

$$\text{Score} = 1 - \frac{E_{Opp}}{E_{Sup}} \tag{4}$$

or

$$\text{Score} = -1 + \frac{E_{Sup}}{E_{Opp}} \tag{5}$$

where E_{Sup} is the sum of the discriminating scores of instances in the class solution that are true for the sample data, and E_{Opp} is the sum of discriminating scores that are false for the sample data. Equation (4) is used if the supporting evidence is greater than the opposing evidence. Otherwise, equation (5) is used.

When the user selects CLASSIFY.UNKNOWN.SAMPLE from the option 2 menu, the Lisp function GET-SCORES-FOR-SAMPLE is called. This function first makes sure that the prerequisite steps have been completed. A list of wavelengths and reflectance data in the unknown sample is constructed. The function CLASSIFY-SAMPLE is called with this list as the argument. Within the function CLASSIFY-SAMPLE, local variables BEST-SCORE and BEST-CLASS are initialized to the values -2 and NIL, respectively. CLASSIFY-SAMPLE iterates through the training problems. For each problem, the score given by equation (4) or (5) is calculated and stored in the slot SAMPLE.SCORE of the training problem. If the score is greater than the value stored in BEST-SCORE, the value of BEST-SCORE is reset to the score and the value of BEST-CLASS is set to the name of the training problem unit that yielded the score. The function CLASSIFY-SAMPLE returns the name of the training problem unit that gave the highest score according to equation (4) or (5). The slot BEST.CLASS of this unit is set to T to indicate that it is the class that has the most evidence that the unknown sample belongs to that class.

3.2.5 Output Results

The final step in option 2 is to output the results. This step is the same as was described for option 1 in section 3.1.3 except that additional results are output in option 2. For each training problem, the score for the unknown sample for that class is also displayed in the box labelled "Results." In addition, when the results for the best class for the sample are displayed, a message is added to the box labelled "Results" stating that the class being displayed is the best class for the sample.

Figures 3-20, 3-21 and 3-22 show the results of classifying the unknown sample stored in the units SAMPLE3, W5 and W6 in VEG. Three possible classes were investigated. These were (DESCRIPTION WHEAT), (DESCRIPTION GRASS) and (DESCRIPTION FOREST). The support scores for the unknown sample in these classes were -1.0000, -1.0000 and 0.0000, respectively. The system correctly reported that the class (DESCRIPTION FOREST) was the best class for the unknown sample. This test is discussed in detail in Section 5.2.

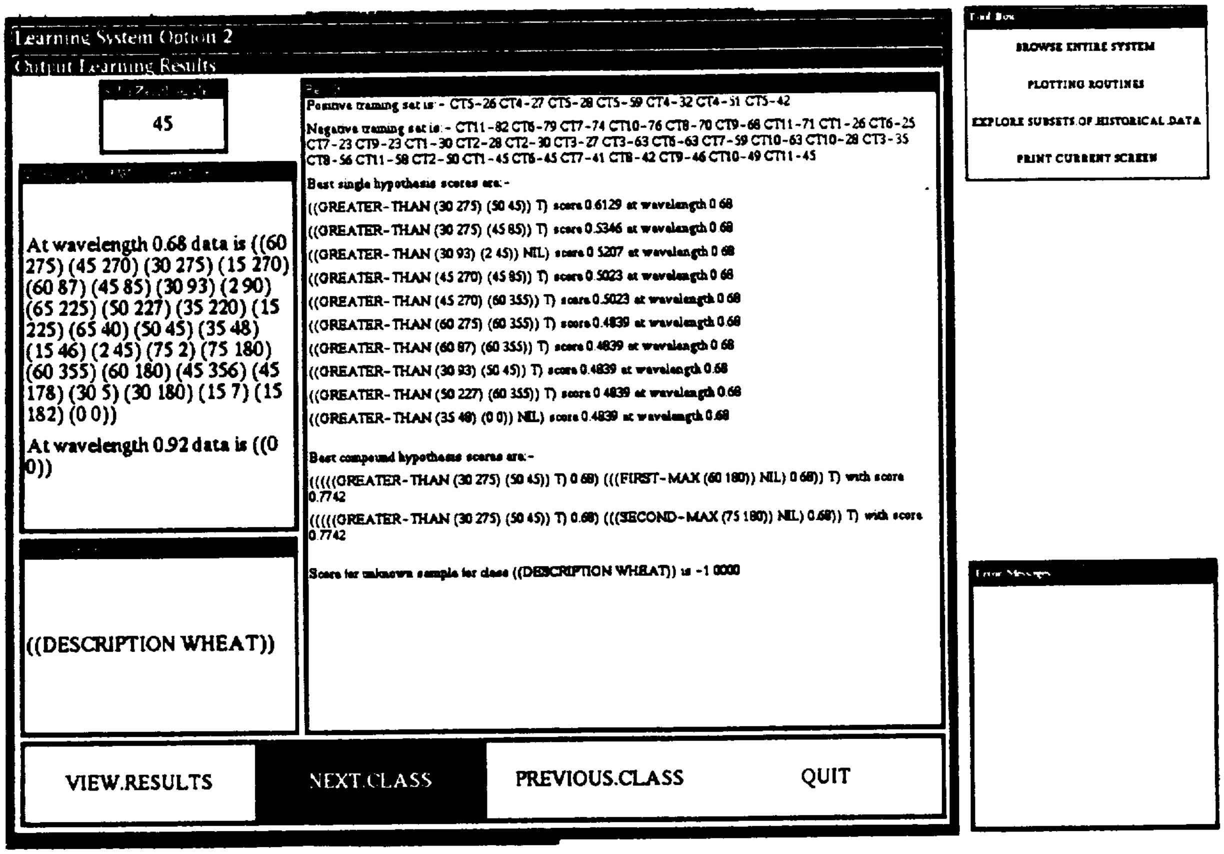


Figure 3-20

Results for SAMPLE3 and the Class (DESCRIPTION WHEAT)

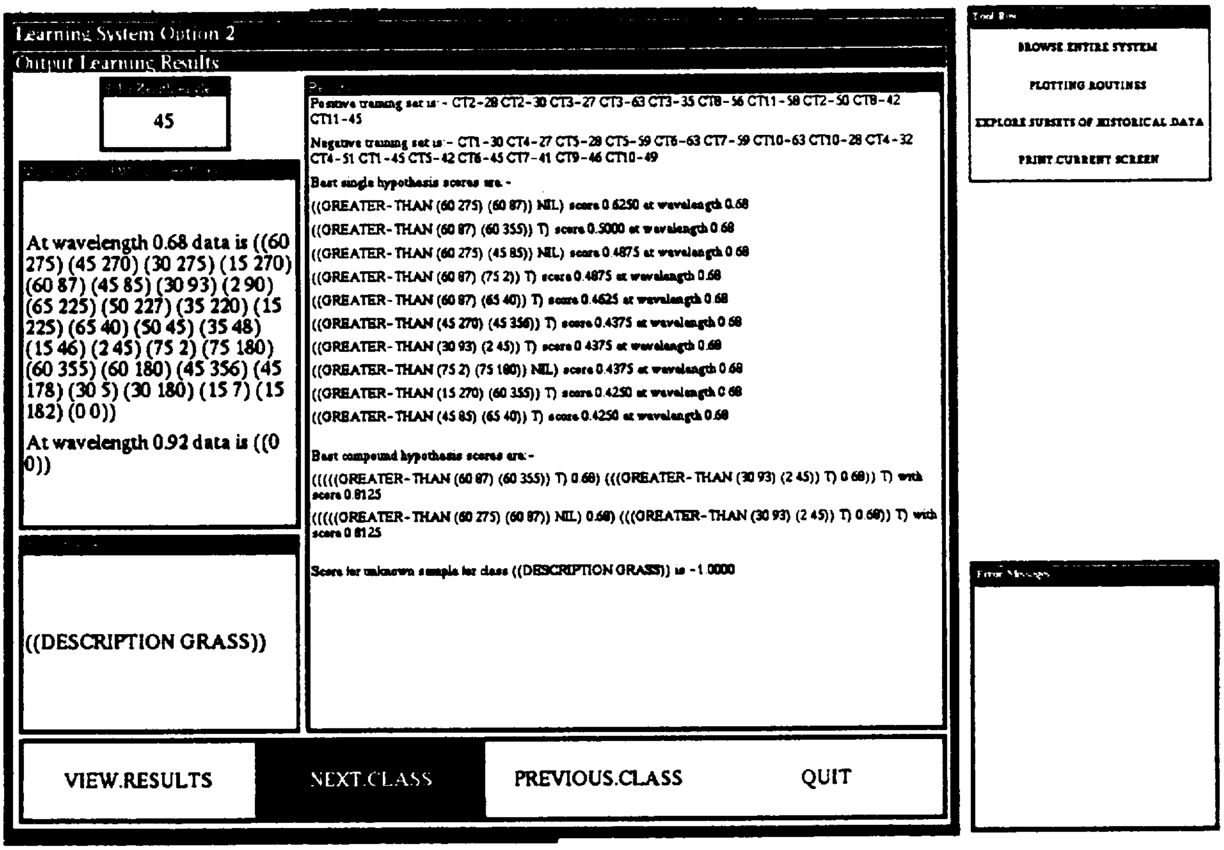


Figure 3-21

Results for SAMPLE3 and the Class (DESCRIPTION GRASS)

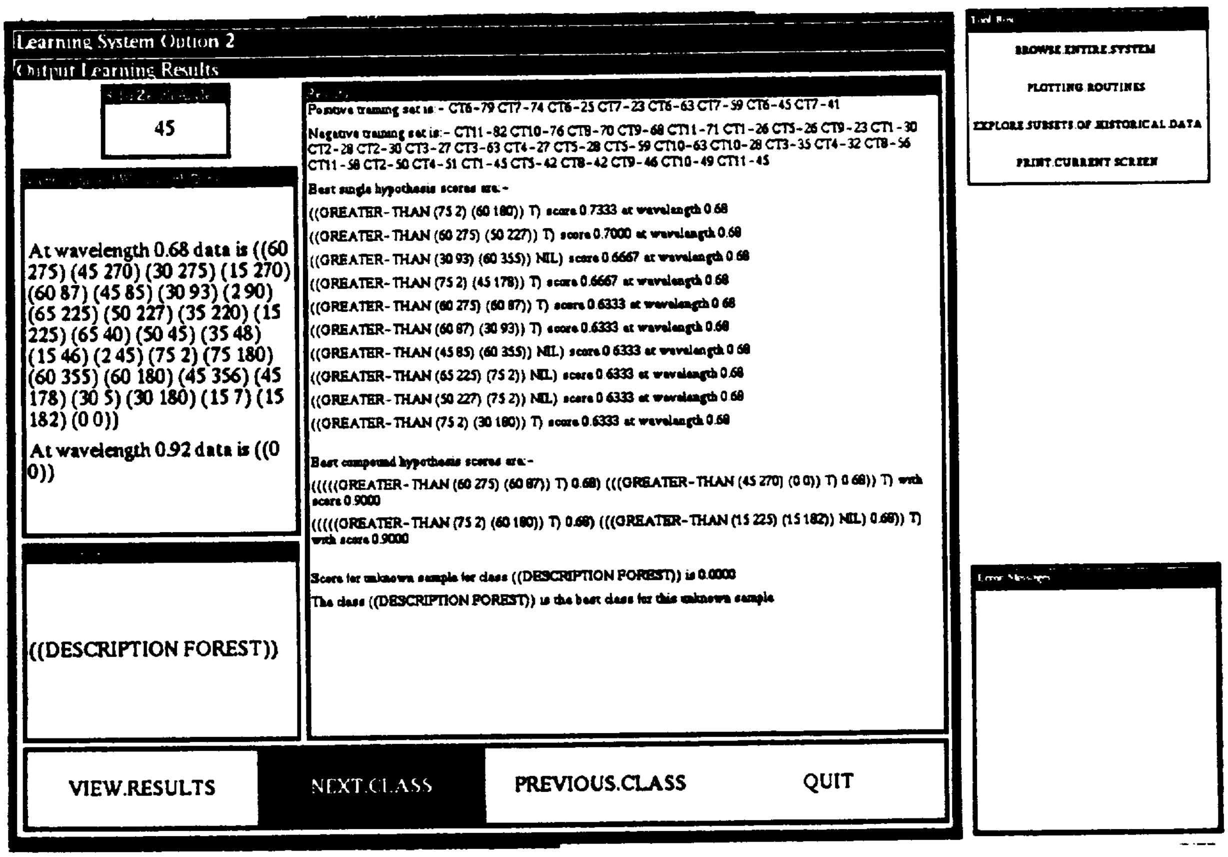


Figure 3-22

Results for SAMPLE3 and the Class (DESCRIPTION FOREST)

3.3 OPTION 3

Referring to Figure 2-6, the purpose of option 3 is to allow the user to test or validate the new classes defined using option 1 or 2. User defined classification systems can also be tested using this option. Class data is entered for one or more classes. The system then learns the class descriptions as in options 1 and 2. The cover types in the positive training set for each class are then classified according to the learned classes. The percentage of correctly classified cover types forms a measure of the classification accuracy achieved by the learning system. Figure 3-23 shows the menu for option 3.

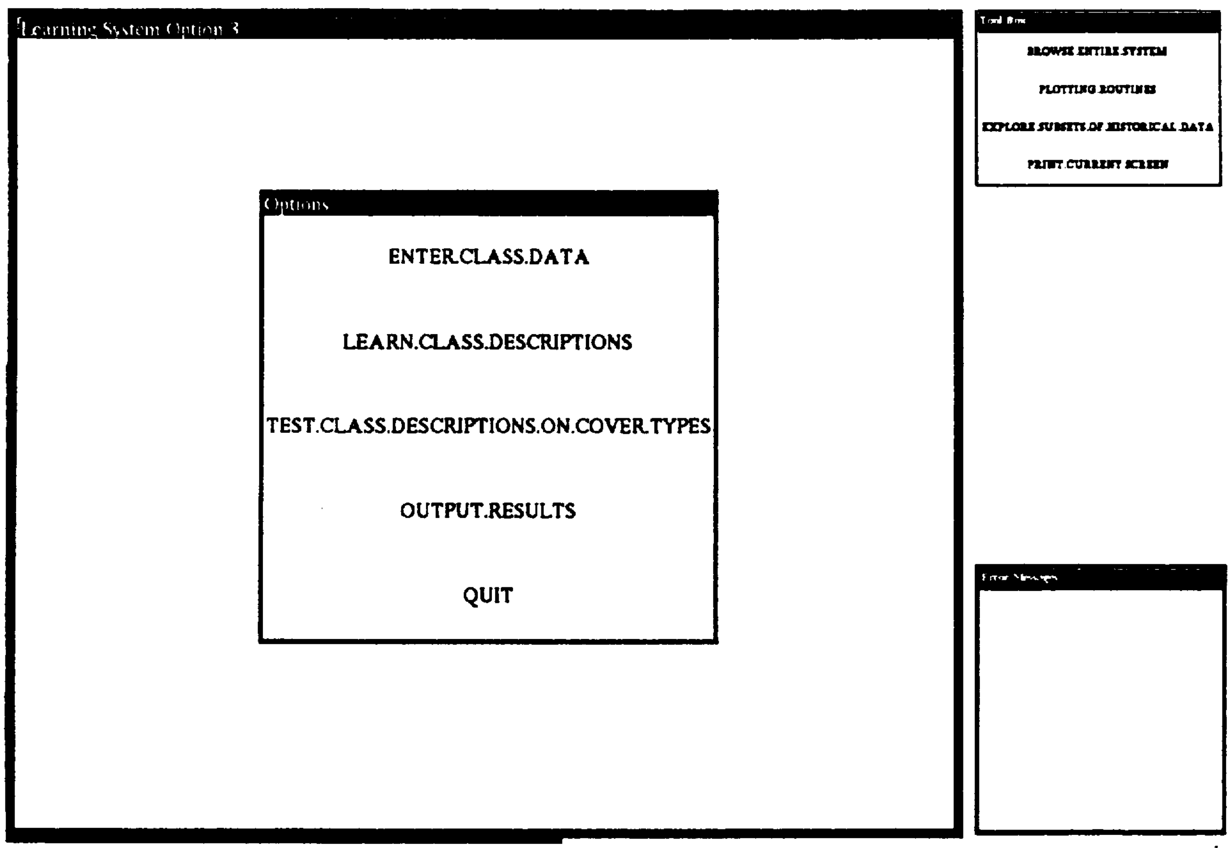


Figure 3-23

The Menu for Option 3

3.3.1 Enter Class Data and Learn Class Descriptions

These options can be selected and run exactly as in option 1. However, option 3 may be selected by the user immediately after running option 1 or option 2 as a means of estimating the accuracy of the results obtained by the previous option. In this case, the user can omit the steps ENTER.CLASS.DATA and LEARN.CLASS.DESCRIPTIONS in option 3, and the learning system will use the class data entered and the class descriptions learned in the previous option.

3.3.2 Test Class Descriptions on Cover Types

When the user selects this option, the Lisp function TEST-CLASS-DESCRIPTIONS-ON-COVER-TYPES is called. This function first makes sure that class data has been entered and class descriptions have been learned. It then processes each training problem in turn. Each cover type in the positive training set is classified using the function CLASSIFY-SAMPLE that was discussed in section 3.2.4. If the best class for the cover type is the class for which the cover type is a member of the positive training set, the cover type is considered to have been correctly classified. This process is repeated for all the cover types in the positive training sets of all the training problems.

The classification accuracy achieved by the learning system is calculated as the number of correctly classified cover types divided by the total number of cover types in the positive training sets of all the training problems. This result is stored in the slot PERFORMANCE.SCORE of the unit TRAINING.DATABASES.

3.3.3 Output Results

As in options 1 and 2, the final step in option 3 is to output the results. The results for option 3 are the same as those of option 1 except for the inclusion of additional data in the Results box. The cover types that were both correctly and incorrectly classified as belonging to the class are listed together with the performance score for all the classes. Figures 3-24, 3-25 and 3-26 show the results obtained when option 3 was applied to the data and classes defined in option 2 and briefly described in Section 3.2.5. The system's classification performance score was 0.8800. These results are discussed in detail in Section 5.2.

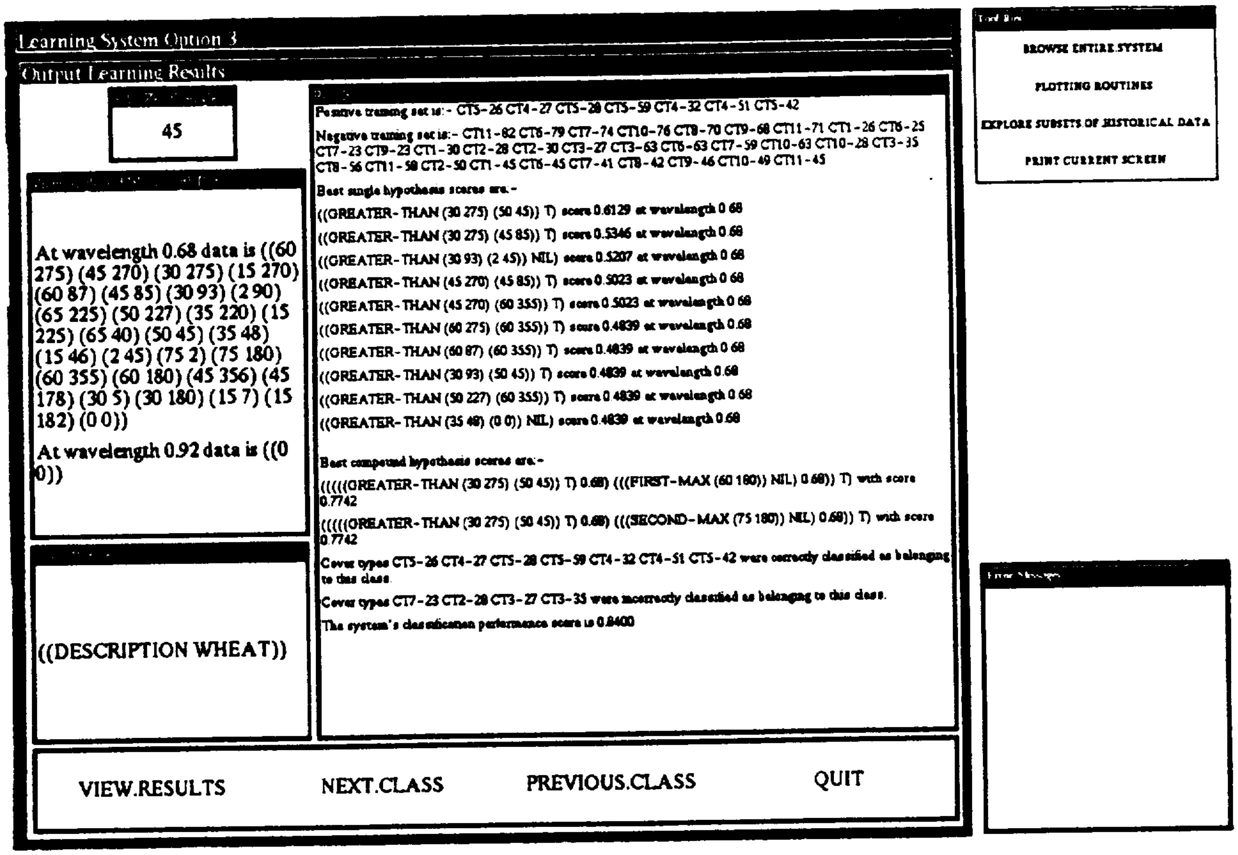


Figure 3-24

Results for SAMPLE3 and the Class (DESCRIPTION WHEAT) in Option 3

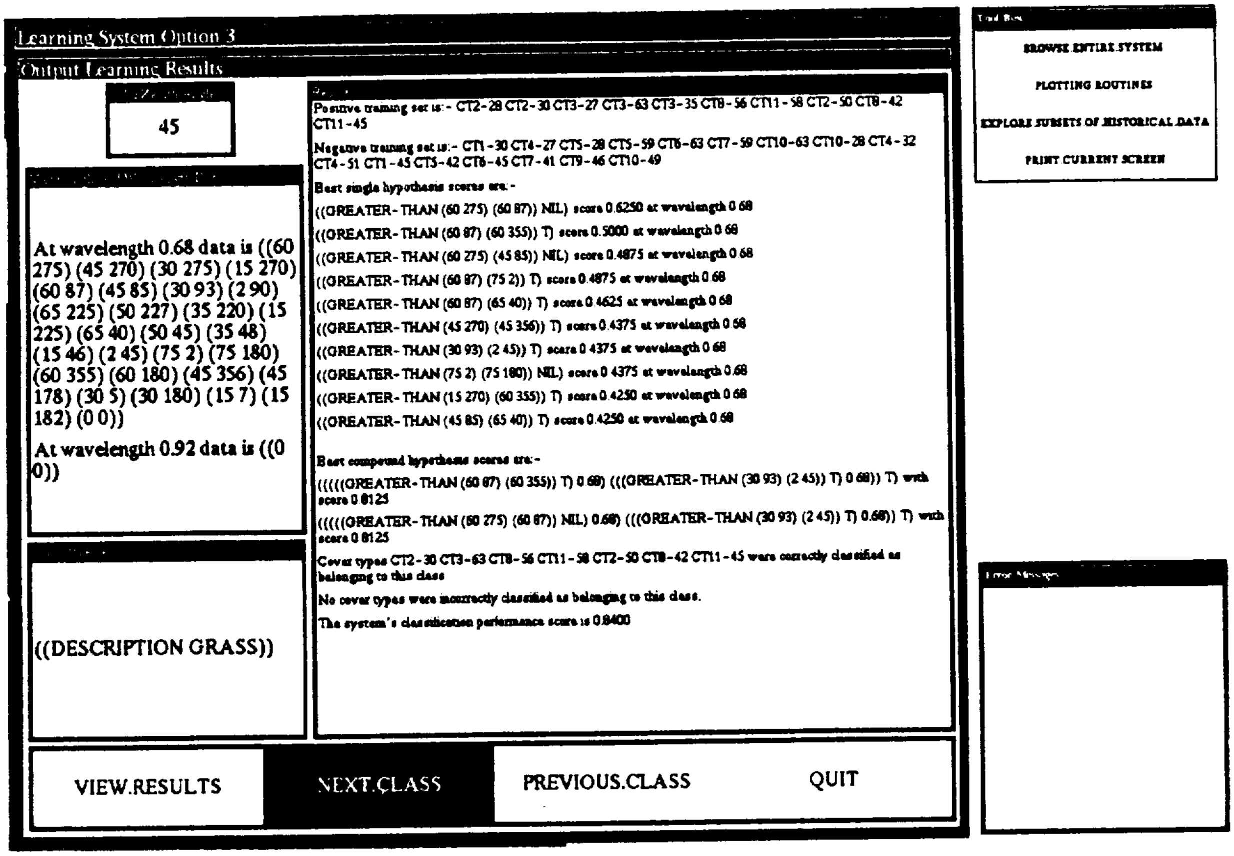


Figure 3-25

Results for SAMPLE3 and the Class (DESCRIPTION GRASS) in Option 3

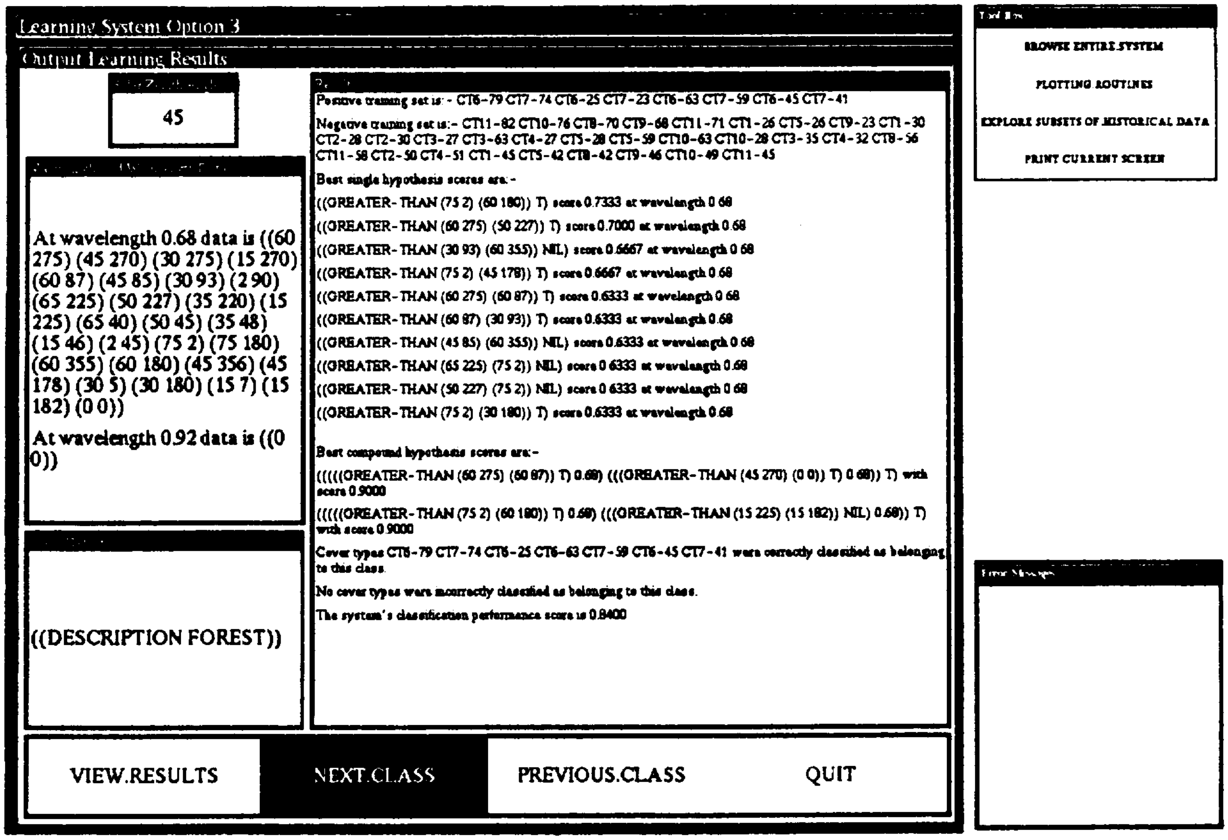


Figure 3-26

Results for SAMPLE3 and the Class (DESCRIPTION FOREST) in Option 3

SECTION 4.0

THE LEARNING SYSTEM IN THE VEG "AUTOMATIC MODE"

The preceding section described the operation of the learning system in the VEG "Research Mode." The learning system can also be run in the VEG "Automatic Mode." Figure 4-1 shows the current version of the "Automatic Mode" top level menu as it first appears on the screen. This screen differs from the screen shown in the reports for Tasks A and B of this contract. In the current version of VEG, the boxes that allow the user to make the selections needed for each subgoal are displayed on the screen only after the subgoal has been selected. If the user selects the subgoal LEARN.CLASS.DESCRIPTIONS, additional boxes are displayed, as shown in Figure 4-2. These boxes allow the user to enter the input and output file names, select the formats of the input and output files and also enter the class descriptions that are to be learned.

The interface to a file of unknown cover type data was described in detail in the report on Task A. This same interface is used by the learning system. When the user enters the name of the input file into the box labelled "Input File Name," the input file interface is opened allowing the user to specify the file format, as described in the previous report.

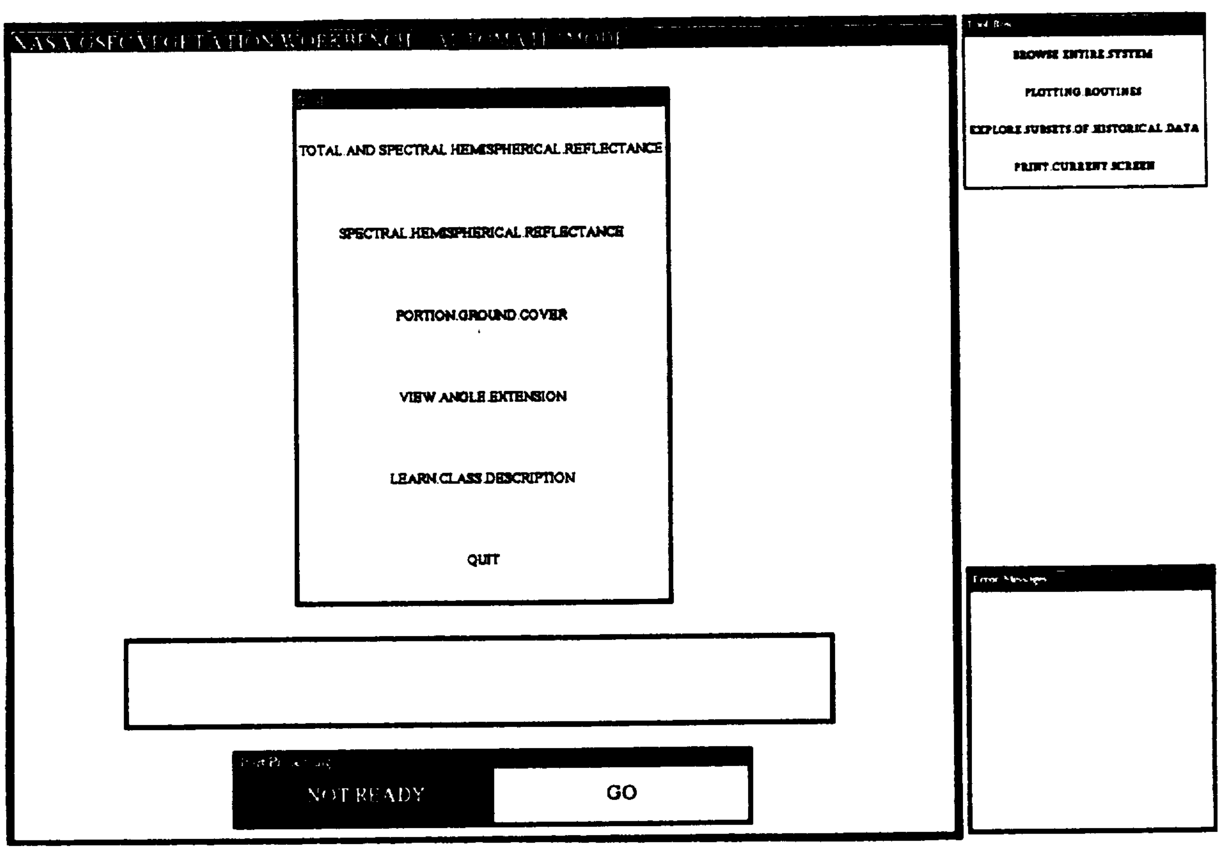


Figure 4-1

The VEG "Automatic Mode" Top Level Menu

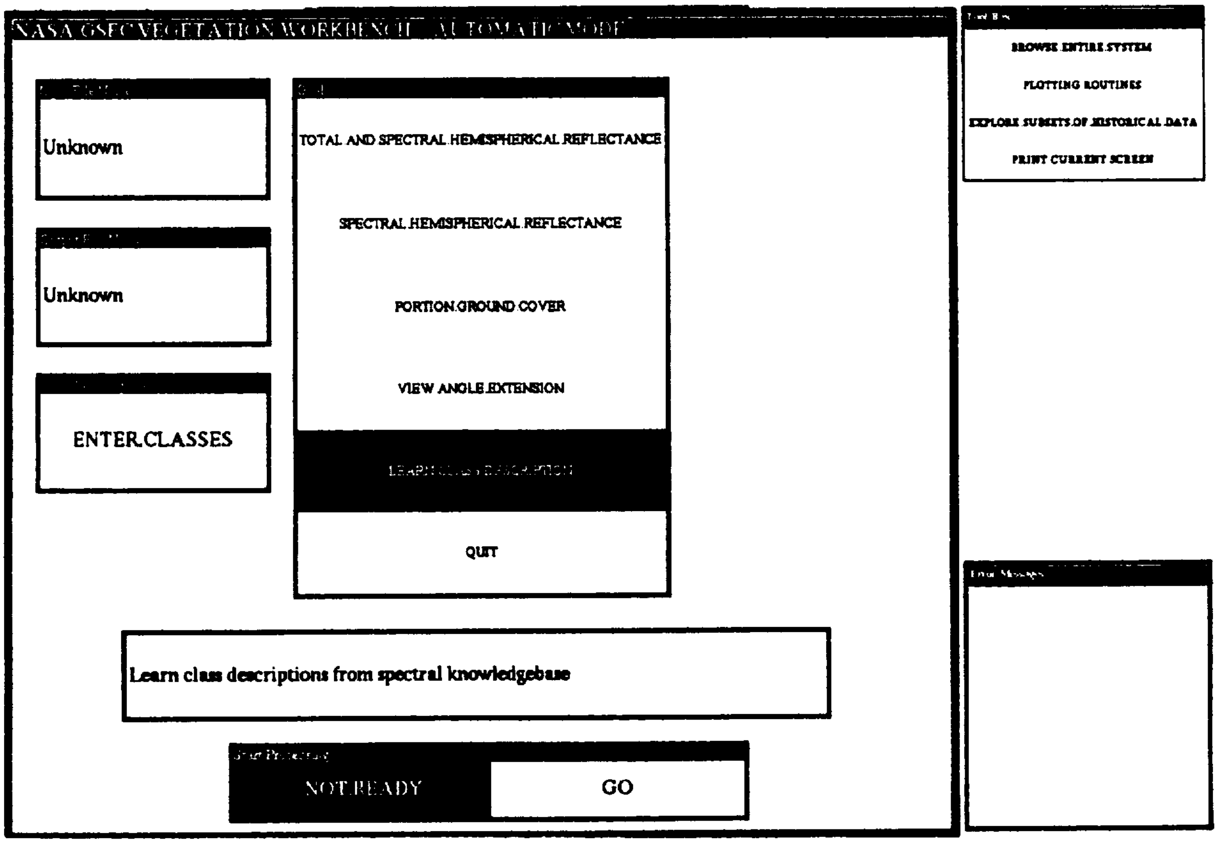


Figure 4-2

Selecting the Learning System from the VEG "Automatic Mode" Top Level Menu

When the user enters the name of the output file, a check is made to determine whether the file already exists. If the file exists, the user is asked whether the file should be overwritten. If the existing file is not to be overwritten, the user is prompted to enter a different file name. The interface shown in Figures 3-15 and 3-16 is then opened, allowing the user to select which data should be written to the output file.

When the user left clicks on the box labelled "Enter Learning Classes," the screen shown in Figure 4-3 is opened. This screen is the same as the screen shown in Figure 3-3 except that the boxes labelled "Additional Wavelength," "Solar Zenith" and "View Angle Data" have been closed. The screen shown in Figure 4-3 enables the user to enter the parameters and ranges of parameter values that define the classes that are to be learned. The selection of parameters and parameter values was described in Section 3.1.1.

Left clicking on "GO" runs the learning system. A check is made to make sure that input and output file names and a class description for at least one class have been entered. If any of these is missing, the user is prompted to enter the missing data.

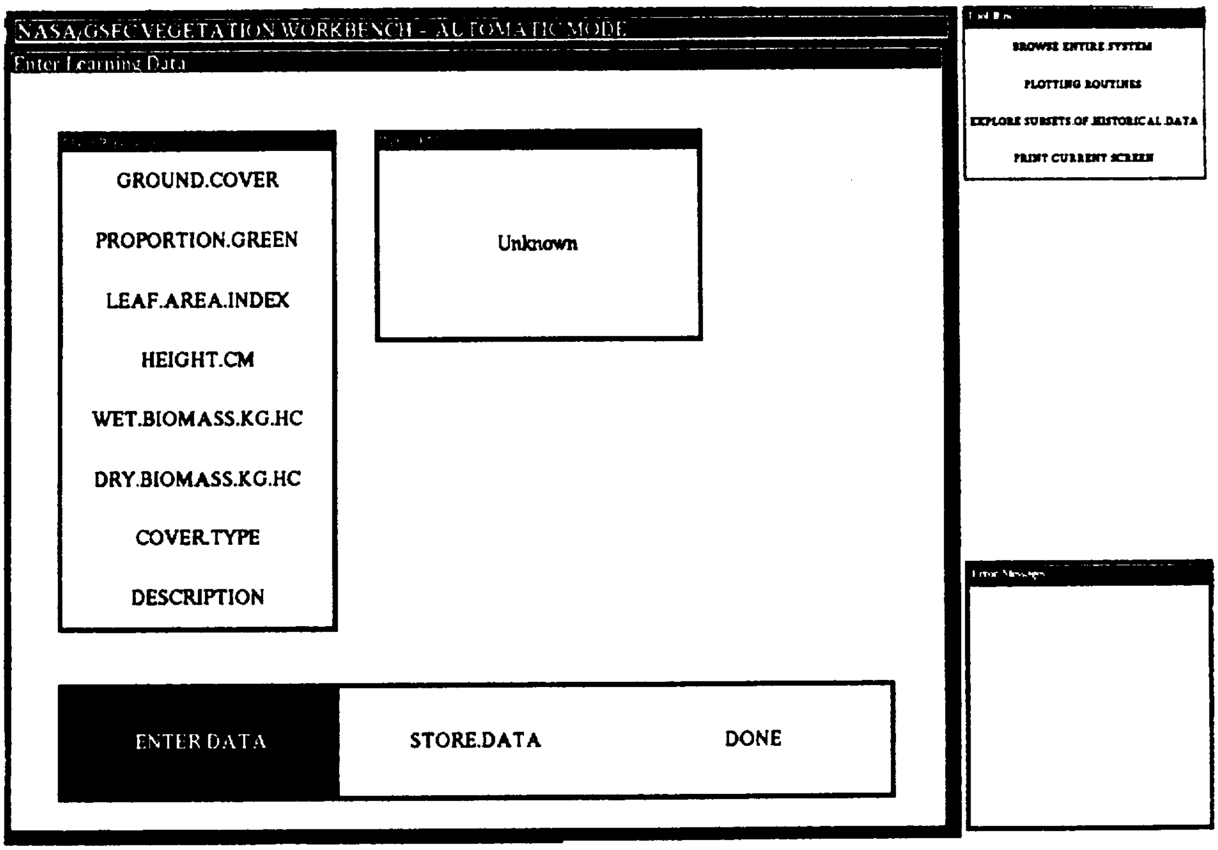


Figure 4-3

Selecting the Class Parameters in the VEG "Automatic Mode"

All the data from the input file is read and stored in units that are subclasses and members of the unit TARGET.DATA. The learning system then processes the data a sample at a time.

The wavelength(s), view angles and solar zenith of the sample under consideration are copied into the appropriate slots of each training problem unit. This data together with the class data entered at the beginning of the run defines one or more training problems that match the current sample. A similar method was applied in Option 2 of the learning system in the VEG "Research Mode." The system then learns the class descriptions of the training problems as was described in Section 3.1.2. The only difference is that the user is not offered the option of writing the intermediate results to a trace file. The sample is then classified according to the learned class descriptions to determine which class it best fits. The cover types in the positive training sets of all the training problems are then classified to measure the classification accuracy of the learning system using the same method as in Option 3 of the learning system in the VEG "Research Mode." The results for the current sample are written to the output file.

Since there may be more than one sample to process, the learning system is re-initialized after each sample is processed (except for the training problem class parameters). This process continues until all samples have been processed.

SECTION 5.0

TESTING AND RESULTS

All the options in the learning system have been tested. The tests included runs in both the VEG "Research Mode" and the "Automatic Mode." In some cases, errors in the coding were found. The code was debugged and the tests repeated until the system performed correctly. The trace files and results files from the test runs are presented in Appendix B.

5.1 TEST 1

This was a simple test in which Option 1 in the "Research Mode" was selected. One class having a ground cover of 0-30% was defined. Four random view angles at wavelength 0.68μm were selected. These were (15 182), (75 90), (0 0) and (35 45). A solar zenith of 45° was chosen. The best single hypothesis scores were:

((GREATER-THAN (15 182) (75 90)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (15 182)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (75 90)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 90) (0 0)) NIL) score 0.7500 at wavelength 0.68
((GREATER-THAN (75 90) (35 45)) NIL) score 0.5833 at wavelength 0.68

Compound hypotheses, each consisting of two separate hypotheses were constructed and tested. However, none produced a discrimination score that was better than the best discrimination score of the separate hypotheses. Thus no further compound hypotheses were investigated and the best compound hypothesis scores were reported as:

(((((FIRST-MAX (75 90)) NIL) 0.68)) T) with score 0.7738
(((((FIRST-MAX (15 182)) T) 0.68)) T) with score 0.7738
(((((GREATER-THAN (15 182) (75 90)) T) 0.68)) T) with score 0.7738

5.2 TEST 2

This test was designed to test Option 2 of the learning system in the VEG "Research Mode." It was briefly described in Sections 3.2.5 and 3.3.3. The sample of cover type data stored in the VEG units SAMPLE3, W5 and W6 was selected as the unknown cover type to be classified. This sample had twenty eight view angles at wavelength 0.68μm stored in W5 and one nadir view angle at wavelength 0.92μm, stored in W6. The reflectance data for SAMPLE3 at wavelength 0.68μm was ((0 0 0.043)(15 182 0.043) (15 7 0.043) (30 180 0.054) (30 5 0.043) (45 178 0.066) (45 356 0.044) (60 180 0.076) (60 355 0.054) (75 180 0.089) (75 2 0.067) (2 45 0.01) (15 46 0.03) (35 48 0.04) (50 45 0.05) (65 40 0.06) (15 225 0.02) (35 220 0.03) (50 227 0.04) (65 225 0.05) (2 90 0.01) (30 93 0.02) (45 85 0.03) (60 87 0.04) (15 270 0.02) (30 275 0.03) (45 270 0.05) (60 275 0.06)). The solar zenith was 45°. Three classes were selected. These were (DESCRIPTION FOREST), (DESCRIPTION GRASS) and (DESCRIPTION WHEAT). The value of the slot SEARCH.DEPTH in the unit LEARNING.METHODS was set to 2. This was to restrict the compound hypotheses investigated to those containing only two separate hypotheses. This test was originally carried out using a search depth of 5. However, after 23 hours of processing, the system was still only considering the hypotheses consisting of 3 separate hypotheses. Each class had approximately 250 separate hypotheses. The run was interrupted because it was concluded that it would not end in a reasonable length of time. The test was repeated with a search depth of 2. This test took 1 hour to complete.

The support scores for the unknown sample in the classes (DESCRIPTION WHEAT), (DESCRIPTION GRASS) and (DESCRIPTION FOREST) were -1.0000, -1.0000 and 0.0000, respectively. The first two of these scores reflected the fact that the unknown sample matched none of the best compound hypotheses for the classes (DESCRIPTION WHEAT) and (DESCRIPTION GRASS). The best compound hypotheses for the class (DESCRIPTION FOREST) were :

(((((GREATER-THAN (60 275)(60 87)) T) 0.68)
(((GREATER-THAN (445 270)(0 0)) T) 0.68)) T) with score 0.9000 (6)

(((((GREATER-THAN (75 2)(60 180)) T) 0.68)
(((GREATER-THAN (15 225)(15 182)) NIL) 0.68)) T) with score 0.9000 (7)

Hypothesis (6) was true for the SAMPLE3 data but hypothesis (7) was not true because the separate hypothesis (((GREATER-THAN (75 2)(60 180)) T) 0.68) was not true. By equation (5), the support score for the class (DESCRIPTION FOREST) was 0.0000. The learning system correctly reported this score. The system correctly reported that the class (DESCRIPTION FOREST) was the best class for the unknown sample. The results of this test are shown in Figures 3-20, 3-21 and 3-22. The trace files and output files for the test are included in Appendix B. This test showed that Option 2 of the learning system in the VEG "Research Mode" was working correctly.

5.3 TEST 3

This test was a continuation of test 2. The tester exited Option 2 and immediately selected Option 3 of the learning system in the VEG "Research Mode." The purpose of this test was to determine the classification accuracy of test 2. The option TEST.CLASS.DESCRIPTIONS.ON. COVER.TYPES was selected from the Option 3 menu. Then the results were output as shown in Figures 3-24, 3-25 and 3-26. Several cover types were correctly classified as belonging to each of the three classes. No cover types were incorrectly classified as belonging to either the class (DESCRIPTION FOREST) or (DESCRIPTION GRASS). However, cover types CT7-23, CT2-28, CT3-27 and CT3-35 were incorrectly classified as belonging to the class (DESCRIPTION FOREST). Cover types CT2-28, CT3-27 and CT3-35 should have been classified as belonging to the class (DESCRIPTION GRASS) and cover type CT7-23 should have been classified as belonging to the class (DESCRIPTION FOREST). The system's performance score of 0.8400 reflected the fact that 21 of the 25 cover types in the positive training sets were correctly classified. This was an acceptable score. This test showed that Option 3 of the learning system in the VEG "Research Mode" was operating correctly.

5.4 TEST 4

The purpose of test 4 was to test the classification accuracy of the learning system. Six separate runs were carried out using test data suggested by the NASA GSFC technical representative. Option 3 of the learning system in the VEG "Research Mode" was selected. The results are summarized in Table 5-1. Listings of the trace files and output files for these runs can be found in Appendix B. In run 6, poorly dispersed view angles were used. As expected, the performance score was lower in run 6 than in the other runs. The performance scores in the various runs were consistent with the expectations of the NASA GSFC technical representative based on his scientific knowledge of the characteristics of the cover types involved in the tests.

Table 5-1

Test 4 Data and Results

Run #	User Defined Classes	View Angles Available	Wavelength (µm)	Solar Zenith Angle	Performance Score
#1	Ground Cover (0-30%) and (31-100%)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	35°	1.00
#2	Ground Cover (0-30%) and (31-100%)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	70°	0.87
#3	Ground Cover (0-30%), (31-60%) and (61-100%)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	35°	0.9524
#4	Plant Height (>10 m) and (<10 m)	HIRIS Example (60 315)(45 315) (30 315)(15 315) (0 0)(15 135)(30 135)	0.91	45°	0.7368
#5	Ground Cover (0-30%) and (31-100%)	Well-Dispersed Data (0 0)(30 45)(60 45) (30 135)(60 135) (30 225)(60 225) (30 315)(60 315)	0.68	40°	0.9565
#6	Ground Cover (0-30%) and (31-100%)	Poorly Dispersed Data (0 0)(10 0)(15 0)	0.68	40°	0.6087

5.5 TEST 5

Test 5 was designed to test the LEARN.CLASS.DESCRIPTIONS part of the learning system in detail and allow the scientist to study the improvement in the discrimination score as increasingly more complex compound hypotheses were constructed and tested. Two learning problems were studied in this test suite.

In run 1 of test 5, Option 2 of the learning system in the VEG "Research Mode" was selected. SAMPLE5 was selected as the unknown sample to be classified. This sample had 16 view angles at wavelength 0.68μm and one view angle at wavelength 0.92μm. The solar zenith was 71°. The classes 0-30%, 31-70% and 71-100% ground cover were defined.

The system took 11 hours to learn the class descriptions for these classes. Studying the file test4-run1-trace, which is included in Appendix B, allows the scientist to study the improvement in the discrimination scores as increasingly more complex compound hypotheses were constructed and tested. This improvement is summarized in Table 5-2. For the class 0-30% ground cover (TRAINING.PROB.276), the best single hypothesis score was 0.8444. This score was increased to 0.9000 when compound hypotheses at level 2 (consisting of 2 separate hypotheses) were considered. Since the improvement between single hypotheses and level 2 hypotheses was less than 10%, no further compound hypotheses were investigated for this training problem. For the class 31-70% ground cover, the best single hypothesis score was 0.4917. This training problem showed the greatest improvement in discrimination score when progressively more complex hypotheses were investigated. The best compound hypothesis discrimination scores for this training problem were 0.8000, 0.9000 and 0.9333 at levels 2, 3 and 4 respectively. Level 5 hypotheses were not investigated since less than a 10% improvement was obtained between the level 3 and level 4 scores. The best single hypothesis discrimination score for the class 71-100% ground cover was 0.8182. This score was increased to 0.9091 when level 2 hypotheses were considered. No improvement was achieved when level 3 hypotheses were investigated for this training problem.

The unknown sample was classified. The classification scores for the classes 0-30%, 31-70% and 71-100% were -0.933, -1.000 and 0.5000 respectively. The sample was classified as belonging to the class 71-100% ground cover because this class had the highest classification score. Option 3 was also selected for this run. The system's classification performance score was calculated as 0.8929.

In run 2 of test 5, Option 2 of the learning system in the VEG "Research Mode" was again selected. SAMPLE3 was selected as the unknown sample to be classified. This sample had 28 view angles at wavelength 0.68μm and one view angle at wavelength 0.92μm. The solar zenith was 45°. The classes 0-30%, 31-70% and 71-100% ground cover were defined. The learning system took 6 hours to learn the class descriptions for these classes. The discrimination scores for the various different levels of compound hypotheses are given in Table 5-3. The best single hypothesis discrimination score for the class 0-30% ground cover was 1. This score represents perfect discrimination so no compound hypotheses were investigated for this class. The best single hypothesis score of 0.5000 for the class 31-70% ground cover was increased to 0.9000 and 0.9667 for the compound hypotheses at levels 2 and 3 respectively. The class 71-100% ground cover had the best discrimination score of 0.9412 at level 1 and 1 at level 2. Run 2 took a shorter time than run 1 of test 5 because less compound hypotheses were investigated. SAMPLE3 had classification scores of -1.0000, 1.0000 and 1.0000 for the classes 0-30%, 31-70% and 71-100% ground cover respectively. The learning system classified the sample as belonging to the class 71-100% ground cover. This class was selected rather than 31-70% ground cover because of the order in which the classes were considered. In fact, the learning system classified the sample as belonging to two different classes in this case. This result has been referred to the NASA GSFC technical representative for interpretation.

Table 5-2

Results of Test 5 Run 1

Class	Training Problem Unit	Best Discrimination Scores				
		Level 1	Level 2	Level 3	Level 4	Level 5
0-30% Ground Cover	TRAINING.PROB.276	0.8444	0.9000	-	-	-
31-70% Ground Cover	TRAINING.PROB.277	0.4917	0.8000	0.9000	0.9333	-
71-100% Ground Cover	TRAINING.PROB.278	0.8182	0.9091	0.9091	-	-

Table 5-3

Results of Test 5 Run 2

Class	Training Problem Unit	Best Discrimination Scores				
		Level 1	Level 2	Level 3	Level 4	Level 5
0-30% Ground Cover	TRAINING.PROB.3	1.0000	-	-	-	-
31-70% Ground Cover	TRAINING.PROB.4	0.5000	0.9000	0.9667	-	-
71-100% Ground Cover	TRAINING.PROB.5	0.9412	1	-	-	-

Test 5 showed that improved discrimination scores can be achieved by investigating progressively more complex compound hypotheses. However, this investigation can be extremely time consuming because of the large number of possibilities involved.

5.6 TEST 6

Test 6 was designed to test the operation of the learning system in the VEG "Automatic Mode." Two learning problems were studied in this test. In each case, a file was constructed to contain the same data as in units of the samples of target data already stored in VEG. The learning problems were solved by the learning system running in the "Automatic Mode," using the data from the specially constructed files. Then the same problems were solved by the learning system running in the "Research Mode," using both options 2 and 3. For each learning problem, the results of the two runs were then compared to make sure the learning system was giving the same results in the "Automatic Mode" as in the "Research Mode." The search depth was limited to 2 in these tests so that the runs would be completed in a reasonable length of time.

In the first learning problem, the unknown sample was SAMPLE7 and the classes were (DESCRIPTION GRASS) and (DESCRIPTION FOREST). In the second problem ,the unknown sample was SAMPLE1 and the classes were 0-30%, 31-70% and 71-100% ground cover. For both learning problems, the learning system returned the same training sets, hypothesis scores, classification scores and performance scores when it was running in the "Automatic Mode" as when it was running in the "Research Mode." The output files for test 6 are included in Appendix B. From the results of test 6, it was concluded that the learning system was operating correctly in the "Automatic Mode."

SECTION 6.0

EXTENDING THE BROWSER TO INCLUDE THE LEARNING SYSTEM

A toolbox is provided in VEG. The toolbox enables the user to browse the system, plot the reflectance data, explore the historical data and print out the screen. The browser enables the user to examine the hierarchy of class, subclass and member units in VEG and display the values of slots.

The learning system was implemented as a separate knowledge base to VEG so it was not accessible to the original version of the browser. The scope of the browser has been extended so that any knowledge base (including the learning system) can be browsed. An additional option "KB" has been added to the browser menu. When the user selects this option, a menu of all the currently loaded knowledge bases is displayed. The user can left click on a knowledge base to select it. The name of the knowledge base is displayed in a box labelled "Current KB" (Figure 6-1). A menu containing all the top level units in the knowledge base, except ActiveImage or ActiveValue units, is then displayed. The user can left click on the name of a unit to select which hierarchy of units to browse. The user can browse up or down the hierarchy of units and display slot values as in the previous version of the browser. When the user attempts to browse "up" from the top level or "down" from the bottom level in a tree of units, a menu containing all the top level units in the knowledge base, except ActiveImage or ActiveValue units, is displayed. The user can select which hierarchy of units to browse next. Figure 6-1 shows a screen dump of VEG with the browser in use.

All the options in the browser were tested using VEG, the learning system and several different system knowledge bases such as KEEINTERFACE. All the tests were successful, showing that the new version of the browser was working correctly. The code for the browser is presented in Appendix C.

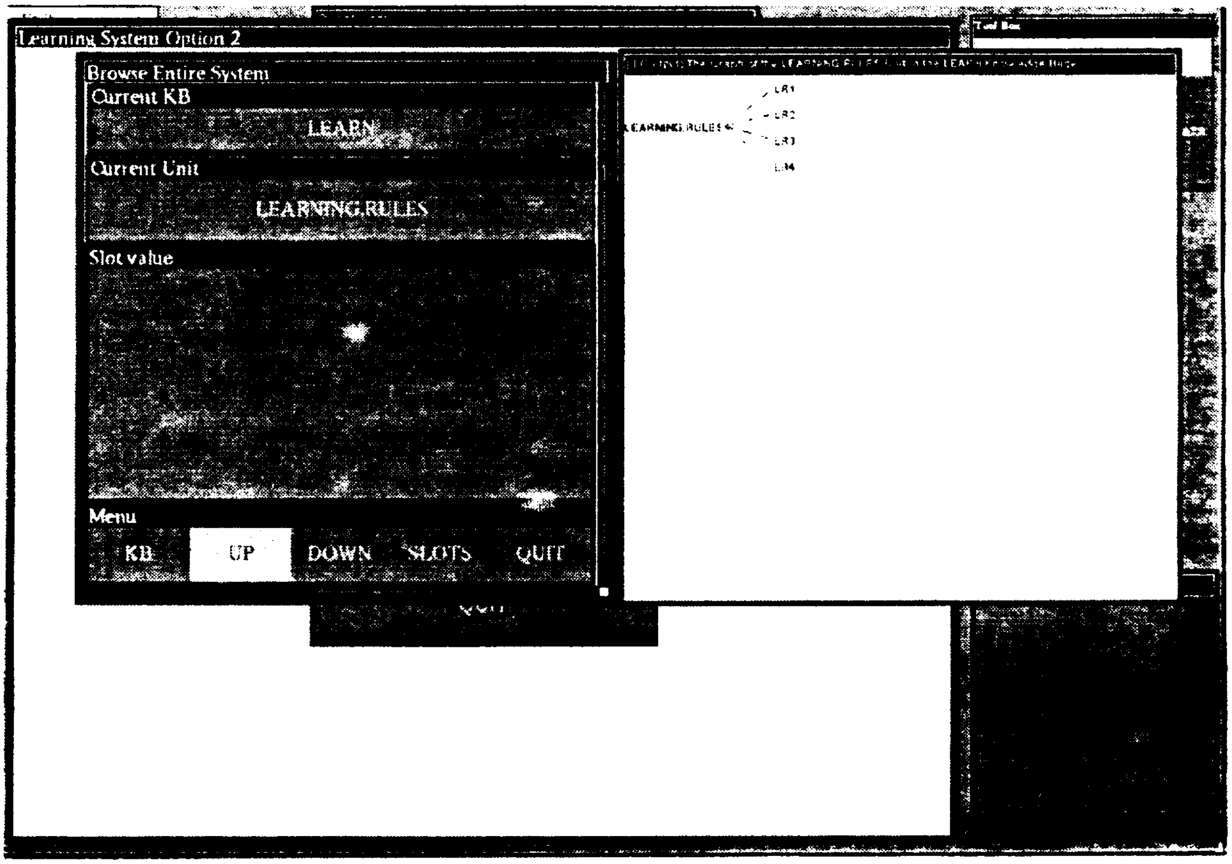


Figure 6-1

Screen Dump of the VEG with the Browser in Use

SECTION 7.0

CONCLUSIONS AND RECOMMENDATIONS

The learning system has been successfully implemented and integrated with the underlying VEG system. It provides additional tools for the scientist to use in exploring classification problems and for archiving results for a variety of research purposes. The browser within VEG has also been expanded to allow the scientist to browse the computational state of the VEG and learning systems.

The combined VEG and learning systems also manage complexity for the scientist, allowing one to focus on important scientific hypotheses rather than on trivial levels of data, computation and step management. It focuses the scientist on the "big" picture rather than burying one in a sea of detail.

The flexibility of the system allows the scientist a platform to conduct any number of explorations of a large body of reflectance data in a very short period of time. What took days in the past can now be accomplished in minutes. This means that the scientist can be much more productive and expansive in his/her thinking than would have been allowable without the time contraction that this system provides.

Finally, the system allows the results of the learning system to be written to a file. This provides a means of creating a ready history that is automatically maintained and is at the service of the scientist.

The current system suggests some interesting challenges for future development. These include:

a) the improved management of computational complexity in forming and testing hypotheses.
b) the development of a fully relational data base interface for the archiving of historical data.
c) additional tools and computational mechanisms to support the exploratory work of the scientist.
d) the generalization of this system to data from other sources such as radar.
e) migration of the system from Lisp to an ANSI standard C/C++ production quality version with explicit memory management rather than implicit memory management via garbage collection.
f) a distributed capability so that large historical data bases could be developed, manages and made available over a network for research work from distributed locations.

The key goal in all of these concepts is to empower the scientist so that he/she can be more productive and creative in his/her work.

REFERENCES

1. Kimes, D. S., Harrison, P. R. and Ratcliffe, P. A. 1991. A Knowledge-Based Expert System for Inferring Vegetation Characteristics. International Journal of Remote Sensing Vol 12, 10, pp. 1987-2020.

2. Kimes, D. S., Harrison, P. A. and Harrison, P. R. 1992. New Developments of a Knowledge Based System (VEG) for Inferring Vegetation Characteristics. International Geoscience and Remote Sensing Symposiom, Houston, Texas, May 1992.

APPENDIX A

LISTING OF METHODS FILES FOR THE LEARNING SYSTEM

```
;;; learn-methods.lisp
;;;
;;; Main methods for the learning system
;;;
;;; Written by Ann and Patrick Harrison
;;; Created 10th July 1992
;;; Last modified 21st September 1992

(in-package 'kee)

;;;-----------------------------------------------------------------------
;;; Top level methods to generate the training sets and scores
;;;-----------------------------------------------------------------------

(defun start-learning-system ()
"Starts the learning system.  Initializes and displays the learning system top
level menu."
  (clear-prompt)
  (remove.all.values 'learning.methods 'option.number)
  (remove.all.values 'learning.system 'learning.options)
  (put.value 'learning.system 'message "")
  (put.value 'learning.system 'learning.menu 'view.options)
  (unitmsg 'viewport-learning.system.1 'open-panel!))

(defun learn-option-1 ()
"Displays the main menu for learning system options 1."
  (put.value 'learning.methods 'option.number 1)
  (remove.all.values 'option.1 'menu)
  (unitmsg 'viewport-option.1.1 'open-panel!))

(defun learn-option-2 ()
"Displays the main menu for learning system options 2."
  (put.value 'learning.methods 'option.number 2)
  (put.value 'estimate.hemispherical.reflectance 'done.enter.data.p nil)
  (remove.all.values 'option.2 'menu)
  (unitmsg 'viewport-option.2.2 'open-panel!))
```

```
(defun match-classes-to-sample()
"Checks to make sure an unknown sample has been selected or entered.  If not
the function displays an error message and returns nil.  Otherwise the function
calls the function set-up-training parameters to set up the necessary class
parameters to match the sample and then initializes and opens the interface for
entering the rest of the training class description(s)."
  (when (null (get.value 'estimate.hemispherical.reflectance
                         'done.enter.data.p))
    (my-documentation-print
     "Enter unknown sample before entering class data")
    (return-from match-classes-to-sample nil))
  (clear-prompt)
  (wipe-out-results 'dummy)
  (initialize-enter-learning-data)
  (set-up-training-parameters)
  (unitmsg 'windowpane-wavelength-of-enter.learning.data.1 'open!)
  (unitmsg 'windowpane-solar.zenith-of-enter.learning.data.2 'open!)
  (unitmsg 'windowpane-view.angle.data.message-of-enter.learning.data.4
           'open!)
  (deactivate-eld-windows)
  (unitmsg 'viewport-enter.learning.data.2 'open-panel!))

(defun set-up-training-parameters ()
"Sets up the directional view angles, solar zenith and wavelength in the
training set to match the unknown sample that is being classified."
  (let ((current-samples (get.values 'estimate.hemispherical.reflectance
                                     'current.sample.wavelengths)))
    (get-view-angle-data current-samples)
    (put.value 'enter.learning.data 'solar.zenith
         (get.value (first current-samples) 'solar.zenith))))

(defun get-view-angle-data (current-samples)
"Stores a list of wavelengths and corresponding view angles in the
view.angle.data slot of enter.learning.data."
  (dolist (sample current-samples)
    (add.value 'enter.learning.data 'view.angle.data
         (list (get.value sample 'wavelength)
             (let ((result ()))
                 (dolist (point (get.value sample 'reflectance.data) result)
                   (push (butlast point) result)))))))

(defun learn-option-3 ()
"Displays the main menu for learning system options 3."
  (put.value 'learning.methods 'option.number 3)
  (remove.all.values 'training.databases 'performance.score)
  (remove.all.values 'option.3 'menu)
  (unitmsg 'viewport-option.3.1 'open-panel!))
```

```
(defun train ()
"If no training classes have been defined the function simply prints an error
message and returns nil.  Otherwise it generates the training sets, runs rules
to generate possible hypotheses and searches for the relationships that best
define the class."
 (let ((current-classes (get.values 'training.databases 'current.classes)))
   (when (null (get.value 'learning.methods 'done.enter.learning.data.p))
     (my-documentation-print
      "Enter classes before learning class descriptions")
     (return-from train nil))
   (my-documentation-print "Learning class descriptions .......... ")
   (initialize-training-sets `dummy)
   (catch 'no-training-sets
     (dolist (problem current-classes)
          (generate-training-sets problem)
          (interp-extrap-training-data problem))
     (forward.chain 'learning.rules)
     (put.value 'learning.methods 'done.learn.class.descriptions.p t)
     (ask-about-trace-file))))

;;;-----------------------------------------------------------------------
;;; Methods to enter data to define the classes to be classified
;;;-----------------------------------------------------------------------

(defun open-enter-learning-data-interface ()
"Method to open the interface for entering data to define a class to be
characterized."
  (clear-prompt)
  (wipe-out-results 'dummy)
  (initialize-enter-learning-data)
  (unitmsg 'windowpane-wavelength-of-enter.learning.data.1 'open!)
  (unitmsg 'windowpane-solar.zenith-of-enter.learning.data.2 'open!)
  (unitmsg 'windowpane-view.angle.data.message-of-enter.learning.data.4
           'open!)
  (reactivate-eld-windows)
  (unitmsg 'viewport-enter.learning.data.2 'open-panel!))

(defun not-in-use (&rest lis)
"Function installed in the mouseleftfn! slot of the wavelength image in the
enter.learning.data interface.  Displays a message and prevents further input
of additional wavelengths when option 2 is in use."
  (declare (ignore lis))
  (my-documentation-print
"In option 2 use only the wavelengths, solar zenith and directional view angles
that match the unknown sample"))

(defun deactivate-eld-windows ()
"Puts the function not-in-use in the mouseleftfn! slot of the appropriate
images in the enter.learning.data interface."
  (put.value 'windowpane-wavelength-of-enter.learning.data.1
     'mouseleftfn! #'not-in-use)
  (put.value 'windowpane-solar.zenith-of-enter.learning.data.2
     'mouseleftfn! #'not-in-use))
```

```
(defun reactivate-eld-windows ()
"Puts the function simple-value-display-actuator back into the mouseleftfn!
 slot of the appropriate images in the enter.learning.data interface."
  (put.value 'windowpane-wavelength-of-enter.learning.data.1
     'mouseleftfn! #'simple-value-display-actuator)
  (put.value 'windowpane-solar.zenith-of-enter.learning.data.2
     'mouseleftfn! #'simple-value-display-actuator))

(defun initialize-enter-learning-data ()
"Initializes the slots in the unit enter.learning.data ready for the entry of
new class(es) to be learned."
   (remove.all.values 'training.databases 'performance.score)
   (remove.all.values 'enter.learning.data 'class.parameter)
   (remove.all.values 'enter.learning.data 'solar.zenith)
   (remove.all.values 'enter.learning.data 'value)
   (remove.all.values 'enter.learning.data 'wavelength)
   (remove.all.values 'enter.learning.data 'class)
   (remove.all.values 'enter.learning.data 'view.angle.data)
   (remove.all.values 'training.databases 'current.classes)
   (put.value 'enter.learning.data 'message "")
   (remove.all.values 'enter.learning.data 'view.angle.data.message)
   (put.value 'enter.learning.data 'menu 'enter.data))

(defun valid-directional-view-angles (data)
"Returns t if the directional view angles are valid and nil otherwise."
   (and (listp data)
        (dolist (point data t)
          (unless (and (listp point)
                       (= (length point) 2))
            (return-from valid-directional-view-angles nil))
          (let ((z (zenith point))
                (a (azimuth-360 point)))
            (unless (and (numberp z) (>= z 0) (<= z 90)
                         (numberp a) (>= a 0) (<= a 360))
              (return-from valid-directional-view-angles nil))))))

(defun store-directional-view-angles (dva)
"Stores the new wavelength and corresponding view angles in the view angle data
slot of the unit enter.learning.data."
   (add.value 'enter.learning.data 'view.angle.data
      (list (get.value 'enter.learning.data 'wavelength)
            dva))
   (remove.all.values 'enter.learning.data 'wavelength))
```

```
(defun store-learning-data()
"Creates the required unit and stores the data defining a class to be learned
in the knowledgebase."
  (let ((class (get.values 'enter.learning.data 'class))
        (view-angle-data
         (get.values 'enter.learning.data 'view.angle.data))
        (solar-zenith (get.value 'enter.learning.data 'solar.zenith)))
    (if (and class view-angle-data solar-zenith)
        (let ((new-unit (create.unit    (gentemp "TRAINING.PROB.")
                                'learn 'training.databases)))
          (put.values new-unit 'class class)
          (put.values new-unit 'view.angle.data
             view-angle-data)
          (put.values new-unit 'view.angle.data.message
             (get.values 'enter.learning.data 'view.angle.data.message))
          (put.value new-unit 'solar.zenith solar-zenith)
          (dolist (va view-angle-data)
           (add.value new-unit 'wavelength (first va)))
          (add.value 'training.databases 'current.classes new-unit)
          (put.value 'learning.methods 'done.enter.learning.data.p t)
          (my-documentation-print "Data stored"))
        (my-documentation-print "Data incomplete  - not stored"))))

(defun valid-parameter (value min max)
"Returns t if the value is a list of numbers, each is within range and the
first number is less than the second.  Otherwise returns nil."
  (and (listp value)
       (= (length value) 2)
       (let ((f (first value))
             (s (second value)))
         (and (numberp f)(numberp s)
              (< f s)
              (>= f min)(<= f max)
              (>= s min)(<= s max)))))

(defun valid-parameter-values(new-val new-parameter)
"Returns t if the parameter value(s) entered are valid for a possible
training class in the learning system and nil otherwise."
  (case new-parameter
    (cover.type (member new-val (get.values 'enter.learning.data
                                    'possible.cover.types)))
    (description (search (string new-val )
                         (get.value 'enter.learning.data
                            'possible.descriptions)))
    (ground.cover (valid-parameter new-val 0 1))
    (proportion.green (valid-parameter new-val 0 1))
    (leaf.area.index (valid-parameter new-val 0 1))
    (height.cm (valid-parameter new-val 0 3000))
    (wet.biomass.kg.hc (valid-parameter new-val 0 1))
    (dry.biomass.kg.hc (valid-parameter new-val 0 1))))
```

```
(defun store-parameter-values (new-val new-parameter)
"Stores the new parameter and value in the class slot of the unit
enter.learning.data."
  (add.value 'enter.learning.data 'class (list new-parameter new-val)))

;;;------------------------------------------------------------------------
;;; Methods to generate the training sets
;;;------------------------------------------------------------------------

(defun data-available (object class)
"Returns t if the object has data for each parameter included in the class
definition."
  (dolist (par class t)
    (unless (get.value object (first par))
      (return-from data-available nil))))

(defun wavelengths-available (covertype class-wavelengths)
"Returns t if covertype data is available for each of the class wavelengths and
 nil otherwise."
  (let ((cover-wavelengths (get.values covertype 'wavelengths)))
    (dolist (wave class-wavelengths t)
      (dolist (band cover-wavelengths
                (return-from wavelengths-available nil))
          (when (and (>= wave (first band))
                     (<= wave (second band)))
            (return t))))))
```

```
(defun generate-training-sets (problem)
"Method to generate the training sets for a class definition. This function
extracts the parameters to be matched and then performs a search through the
historical data to find the cover types that best match the required
parameters.  The search ends when sufficient cover types have been found or
it has been established that insufficient cover types are available."
  (let* ((class (get.values problem 'class))
         (class-solar-zenith (get.value problem 'solar.zenith))
       (solar-zenith-interval (* 0.10 class-solar-zenith))
         (class-wavelengths (get.values problem 'wavelength))
         (min-set-size (get.value 'training.databases 'minimum.set.size))
         (max-set-size (get.value 'training.databases 'maximum.set.size)))
     (do* ((n 1 (1+ n))
           (solar-zenith-int solar-zenith-interval
                         (* n solar-zenith-interval))
           (num-pos-sets 0 (length (get.values problem 'pos.training.set)))
           (num-neg-sets 0 (length (get.values problem 'neg.training.set))))
          ((or (> num-pos-sets max-set-size)
               (> num-neg-sets max-set-size)
               (and (> num-pos-sets min-set-size)
                  (> num-neg-sets min-set-size))
               (= n 10)) ;Sets full or insufficient data available
           (cond ((zerop num-pos-sets)
                  (my-documentation-print
            "Discrimination not possible - no positive training sets")
                  (throw 'no-training-sets))
                 ((zerop num-neg-sets)
                  (my-documentation-print
                  "Discrimination not possible - no negative training sets")
                  (throw 'no-training-sets))))
  ;Consider every possible wavelength and sun angle for every cover-type
     (dolist (covertype (unit.children 'historical.cover.types 'subclass))
        (when (and (data-available covertype class)
                       ;Data available for classification parameter
                  (wavelengths-available covertype class-wavelengths))
          (dolist (sun-angle (unit.children covertype 'subclass))
            (when
                (solar-zenith-p sun-angle class-solar-zenith ;Sun angle matches
                                solar-zenith-int)
             (add-to-training-set problem class sun-angle))))))))
```

```
(defun pos-training-example (object class)
"Return t if the example is in the positive training class and nil otherwise."
 (dolist (par class t)
  (let* ((parameter (first par))
         (class-value (get.value object parameter))
         (val (second par)))
    (cond ((listp val)                    ;Continuous parameter
           (unless (and (<= class-value (second val))
                        (>= class-value (first val)))
             (return-from pos-training-example nil)))
          ((eq parameter 'cover.type)
           (unless (eq class-value val)
             (return-from pos-training-example nil)))
          (t                              ;Parameter is description
           (unless (search (string val) class-value)
             (return-from pos-training-example nil)))))))

(defun add-to-training-set (problem class cover-type)
"Adds the cover type name to the positive.training.set or negative.training.set
 slot of the training problem unit, as approproate."
  (if (pos-training-example cover-type class)
     (add.value problem 'pos.training.set cover-type)
     (add.value problem 'neg.training.set cover-type)))

(defun delete-sub-units (parent)
"If parent is a training problem unit, this function deletes all the subunits
of the training set unit."
  (when (member parent (unit.children 'training.databases 'subclass))
    (dolist (ts (unit.children parent 'subclass))
      (dolist (ts1 (unit.children ts 'subclass))
          (dolist (un (unit.children ts1 'member))
            (delete.unit un))
          (delete.unit ts1))
      (delete.unit ts))))

(defun initialize-training-set(parent)
"Initializes a training set by removing all the units in the positive and
negative training sets and by removing the values in the hypotheses, scores and
training set slots of the training problem unit."
  (delete-sub-units parent)
  (remove.all.values parent 'hypotheses)
  (remove.all.values parent 'scores)
  (remove.all.values parent 'pos.training.set)
  (remove.all.values parent 'neg.training.set)
  (remove.all.values parent 'best.score)
  (remove.all.values parent 'best.class)
  (remove.all.values parent 'sample.score))

(defun initialize-training-sets (self)
"Initializes all the training sets."
  (declare (ignore self))
  (dolist (prob (unit.children 'training.databases 'subclass))
    (initialize-training-set prob)))
```

```
(defun wipe-out-results (self)
"Removes all the current training problems."
  (declare (ignore self))
  (put.value 'learning.methods 'done.enter.learning.data.p nil)
  (put.value 'learning.methods 'done.classify.sample.p nil)
  (put.value 'learning.methods 'done.classify.cover.types.p nil)
  (put.value 'learning.methods 'done.learn.class.descriptions.p nil)
  (remove.all.values 'training.databases 'current.classes)
  (remove.all.values 'training.databases 'performance.score)
  (dolist (prob (unit.children 'training.databases 'subclass))
    (delete-sub-units prob)
    (delete.unit prob)))

;;;------------------------------------------------------------------------
;;; Functions used to extrapolate the cover type data to set up the training
;;; set unit
;;;------------------------------------------------------------------------

(defun create-training-set-parent-units(problem)
"Creates the positive and negative training set parent units as subclasses of
the training problem unit."
  (let ((unit-name (string (unit.name problem))))
        (values (create.unit
                    (intern (string-append "POS." unit-name))
                    'learn problem)
                (create.unit
                    (intern (string-append "NEG." unit-name))
                    'learn problem))))

(defun interp-extrap-training-data(problem)
"Sets up the units be the parents of the positive and negative training data
sets and calls the function to create the child units to hold the training
data."
  (let ((view-angle-data (get.values problem 'view.angle.data)))
    (multiple-value-bind (pos-parent neg-parent)
        (create-training-set-parent-units problem)
      (interp-extrap-training-data-aux view-angle-data pos-parent
                                        (get.values problem 'pos.training.set)
                                        "POS")
      (interp-extrap-training-data-aux view-angle-data neg-parent
                                        (get.values problem 'neg.training.set)
                                        "NEG"))))
```

```
(defun interp-extrap-training-data-aux (view-angle-data parent
                                        training-set str)
"Interpolates and extrapolates the covertype data and stores it in the
appropriate units in the positive and negative training data sets."
  (dolist (sun-angle training-set)
    (let ((new-unit (create.unit (gentemp str) 'learn parent)))
      (put.value new-unit 'cover.type sun-angle)
      (dolist (vad view-angle-data)
          (let ((wave-unit (create.unit (gentemp str) 'learn nil new-unit))
                (wave (first vad)))
            (put.value wave-unit 'wavelength wave)
            (put.value wave-unit 'reflectance.data
              (match-unaltered-target-data (second vad)
                                           (get-wave-unit wave sun-angle))))))))

(defun get-wave-unit (wave sun-angle)
"Finds the appropriate member unit of the unit sun-angle that has the
required wavelength to match the training class."
  (dolist (wav (unit.children sun-angle 'member))
    (when (and (>= wave (get.value wav 'wavelength.min))
               (<= wave (get.value wav 'wavelength.max)))
      (return-from get-wave-unit wav))))

;;;------------------------------------------------------------------------
;;;Functions used to test possible discriminating relationships
;;;------------------------------------------------------------------------

(defun first-max (data flag direction &rest lis)
"Tests whether the maximum of all the reflectance values in the data is at
direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
    (if (eq (= (third point)
               (apply #'max (mapcar #'third data)))
            flag)
        1
        0)))

(defun first-min (data flag direction &rest lis)
"Tests whether the minimum of all the reflectance values in the data is at
direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
    (if (eq (= (third point)
               (apply #'min (mapcar #'third data)))
            flag)
        1
        0)))
```

```
(defun second-max (data flag direction &rest lis)
"Tests whether the second maximum of all the reflectance values in the data is
at direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
    (if (eq (= (third point)
               (third (second (sort data #'> :key #'third))))
            flag)
        1
        0)))

(defun second-min (data flag direction &rest lis)
"Tests whether the second minimum of all the reflectance values in the data is
at direction 1. If this test gives a result equal to the flag, the function
returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((point (find direction data :test #'zeniths-and-azimuths-match)))
    (if (eq (= (third point)
               (third (second (sort data #'< :key #'third))))
            flag)
        1
        0)))

(defun greater-than (data flag direction1 direction2)
"Tests whether the reflectance value at direction 1 is greater than the
reflectance value at direction 2.  If this test gives a result equal to the
flag, the function returns 1.  Otherwise it returns 0."
  (let ((point1 (find direction1 data :test #'zeniths-and-azimuths-match))
        (point2 (find direction2 data :test #'zeniths-and-azimuths-match)))
  (if (eq (> (third point1)(third point2))
          flag)
    1
    0)))
```

```
(defun backscatter>forwardscatter (data flag &rest lis)
"Tests whether the average backscatter reflectance value in the data is greater
than the average forwardscatter reflectance value.  If this test gives a
result equal to the flag, the function returns 1.  Otherwise it returns 0."
  (declare (ignore lis))
  (let ((sum-back 0)
        (num-back 0)
        (sum-forward 0)
        (num-forward 0))
    (dolist (ang data)
     (let ((az (azimuth-360 ang)))
        (cond ((>= az 180)              ;backscatter angle
               (incf num-back)
               (incf sum-back (third ang)))
              ((< az 180)               ;forwardscatter angle
               (incf num-forward)
               (incf sum-forward (third ang)))
              (t nil))))                ;nadir - ignore
    (if (eq (> (/ sum-back num-back)          ;average back-scatter
               (/ sum-forward num-forward))   ;average back-scatter
            flag)
        1
        0)))

(defun score (problem func wave &optional arg1 arg2)
"Calculates the best discrimination score for a relationship.  For example it
might test ((GREATER-THAN (60 180)(20 0)) T) and ((GREATER-THAN (60 180)(20 0))
 NIL).  The function returns the set of data containing the higher of these
scores (eg (((GREATER-THAN (60 180)(20 0)) T) 0.6543)) or
(((GREATER-THAN (60 180)(20 0)) ?) 0.0001) if both tests gave a discrimination
score of 0."
   (let ((training-data-sets (unit.children problem 'subclass)))
    (multiple-value-bind (num-pos pos-total)
        (score-aux (second training-data-sets) func wave arg1 arg2)
      (multiple-value-bind (num-neg neg-total)
          (score-aux (first training-data-sets) func wave arg1 arg2)
        (let ((pos-score (/ pos-total num-pos))
              (neg-score (/ neg-total num-neg)))
          (values (make-result func arg1 arg2 t
                               (- pos-score neg-score)
                       pos-score
                       neg-score
                       wave)
                  (make-result func arg1 arg2 nil
                               (- neg-score pos-score)
                       (- 1 pos-score)
                       (- 1 neg-score)
                       wave)))))))
```

```
(defun score-aux (training-set func wave arg1 arg2)
"Returns the number of elements and the total score for a training set."
 (let ((total 0)
       (num 0))
  (dolist (parent (unit.children training-set `subclass))
   (dolist (unit (unit.children parent 'member))
      (when (= wave (get.value unit 'wavelength))
       (incf num)
       (incf total
            (funcall func (get.value unit 'reflectance.data)
                   t arg1 arg2)))))
  (values num total)))

(defun make-result (func arg1 arg2 flag score pos-score neg-score wave)
"Sets up the format of a set of result data to allow for different numbers of
arguments without including unnecessary nils."
 (cond ((null arg1)
       `(((((,func)            ;No arguments
           ,flag)
           ,wave))
           ,score ,pos-score ,neg-score))
      ((null arg2)
       `(((((,func ,arg1)        ;1 argument
           ,flag)
           ,wave))
           ,score ,pos-score ,neg-score))
      (t
       `(((((,func ,arg1 ,arg2)
           ,flag)
           ,wave))
           ,score ,pos-score ,neg-score))))

;;;-----------------------------------------------------------------------
;;;Functions required by learning system rules
;;;-----------------------------------------------------------------------

(defun try-direction-relationships (unit view-angle-data)
"Sets up all possible distinct greater-than relationships for a set of data
and adds them to the hypotheses slot of the problem unit."
 (let ((hypotheses nil)
       (wave (first view-angle-data))
       (view-angles (second view-angle-data)))
  (do ((first-ang (first view-angles)(first remaining-angles))
       (remaining-angles (rest view-angles)(rest remaining-angles)))
      ((null remaining-angles))
   (dolist (second-ang remaining-angles)
      (push (list 'greater-than wave first-ang second-ang) hypotheses)))
  (add.values unit 'hypotheses hypotheses)))
```

```
(defun try-max-min-relationships (unit view-angle-data)
"Sets up all possible distinct first-max and first-min relationships for a set
of data and adds them to the hypotheses slot of the problem unit."
 (let ((hypotheses nil)
        (wave (first view-angle-data))
        (view-angles (second view-angle-data)))
   (dolist (ang view-angles)
    (push (list 'first-max wave ang) hypotheses)
    (push (list 'first-min wave ang) hypotheses))
   (add.values unit 'hypotheses hypotheses)))

(defun try-second-max-min-relationships (unit view-angle-data)
"Sets up all possible distinct second-max and second-min relationships for a
set of data and adds them to the hypotheses slot of the problem unit."
 (let ((hypotheses nil)
        (wave (first view-angle-data))
        (view-angles (second view-angle-data)))
   (dolist (ang view-angles)
    (push (list 'second-max wave ang) hypotheses)
    (push (list 'second-min wave ang) hypotheses))
   (add.values unit 'hypotheses hypotheses)))

(defun full-string-data-in-plane(view-angles)
"Returns t if the string data is a full string in a plane and nil otherwise."
  (let ((copy-view-angles (copy-list view-angles)))
  (multiple-value-bind (backscatter-az forwardscatter-az)
    (get-azimuth-angles (azimuth-360 (first copy-view-angles)))
   (let ((nadir (find-if #'(lambda (x) (zerop (zenith x))) copy-view-angles))
          (backscatter (sort (find-half-string backscatter-az copy-view-angles)
                             #'< :key #'zenith))
          (forwardscatter (sort (find-half-string forwardscatter-az
                                                   copy-view-angles)
                                 #'< :key #'zenith)))
    (and (= (length copy-view-angles)    ;All data makes up only 1 string
           (+ (if (null nadir) 0 1)
                (length backscatter)
                (length forwardscatter)))
          (half-string backscatter)
          (half-string forwardscatter))))))

(defun try-backscatter>forwardscatter-relationship (unit view-angle-data)
"Adds backscatter>forwardscatter to the hypotheses slot of the problem unit."
    (add.value unit 'hypotheses
          (list 'backscatter>forwardscatter (first view-angle-data))))
```

```
;;;--------------------------------------------------------------------------
;;; Functions to Output Results of Learning System
;;;--------------------------------------------------------------------------

(defun open-output-learning-results-interface ()
"Test to make sure the pre-requisite steps have been carried out - if not
prints error message and exits.  Opens the interface for displaying the results
of the learning system."
  (let ((option-number (get.value 'learning.methods 'option.number)))
    (case option-number
        (1 (when (not (get.value 'learning.methods
                        'done.learn.class.descriptions.p))
            (my-documentation-print
             "Learn class descriptions before outputing the results")
            (return-from open-output-learning-results-interface nil)))
        (2 (when (not (get.value 'learning.methods 'done.classify.sample.p))
            (my-documentation-print
             "Classify the unknown sample before outputting the results")
            (return-from open-output-learning-results-interface nil)))
        (3 (when (not
                  (get.value 'learning.methods 'done.classify.cover.types.p))
            (my-documentation-print
"Test class descriptions on cover types before outputting the results")
            (return-from open-output-learning-results-interface nil)))))
  (clear-prompt)
  (put.value 'output.learning.results 'message "")
  (put.value 'output.learning.results 'current.class.number 0)
  (remove.all.values 'output.learning.results 'menu)
  (unitmsg 'viewport-output.learning.results.1 'open-panel!)
  (display-learning-results 0))

(defun display-learning-results (num)
"Displays the results for one class in the learning system."
  (let* ((problem (nth num (get.values 'training.databases 'current.classes)))
         (class (get.values problem 'class)))
    (put.value 'output.learning.results 'class class)
    (put.values 'output.learning.results 'view.angle.data.message
         (get.values problem 'view.angle.data.message))
    (put.value 'output.learning.results 'solar.zenith
         (get.value problem 'solar.zenith))
    (put.value 'output.learning.results 'message
         (get-scores (get.values problem 'scores) class problem))))
```

```
(defun get-scores (scores class problem)
"Displays the classification scores for a class. If option 2 has been
selected it also displays the class score for the unknown sample and
indicates if this class is the best class for the sample. If option 3 has
been selected it also displays the classification performance score for
the class operating on the data base of cover type data."
  (let ((result "")
        (option-number (get.value 'learning.methods 'option.number)))
    (setf result (format () "Positive training set is:-~{ ~S~}  Negative training set is:-~{ ~S~}  Best
single hypothesis scores are:-  "
                         (get-unit-names
                          (get.values problem 'pos.training.set))
                         (get-unit-names
                          (get.values problem 'neg.training.set))))
    (dolist (score scores result)
     (let ((sc (first (first score))))
         (setf result (string-append result (format ()
            "~S score ~,4F at wavelength ~S  "
            (first sc)
            (second score)
            (second sc))))))
    (setf result (string-append result (format ()
         "  Best compound hypothesis scores are:-  ")))
    (dolist (score (get.values problem 'best.score))
     (setf result (string-append result (format ()
       "~S with score ~,4F  " (first score)(second score)))))
    (case option-number
     (2
       (setf result (string-append result (option-2-result problem class))))
     (3
       (setf result (string-append result (option-3-result problem)))))
    result))

(defun option-2-result (problem class)
  (let ((result (format ()
                        "  Score for unknown sample for class ~S is ~,4F"
                        class
                        (get.value problem 'sample.score))))
    (when (get.value problem 'best.class)
     (setf result (string-append result (format ()
          "  The class ~S is the best class for this unknown sample" class))))
    result))
```

```
(defun option-3-result (problem)
 (let ((result "")
        (correct-matching-c-t
         (get-unit-names
          (get.values problem 'correct.matching.cover.types)))
        (incorrect-matching-c-t
         (get-unit-names
          (get.values problem 'incorrect.matching.cover.types))))
   (if correct-matching-c-t
        (setf result (format ()
"Cover types~{ ~S~} were correctly classified as belonging to this class.  "
                  correct-matching-c-t))
        (setf result (format ()
"No cover types were correctly classified as belonging to this class.  ")))
   (if incorrect-matching-c-t
        (setf result (string-append result (format ()
"Cover types~{ ~S~} were incorrectly classified as belonging to this class.  "
                 incorrect-matching-c-t)))
        (setf result (string-append result (format ()
"No cover types were incorrectly classified as belonging to this class.  "))))
   (string-append result (format ()
"The system's classification performance score is ~,4F"
           (get.value 'training.databases 'performance.score)))))

(defun display-next-learning-results ()
"Displays the next set of learning results, if any."
  (let* ((old-num (get.value 'output.learning.results 'current.class.number))
         (new-num (if (= (- (length (get.values 'training.databases
                                               'current.classes))
                            old-num)
                         1)  ;Was displaying last class
                      0     ;Display first class again
                      (1+ old-num)))) ;Change to next class
    (put.value 'output.learning.results 'current.class.number new-num)
    (display-learning-results new-num)))

(defun display-previous-learning-results ()
"Displays the previous set of learning results, if any."
  (let* ((old-num (get.value 'output.learning.results 'current.class.number))
         (new-num (if (> old-num 0)   ;Was not displaying first class
                      (1- old-num)    ;Change to previous class
                      (1- (length (get.values 'training.databases
                                             'current.classes))))))
                          ;Change to last class
    (put.value 'output.learning.results 'current.class.number new-num)
    (display-learning-results new-num)))
```

```
;;;-------------------------------------------------------------------------
;;; Methods to classify an unknown sample
;;;-------------------------------------------------------------------------

(defun get-scores-for-sample ()
"If the scores slot of the first training class unit is empty, print an error
message and return nil.  Otherwise call the function classify-sample to
classify the unknown sample and return the best class for the sample."
  (let ((current-samples (get.values 'estimate.hemispherical.reflectance
                                     'current.sample.wavelengths)))
    (when (or (null current-samples)
              (not (get.value 'learning.methods
                              'done.learn.class.descriptions.p)))
      (my-documentation-print
"Enter unknown sample and learn class descriptions before classifying the sample")
      (return-from get-scores-for-sample nil))
    (my-documentation-print "Classifying unknown sample")
    (put.value
        (classify-sample (get-sample-data current-samples) 2)
        'best.class t)
    (put.value 'learning.methods 'done.classify.sample.p t)
    (my-documentation-print "Finished classifying unknown sample")))

(defun get-sample-data (samples)
"Returns a list of the wavelengths and reflectance data for all the samples."
  (let ((result ()))
    (dolist (sam samples result)
      (push (list (get.value sam 'wavelength)
                  (get.value sam 'reflectance.data))
            result))))
```

```
(defun classify-sample (data option-num)
"Data is a list of wavelengths and corresponding directional reflectance data.
Classify the data by testing the best class hypotheses for each class on the
data and determining the matching score for the class. Return the best class
for the sample."
 (let ((best-score -2)
       (best-class nil))
  (dolist (problem (get.values 'training.databases 'current.classes))
   (let ((class-ev (get.values problem 'best.score))
         (e-sup 0)
         (e-opp 0))
      (dolist (ev class-ev)
       (let ((hyp-score (second ev))
             (total 0))
        (dolist (relation (first (first ev)))
         (let* ((rel (first (first relation)))
                (func (first rel))
                (arg1 (second rel))
                (arg2 (third rel))
                (flag (second (first relation)))
                (wave (second relation)))
           (dolist (dat data)
            (when (= wave (first dat))
             (incf total (funcall func (second dat)
                                  flag arg1 arg2))))))
        (if (= total (length (first (first ev))))
            (incf e-sup hyp-score)
            (incf e-opp hyp-score))))
      (let ((score (if (zerop (length class-ev))
                       0
                       (if (> e-sup e-opp)
                          (- 1 (/ e-opp e-sup))
                          (- (/ e-sup e-opp) 1)))))
        (when (= option-num 2)(put.value problem 'sample.score score))
        (when (> score best-score)
         (setf best-score score)
         (setf best-class problem)))))
   best-class))

;;;----------------------------------------------------------------------
;;; Methods to test class descriptions on cover types
;;;----------------------------------------------------------------------

(defun test-class-descriptions-on-cover-types ()
"Checks that the pre-requisite steps have been carried out - if not prints an
error message and exits the function."
  (when (not (get.value 'learning.methods 'done.learn.class.descriptions.p))
   (my-documentation-print
"Learn class descriptions before testing the class descriptions on the data base of cover types")
   (return-from test-class-descriptions-on-cover-types nil))
  (test-class-descriptions-on-cover-types-aux))
```

```
(defun test-class-descriptions-on-cover-types-aux()
"Tests the class definitions on the cover types contained in the positive
training sets for each posible class definition.  If the best class for the
sample is the class for which it is a positive training example, add 1 to
pos-total.  Return the score which is the proportion of correctly classified
samples.  In this version it is assumed that each class contains only one
wavelength."
  (let ((pos-total 0)
        (num-pos 0))
    (dolist (problem (get.values 'training.databases 'current.classes))
      (let ((training-data-sets (unit.children problem 'subclass)))
        (dolist (parent (unit.children (second training-data-sets)
                              `subclass))
          (let ((data ()))
            (dolist (unit (unit.children parent 'member))
              (push (list (get.value unit 'wavelength)
                          (get.value unit 'reflectance.data))
                    data))
            (incf num-pos)
            (let ((best-class (classify-sample data 3)))
              (cond ((eq best-class problem)
                     (incf pos-total)
                     (add.value best-class 'correct.matching.cover.types
                          (get.value parent 'cover.type)))
                    (t (add.value best-class 'incorrect.matching.cover.types
                          (get.value parent 'cover.type))))))))))
    (put.value 'training.databases 'performance.score
         (/ pos-total num-pos)))
  (put.value 'learning.methods 'done.classify.cover.types.p t))

(load "learn-methods1")

(load "learn-methods2")

(load "learn-methods3")
```

```
;;; learn-methods1.lisp
;;;
;;; Additional methods for constructing and testing compound hypotheses in the
;;; learning system
;;;
;;; Written by Ann and Patrick Harrison
;;; Created 30th July 1992
;;; Last modified18th September 1992

(in-package 'kee)

;;;-------------------------------------------------------------------------
;;;Additional methods for improved search technique
;;;-------------------------------------------------------------------------

(defun ask-about-trace-file ()
"Display the screen to ask the user whether or not the hypothesis testing
should be traced."
  (remove.all.values 'learning.methods 'yes.no)
  (remove.all.values 'learning.methods 'trace.file)
  (put.value 'learning.methods 'message
    "Do you want to trace the hypothesis testing?")
  (unitmsg 'viewport-learning.methods.1 'open-panel!))

(defun generalization-search ()
"Opens the trace file, if necessary and then calls the function to perform an
exhaustive search and testing of all hypotheses up to the depth specified."
   (let ((trace-file (get.value 'learning.methods 'trace.file))
         (current-classes (get.values 'training.databases 'current.classes)))
     (if trace-file
         (with-open-file (trace-str trace-file :direction :output
                                    :if-does-not-exist :create)
           (generalization-search-aux-1 current-classes trace-str))
         (generalization-search-aux-1 current-classes nil))))
```

```
(defun generalization-search-aux-1 (current-classes trace-str)
"Calls the function to test the single hypotheses. Then displays the interrupt
button and calls another function to test the compound hypotheses. At the end
of the search write the time to the trace file and removes the interrupt button
from the screen. "
  (let ((option-number (get.value 'learning.methods 'option.number)))
   (test-level-1 current-classes trace-str)
   (display-interrupt-button option-number)
   (catch 'search-over
    (generalization-search-aux-2
       current-classes
       (get.value 'learning.methods 'search.depth)
       trace-str))
   (when trace-str
    (multiple-value-bind (a b c d e f) (get-decoded-time)
       (princ (format ()
                     "~%Learning completed ~S/~S/~S at time ~S.~S.~S"
                     f e d c b a)
              trace-str)))
   (my-documentation-print "Finished learning class descriptions")
   (remove-interrupt-button option-number)))

(defun generalization-search-aux-2 (current-classes level str)
"Searches at increasingly deeper levels for the best hypothesis for each
problem. Completes the search at one level for all the problems before moving
to the next level. This is so that if the search is interrupted an
intermediate result for all problems is available."
 (let ((current-best-score nil))
  (when (> level 1)
   (dolist (problem current-classes)
       (setf current-best-score (second (get.value problem 'best.score)))
       (put.value problem 'previous.best.score current-best-score)
       (when (/= current-best-score 1)
        (test-level-2 problem current-best-score)
        (when str
         (princ-report 2 problem str))))
   (when (> level 2)
       (dolist (problem current-classes)
        (setf current-best-score (get-improved-score problem))
        (when current-best-score
         (test-level-3 problem current-best-score)
         (when str
          (princ-report 3 problem str))))
       (when (> level 3)
        (dolist (problem current-classes)
         (setf current-best-score (get-improved-score problem))
         (when current-best-score
          (test-level-4 problem current-best-score)
          (when str
              (princ-report 4 problem str))))
```

```
        (when (> level 4)
          (dolist (problem current-classes)
            (setf current-best-score (get-improved-score problem))
            (when current-best-score
                (test-level-5 problem current-best-score)
                (when str
                  (princ-report 5 problem str))))))))

(defun display-interrupt-button (option-number)
  (remove.all.values 'learning.methods 'search.over)
  (case option-number
    (1 (unitmsg 'windowpane-search.over-of-learning.methods.2 'open!))
    (2 (unitmsg 'windowpane-search.over-of-learning.methods.1 'open!))
    (3 (unitmsg 'windowpane-search.over-of-learning.methods.3 'open!))))

(defun remove-interrupt-button (option-number)
  (case option-number
    (1 (unitmsg 'windowpane-search.over-of-learning.methods.2 'close!))
    (2 (unitmsg 'windowpane-search.over-of-learning.methods.1 'close!))
    (3 (unitmsg 'windowpane-search.over-of-learning.methods.3 'close!))))

(defun princ-level-1-scores (problem str)
"Writes the level 1 scores for a problem to the trace file."
  (princ (format ()
                 "~%Problem ~S Level 1~%Best scores"
                 (unit.name problem)) str)
  (dolist (score (get.values problem 'scores))
    (princ (format () "~% ~S Overall score ~,4F Positive ~,4F Negative ~,4F"
                   (first score)(second score)(third score)(fourth score))
           str)))

(defun princ-report (level problem str)
"Writes the scores for one problem and level."
  (princ (format ()
                 "~%Problem ~S Level ~S~%Best scores"
                 (unit.name problem) level) str)
  (dolist (score (get.values problem 'best.score))
    (princ (format () "~% ~S Overall score ~,4F Positive ~,4F Negative ~,4F"
                   (first score)(second score)(third score)(fourth score))
           str)))

(defun get-improved-score (problem)
"Returns the current best score for the problem if it is at least 10% greater
than the previous best score and nil otherwise."
  (let ((previous-best-score (get.value problem 'previous.best.score))
        (current-best-score (second (get.value problem 'best.score))))
    (when (> (- current-best-score previous-best-score)
             (* previous-best-score 0.1))
      (put.value problem 'previous.best.score current-best-score)
      current-best-score)))
```

```
(defun test-level-1 (current-classes trace-str)
"Tests all the single (level 1) hypotheses for all classes.  Copies all
possible components of compound hypotheses in the HYPOTHESES slot of each
problem."
 (dolist (problem current-classes)
   (let ((scores nil))
     (dolist (hyp (get.values problem 'hypotheses))
         (multiple-value-bind (score-t score-nil)
            (apply #'score problem hyp)
          (push score-t scores)
          (push score-nil scores)))
     (setf scores (sort scores #'> :key #'second))
     (put.values problem 'hypotheses scores)
     (put.values problem 'scores
          (initial-get-best-scores (copy-tree scores) problem))
     (when trace-str
         (princ-level-1-scores problem trace-str))
     (initial-reduce-hypothesis-set problem))))

(defun test-level-2 (problem best-score-prev-level)
"Tests all the possible compund hypotheses cocnsiting of two single hypotheses anded together
(level 2 hypotheses) for a problem."
 (let ((hyps (get.values problem 'hypotheses)))
   (dolist (a hyps)
    (let ((a1 (first a)))
        (dolist (b (rest (member a hyps :test #'equal)))
          (when (get.value 'learning.methods 'search.over)
            (throw 'search-over))
          (test-hyp problem
                  (append a1 (first b)) best-score-prev-level))))))

(defun test-level-3 (problem best-score-prev-level)
"Tests all the possible level 3 hypotheses for a problem."
 (reduce-hypothesis-set problem)
 (let ((hyps (get.values problem 'hypotheses)))
   (dolist (a hyps)
    (let ((a1 (first a)))
        (dolist (b (rest (member a hyps :test #'equal)))
         (let ((b1 (first b)))
           (when (get.value 'learning.methods 'search.over)
             (throw 'search-over))
           (dolist (c (rest (member b hyps :test #'equal)))
             (test-hyp problem
                     (append a1 b1 (first c)) best-score-prev-level))))))))
```

```
(defun test-level-4 (problem best-score-prev-level)
"Tests all the possible level 4 hypotheses for a problem."
 (reduce-hypothesis-set problem)
 (let ((hyps (get.values problem 'hypotheses)))
  (dolist (a hyps)
   (let ((a1 (first a)))
      (dolist (b (rest (member a hyps :test #'equal)))
       (let ((b1 (first b)))
        (dolist (c (rest (member b hyps :test #'equal)))
         (let ((c1 (first c)))
            (when (get.value 'learning.methods 'search.over)
             (throw 'search-over))
            (dolist (d (rest (member c hyps :test #'equal)))
             (test-hyp problem
                    (append a1 b1 c1 (first d))
                    best-score-prev-level))))))))))

(defun test-level-5 (problem best-score-prev-level)
"Tests all the possible level 5 hypotheses for a problem."
 (reduce-hypothesis-set problem)
 (let ((hyps (get.values problem 'hypotheses)))
  (dolist (a hyps)
   (let ((a1 (first a)))
      (dolist (b (rest (member a hyps :test #'equal)))
       (let ((b1 (first b)))
        (dolist (c (rest (member b hyps :test #'equal)))
         (let ((c1 (first c)))
            (dolist (d (rest (member c hyps :test #'equal)))
             (let ((d1 (first d)))
              (when (get.value 'learning.methods 'search.over)
               (throw 'search-over))
              (dolist (e (rest (member d hyps :test #'equal)))
               (test-hyp problem
                       (append a1 b1 c1 d1 (first e))
                       best-score-prev-level))))))))))))

(defun test-hyp (problem this-hyp best-score-prev-level)
"Calls a function to test a hypothesis. Updates the best.score slot when
necessary. Only adds to the best score slot a score that is better than the
best score at the previous level."
  (let* ((this-score (funcall #'complex-score problem this-hyp))
         (this-actual-score (second this-score))
         (best-actual-score (second (get.value problem 'best.score))))
   (when (> this-actual-score best-score-prev-level)
    (cond ((> this-actual-score best-actual-score)
           (put.value problem 'best.score this-score)
           (setf best-actual-score this-actual-score))
          ((= this-actual-score best-actual-score)
           (add.value problem 'best.score this-score))))))
```

```
(defun initial-get-best-scores (scores problem)
"Puts all the relationships having the best actual score into the best.score
slot of the problem. Then returns the best number of scores."
   (let* ((best-hyp-and-score (first scores))
          (best-actual-score (second best-hyp-and-score)))
     (put.value problem 'best.score
          (convert-to-best-format best-hyp-and-score))
     (dolist (this-hyp (rest scores))
        (if (= best-actual-score (second this-hyp))
          (add.value problem 'best.score (convert-to-best-format this-hyp))
          (return-from nil))))
   (let ((required-num-scores (* (get.value 'training.databases 'num.scores)
                                 (length (get.values problem 'wavelength))))
          (actual-num-scores (length scores)))
     (if (<= actual-num-scores required-num-scores)
          scores
          (butlast scores (- actual-num-scores required-num-scores)))))

(defun convert-to-best-format (single-hyp)
"Converts the format of a single hypothesis to the format consistent with the
way compound hypotheses will be stored in the best.score slot."
  `((,(first single-hyp) t) ,@(rest single-hyp)))

(defun initial-reduce-hypothesis-set (problem)
"This heuristic function reduces the list of possible hypotheses. A compound
hypothesis cannot have a discrimination score that is greater than the minimum
of its components' positive training set scores so all hypotheses with positive
score less than the current best score are removed. Also, if a hypotheses
scores 1 for both the positive and negative training sets it does not
discriminate at all and it is removed from the hypothesis set. If a hypothesis
scores 0 for the negative training set, combining it with other hypotheses
cannot reduce this value so it is removed from the set of hypotheses."
  (let ((best-actual-score (second (get.value problem 'best.score))))
    (dolist (this-hyp (get.values problem 'hypotheses))
     (let ((this-pos-score (third this-hyp))
           (this-neg-score (fourth this-hyp)))
        (when (or (<= this-pos-score best-actual-score) ;Cannot be better
                 (and (= this-pos-score 1)          ;Hyp true for all +ve
                    (= this-neg-score 1))          ;and -ve training set
                 (zerop this-neg-score))           ;Cannot be better
          (remove.value problem 'hypotheses this-hyp))))))

(defun reduce-hypothesis-set (problem)
"This heuristic function reduces the list of possible hypotheses. A compound
hypothesis cannot have a discrimination score that is greater than the minimum
of its components' positive training set score so all hypotheses with positive
score less than the current best score are removed. This function is called
after the level 2 and subsequent level searches."
  (let ((best-actual-score (second (get.value problem 'best.score))))
    (dolist (this-hyp (get.values problem 'hypotheses))
     (when (< (third this-hyp) best-actual-score)    ;Cannot be better
          (remove.value problem 'hypotheses this-hyp)))))
```

```
(defun complex-score (problem data)
"Calculates the discrimination score for a relationship.  For example it
might test ((GREATER-THAN (60 180)(20 0)) T) and return the scores for this
hypothesis."
    (let ((training-data-sets (unit.children problem 'subclass)))
        (multiple-value-bind (num-pos pos-total)
          (complex-score-aux (second training-data-sets) data)
         (multiple-value-bind (num-neg neg-total)
           (complex-score-aux (first training-data-sets) data)
          (let ((pos-score (/ pos-total num-pos))
                 (neg-score (/ neg-total num-neg)))
           (make-complex-result data t
                                 (- pos-score neg-score)
                                 pos-score
                                 neg-score))))))


(defun complex-score-aux (training-set data)
"Returns the number of elements in the training set and the number that
matched the hypothesis."
  (let ((total 0)
        (num 0))
    (dolist (parent (unit.children training-set `subclass))
      (incf num)
      (incf total (find-score data parent)))
    (values num total)))

(defun find-score (data parent)
"Tests whether a training set member  matches a compound hypothesis - ie all
the single hypotheses in the compound hypothesis are true for the reflectance
data in the training set member.  Returns 1 if the data matches the hypothesis
and nil otherwise."
  (let ((total 0))
    (dolist (dat data)
      (let* ((wave (second dat))
             (rel (first dat))
             (func (first (first rel)))
             (arg1 (second (first rel)))
             (arg2 (third (first rel)))
             (old-flag (second rel)))
        (dolist (unit (unit.children parent 'member))
          (when (= wave (get.value unit 'wavelength))
            (incf total
                  (funcall func (get.value unit 'reflectance.data)
                               old-flag arg1 arg2))))))
    (if (= total (length data))
        1
        0)))

(defun make-complex-result (data flag s1 s2 s3)
"Puts the elements of a hypothesis into the required form."
  `((,data ,flag) ,s1 ,s2 ,s3))
```

```
;;; learn-methods2.lisp
;;;
;;; Output of data from the Learning System to a file
;;;
;;; Written by Ann Harrison
;;; Created 11th September 1992
;;; Last Modified 21st September 1992

(in-package 'kee)

(defun ls-open-output-to-file-interface ()
"Opens the interface for outputting the results to a file."
  (remove.all.values 'output.learning.results 'output.parameters)
  (remove.all.values 'output.learning.results 'format.list)
  (put.value 'output.learning.results 'message
"Select the parameters to be output to the file, in the correct order =>")
  (unitmsg 'viewport-output.learning.results.2 'open-panel!))

(defun ls-open-output-to-file-template-interface ()
"Opens the window that allows the user to select the format for the
reflectance data."
  (put.value 'windowpane-output.parameters-of-output.learning.results.5
     'mouseleftfn!
     'deactivate-left-mouse)
  (remove.all.values 'output.learning.results 'template.number)
  (put.value 'output.learning.results 'message
     "Select the required template =>")
  (unitmsg 'windowpane-template.message-of-output.learning.results.8 'open!)
  (unitmsg 'windowpane-template.number-of-output.learning.results.7 'open!))

(defun ls-store-template (template)
"Stores the format corresponding to the specified format."
  (put.values 'output.learning.results 'format.list
     (case template
         (template.1 '(template.1))
         (template.2 '(results class.definition view.angles solar.zenith))
         (template.3 '(results class.definition training.sets view.angles
                    solar.zenith)))))

(defun ls-write-results-to-file ()
"Writes the results to a file in the specified format."
  (let ((format-list (reverse (get.values 'output.learning.results
                                   'format.list))))
    (with-open-file (out-str (get.value '9.output 'output.file.name)
                        :direction :output
                        :if-does-not-exist :create)
      (if (eq (first format-list) 'template.1)
          (ls-output-data-to-file
           (get.values 'training.databases
                 'current.classes)
           out-str)
          (ls-write-results-to-file-aux out-str format-list)))))
```

```
(defun ls-write-results-to-file-aux (out-str format-list)
  (dolist (class (get.values 'training.databases
                  'current.classes))
    (dolist (parameter format-list)
      (if (eq parameter 'done)
        nil
        (ls-write-simple-results-to-file out-str class parameter)))))

(defun ls-write-simple-results-to-file (out-str class parameter)
 (case parameter
  (solar.zenith (princ
                 (get.value class 'solar.zenith) out-str))
  (wavelength (princ
               (get.values class 'wavelength) out-str))
  (view.angles (princ (get.values class 'view.angle.data.message) out-str))
  (class.definition (princ (get.value class 'class) out-str))
  (training.sets (ls-princ-training-sets class out-str))
  (results (ls-write-results class out-str)))
 (princ " " out-str))

(defun ls-write-results (class out-str)
 (dolist (score (get.values class 'scores))
  (let ((sc (first (first score))))
   (princ (format () "~S score ~,4F at wavelength ~S "
                  (first sc)
                  (second score)
                  (second sc)) out-str)))
 (dolist (score (get.values class 'best.score))
  (princ (format () "~S with score ~,4F  "
                 (first score)(second score)) out-str))
 (let ((sample-score (get.value class 'sample.score)))
  (when sample-score
   (let ((this-class (get.value class 'class)))
     (princ (format ()
                    "Score for unknown sample for class ~S is ~,4F "
                    this-class
                    sample-score) out-str)
     (when (get.value class 'best.class)
      (princ (format ()
                     "The class ~S is the best class for the sample "
                     this-class) out-str)))))
 (let ((performance-score
        (get.value 'training.databases 'performance.score)))
  (when performance-score
   (princ (format ()
                  "The system's classification performance score was ~,4F "
                  performance-score)
                  out-str))))
```

```
(defun ls-princ-training-sets (class out-str)
 (princ (format ()
        "Positive training set is:-~{ ~S~} Negative training set is:-~{ ~S~} "
        (get-unit-names (get.values class 'pos.training.set))
        (get-unit-names (get.values class 'neg.training.set)))
        out-str)
 (when (= (get.value 'learning.methods 'option.number) 3)
  (let ((correct-matching-c-t
          (get-unit-names
           (get.values class 'correct.matching.cover.types)))
         (incorrect-matching-c-t
          (get-unit-names
           (get.values class 'incorrect.matching.cover.types))))
    (if correct-matching-c-t
         (princ (format ()
"Cover types~{ ~S~} were correctly classified as belonging to this class "
            correct-matching-c-t) out-str)
         (princ
"No cover types were correctly classified as belonging to this class "
       out-str))
    (if incorrect-matching-c-t
         (princ (format ()
"Cover types~{ ~S~} were incorrectly classified as belonging to this class"
            correct-matching-c-t) out-str)
         (princ
"No cover types were incorrectly classified as belonging to this class"
       out-str)))))

(defun ls-output-data-to-file (classes out-str)
 (let ((class (first classes)))
   (princ (format () "~%Solar Zenith Angle:- ~S"
                  (get.value class 'solar.zenith))
          out-str)
   (terpri out-str)
   (princ "View Angle Data:- " out-str)
   (dolist (mess (get.values class 'view.angle.data.message))
    (terpri out-str)
    (princ mess out-str)))
  (dolist (class classes)
   (let ((class-def (get.value class 'class)))
    (princ (format () "~2%Class Definition:- ~S"
                   class-def)
           out-str)
    (ls-get-scores (get.values class 'scores) class-def class out-str))))
```

```
(defun ls-get-scores (scores class problem out-str)
"Writes to the file the classification scores for a class.  If option 2 has
been selected it also writes the class score for the unknown sample and
indicates if this class is the best class for the sample.  If option 3 has
been selected it also writes the classification performance score for
the class operating on the data base of cover type data."
  (princ (format () "~%Positive training set is:-~{ ~S~}"
                 (get-unit-names
                  (get.values problem 'pos.training.set)))
         out-str)
  (princ (format () "~%Negative training set is:-~{ ~S~}"
                 (get-unit-names
                  (get.values problem 'neg.training.set)))
         out-str)
  (princ (format () "~%Best single hypothesis scores are:-") out-str)
  (dolist (score scores)
   (let ((sc (first (first score))))
    (princ (format () "~%~S score ~,4F at wavelength ~S"
                   (first sc)
                   (second score)
                   (second sc)) out-str)))
  (princ (format () "~%Best compound hypothesis scores are:-")  out-str)
  (dolist (score (get.values problem 'best.score))
   (princ (format () "~%~S with score ~,4F  "
                  (first score)(second score)) out-str))
  (let ((sample-score (get.value problem 'sample.score)))
   (when sample-score
    (ls-option-2-result problem class sample-score out-str)))
  (let ((performance-score
         (get.value 'training.databases 'performance.score)))
   (when performance-score
    (ls-option-3-result problem performance-score out-str))))

(defun ls-option-2-result (problem class sample-score out-str)
"Write to the file the results that apply to option 2 only."
  (princ (format ()
                 "~%Score for unknown sample for class ~S is ~,4F"
                 class
                 sample-score) out-str)
  (when (get.value problem 'best.class)
   (princ (format ()
                  "~%The class ~S is the best class for this unknown sample"
                  class) out-str)))
```

```
(defun ls-option-3-result (problem performance-score out-str)
"Writes to the file the results that apply to option 3 only."
 (let ((correct-matching-c-t
         (get-unit-names
          (get.values problem 'correct.matching.cover.types)))
        (incorrect-matching-c-t
         (get-unit-names
          (get.values problem 'incorrect.matching.cover.types))))
   (if correct-matching-c-t
        (princ (format ()
"~%Cover types~{ ~S~} were correctly classified as belonging to this class."
               correct-matching-c-t) out-str)
        (princ (format ()
"~%No cover types were correctly classified as belonging to this class.")
             out-str))
   (if incorrect-matching-c-t
        (princ (format ()
"~%Cover types~{ ~S~} were incorrectly classified as belonging to this class."
              incorrect-matching-c-t) out-str)
        (princ (format ()
"~%No cover types were incorrectly classified as belonging to this class.")
             out-str))
   (princ (format ()
"~%The system's classification performance score is ~,4F"
        performance-score)
          out-str)))
```

```
;;; learn-methods3.lisp
;;;
;;; Code for the Learning System - Automatic Mode
;;;
;;; Written by Ann and Patrick Harrison
;;; Created 18th September 1992
;;; Last Modified 25th September 1992

(in-package 'kee)

(defun auto-enter-classes ()
 (clear-prompt)
 (wipe-out-results 'dummy)
 (unitmsg 'windowpane-wavelength-of-enter.learning.data.1 'close!)
 (unitmsg 'windowpane-solar.zenith-of-enter.learning.data.2 'close!)
 (unitmsg 'windowpane-view.angle.data.message-of-enter.learning.data.4
         'close!)
 (remove.all.values 'enter.learning.data 'class)
 (remove.all.values 'enter.learning.data 'class.parameter)
 (put.value 'enter.learning.data 'menu 'enter.data)
 (unitmsg 'viewport-enter.learning.data.2 'open-panel!))

(defun auto-learn-class-descriptions (input-file output-file classes)
 (remove.all.values 'estimate.hemispherical.reflectance 'new.samples)
 (catch 'invalid-data
  (input-data-from-file input-file)
  (with-open-file (output-str output-file :direction :output
                              :if-does-not-exist :create)
    (dolist (sample (get.values 'estimate.hemispherical.reflectance
                         'new.samples))
         (auto-learn-class-descriptions-aux output-str classes sample)
         (dolist (uni (unit.children sample 'member))
          (delete.unit uni))
         (delete.unit sample)))
  (initialize-auto-system)))

(defun auto-learn-class-descriptions-aux (output-str classes sample)
 (let ((wavelength-units (unit.children sample 'member)))
  (complete-training-classes classes sample wavelength-units)
  (catch 'no-training-sets
   (dolist (problem classes)
        (generate-training-sets problem)
        (interp-extrap-training-data problem))
   (forward.chain 'learning.rules)
   (generalization-search-aux-1 classes nil)
   (put.value (classify-sample (get-sample-data wavelength-units) 2)
        'best.class t)
   (test-class-descriptions-on-cover-types-aux)
   (ls-output-data-to-file classes output-str))))
```

```
(defun complete-training-classes (classes sample wavelength-units)
  (remove.all.values 'enter.learning.data 'view.angle.data)
  (remove.all.values 'enter.learning.data 'view.angle.data.message)
  (get-view-angle-data wavelength-units)
  (let* ((view-angle-data
           (get.values 'enter.learning.data 'view.angle.data))
          (view-angle-data-message
           (get.values 'enter.learning.data 'view.angle.data.message))
          (solar-zenith (get.value sample 'solar.zenith))
          (wavelengths (mapcar #'first view-angle-data)))
    (dolist (problem classes)
      (put.values problem 'view.angle.data view-angle-data)
      (put.values problem 'view.angle.data.message
           view-angle-data-message)
      (put.value problem 'solar.zenith solar-zenith)
      (put.values problem 'wavelength wavelengths)
      (initialize-training-set problem)
      (delete-sub-units problem)))
   (remove.all.values 'training.databases 'performance.score))
```

APPENDIX B

TEST RESULTS

test1-trace

Problem TRAINING.PROB.387 Level 1
Best scores
((((GREATER-THAN (15 182) (75 90)) T) 0.68)) Overall score 0.7738 Positive 0.9167 Negative 0.1429
((((FIRST-MAX (15 182)) T) 0.68)) Overall score 0.7738 Positive 0.9167 Negative 0.1429
((((FIRST-MAX (75 90)) NIL) 0.68)) Overall score 0.7738 Positive 0.9167 Negative 0.1429
((((GREATER-THAN (75 90) (0 0)) NIL) 0.68)) Overall score 0.7500 Positive 0.7500 Negative 0.0000
((((GREATER-THAN (75 90) (35 45)) NIL) 0.68)) Overall score 0.5833 Positive 0.5833 Negative 0.0000

Problem TRAINING.PROB.387 Level 2
Best scores
(((((FIRST-MAX (75 90)) NIL) 0.68)) T) Overall score 0.7738 Positive 0.9167 Negative 0.1429
(((((FIRST-MAX (15 182)) T) 0.68)) T) Overall score 0.7738 Positive 0.9167 Negative 0.1429
(((((GREATER-THAN (15 182) (75 90)) T) 0.68)) T) Overall score 0.7738 Positive 0.9167 Negative 0.1429
Learning completed 1992/9/22 at time 12.58.57

test1-option1

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((15 182) (75 90) (0 0) (35 45))

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 CT9-46
Negative training set is:- CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT11-58 CT5-42 CT6-45 CT7-41 CT8-42 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (15 182) (75 90)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (15 182)) T) score 0.7738 at wavelength 0.68
((FIRST-MAX (75 90)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 90) (0 0)) NIL) score 0.7500 at wavelength 0.68
((GREATER-THAN (75 90) (35 45)) NIL) score 0.5833 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MAX (75 90)) NIL) 0.68)) T) with score 0.7738
(((((FIRST-MAX (15 182)) T) 0.68)) T) with score 0.7738
(((((GREATER-THAN (15 182) (75 90)) T) 0.68)) T) with score 0.7738

test2-trace

Problem TRAINING.PROB.3 Level 1
Best scores
((((GREATER-THAN (30 275) (50 45)) T) 0.68)) Overall score 0.6129 Positive 1.0000 Negative 0.3871
((((GREATER-THAN (30 275) (45 85)) T) 0.68)) Overall score 0.5346 Positive 0.8571 Negative 0.3226
((((GREATER-THAN (30 93) (2 45)) NIL) 0.68)) Overall score 0.5207 Positive 0.7143 Negative 0.1935
((((GREATER-THAN (45 270) (45 85)) T) 0.68)) Overall score 0.5023 Positive 0.8571 Negative 0.3548
((((GREATER-THAN (45 270) (60 355)) T) 0.68)) Overall score 0.5023 Positive 0.8571 Negative 0.3548
((((GREATER-THAN (60 275) (60 355)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (60 87) (60 355)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (30 93) (50 45)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (50 227) (60 355)) T) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161
((((GREATER-THAN (35 48) (0 0)) NIL) 0.68)) Overall score 0.4839 Positive 1.0000 Negative 0.5161

Problem TRAINING.PROB.2 Level 1
Best scores
((((GREATER-THAN (60 275) (60 87)) NIL) 0.68)) Overall score 0.6250 Positive 1.0000 Negative 0.3750
((((GREATER-THAN (60 87) (60 355)) T) 0.68)) Overall score 0.5000 Positive 1.0000 Negative 0.5000
((((GREATER-THAN (60 275) (45 85)) NIL) 0.68)) Overall score 0.4875 Positive 0.8000 Negative 0.3125
((((GREATER-THAN (60 87) (75 2)) T) 0.68)) Overall score 0.4875 Positive 0.8000 Negative 0.3125
((((GREATER-THAN (60 87) (65 40)) T) 0.68)) Overall score 0.4625 Positive 0.9000 Negative 0.4375
((((GREATER-THAN (45 270) (45 356)) T) 0.68)) Overall score 0.4375 Positive 1.0000 Negative 0.5625
((((GREATER-THAN (30 93) (2 45)) T) 0.68)) Overall score 0.4375 Positive 1.0000 Negative 0.5625
((((GREATER-THAN (75 2) (75 180)) NIL) 0.68)) Overall score 0.4375 Positive 1.0000 Negative 0.5625
((((GREATER-THAN (15 270) (60 355)) T) 0.68)) Overall score 0.4250 Positive 0.8000 Negative 0.3750
((((GREATER-THAN (45 85) (65 40)) T) 0.68)) Overall score 0.4250 Positive 0.8000 Negative 0.3750

Problem TRAINING.PROB.1 Level 1
Best scores
((((GREATER-THAN (75 2) (60 180)) T) 0.68)) Overall score 0.7333 Positive 1.0000 Negative 0.2667
((((GREATER-THAN (60 275) (50 227)) T) 0.68)) Overall score 0.7000 Positive 1.0000 Negative 0.3000

((((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) Overall score 0.6667 Positive 1.0000 Negative 0.3333
((((GREATER-THAN (75 2) (45 178)) T) 0.68)) Overall score 0.6667 Positive 1.0000 Negative 0.3333
((((GREATER-THAN (60 275) (60 87)) T) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (60 87) (30 93)) T) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (45 85) (60 355)) NIL) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (65 225) (75 2)) NIL) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (50 227) (75 2)) NIL) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667
((((GREATER-THAN (75 2) (30 180)) T) 0.68)) Overall score 0.6333 Positive 1.0000 Negative 0.3667

Problem TRAINING.PROB.3 Level 2
Best scores
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((FIRST-MAX (60 180)) NIL) 0.68)) T) Overall score 0.7742 Positive 1.0000 Negative 0.2258
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((SECOND-MAX (75 180)) NIL) 0.68)) T) Overall score 0.7742 Positive 1.0000 Negative 0.2258
Problem TRAINING.PROB.2 Level 2
Best scores
(((((GREATER-THAN (60 87) (60 355)) T) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) Overall score 0.8125 Positive 1.0000 Negative 0.1875
(((((GREATER-THAN (60 275) (60 87)) NIL) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) Overall score 0.8125 Positive 1.0000 Negative 0.1875

Problem TRAINING.PROB.1 Level 2
Best scores
(((((GREATER-THAN (60 275) (60 87)) T) 0.68) (((GREATER-THAN (45 270) (0 0)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
(((((GREATER-THAN (75 2) (60 180)) T) 0.68) (((GREATER-THAN (15 225) (15 182)) NIL) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
Learning completed 1992/9/28 at time 9.29.39

test2-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((60 275) (45 270) (30 275) (15 270) (60 87) (45 85) (30 93) (2 90) (65 225) (50 227) (35 220) (15 225) (65 40) (50 45) (35 48) (15 46) (2 45) (75 2) (75 180) (60 355) (60 180) (45 356) (45 178) (30 5) (30 180) (15 7) (15 182) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (DESCRIPTION WHEAT)
Positive training set is:- CT5-26 CT4-27 CT5-28 CT5-59 CT4-32 CT4-51 CT5-42
Negative training set is:- CT11-82 CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT8-56 CT11-58 CT2-50 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49 CT11-45

Best single hypothesis scores are:-
((GREATER-THAN (30 275) (50 45)) T) score 0.6129 at wavelength 0.68
((GREATER-THAN (30 275) (45 85)) T) score 0.5346 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) NIL) score 0.5207 at wavelength 0.68
((GREATER-THAN (45 270) (45 85)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (45 270) (60 355)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (60 275) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (30 93) (50 45)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (50 227) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (35 48) (0 0)) NIL) score 0.4839 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((FIRST-MAX (60 180)) NIL) 0.68)) T) with score 0.7742
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((SECOND-MAX (75 180)) NIL) 0.68)) T) with score 0.7742
Score for unknown sample for class (DESCRIPTION WHEAT) is -1.0000

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT2-28 CT2-30 CT3-27 CT3-63 CT3-35 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45
Negative training set is:- CT1-30 CT4-27 CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT4-32 CT4-51 CT1-45 CT5-42 CT6-45 CT7-41 CT9-46 CT10-49
Best single hypothesis scores are:-
((GREATER-THAN (60 275) (60 87)) NIL) score 0.6250 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.5000 at wavelength 0.68
((GREATER-THAN (60 275) (45 85)) NIL) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (75 2)) T) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (65 40)) T) score 0.4625 at wavelength 0.68
((GREATER-THAN (45 270) (45 356)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (75 2) (75 180)) NIL) score 0.4375 at wavelength 0.68
((GREATER-THAN (15 270) (60 355)) T) score 0.4250 at wavelength 0.68
((GREATER-THAN (45 85) (65 40)) T) score 0.4250 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 87) (60 355)) T) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
(((((GREATER-THAN (60 275) (60 87)) NIL) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 2) (60 180)) T) score 0.7333 at wavelength 0.68
((GREATER-THAN (60 275) (50 227)) T) score 0.7000 at wavelength 0.68
((GREATER-THAN (30 93) (60 355)) NIL) score 0.6667 at wavelength 0.68
((GREATER-THAN (75 2) (45 178)) T) score 0.6667 at wavelength 0.68
((GREATER-THAN (60 275) (60 87)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (60 87) (30 93)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (45 85) (60 355)) NIL) score 0.6333 at wavelength 0.68

((GREATER-THAN (65 225) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (50 227) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (75 2) (30 180)) T) score 0.6333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 275) (60 87)) T) 0.68) (((GREATER-THAN (45 270) (0 0)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (75 2) (60 180)) T) 0.68) (((GREATER-THAN (15 225) (15 182)) NIL) 0.68)) T) with score 0.9000
Score for unknown sample for class (DESCRIPTION FOREST) is 0.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample

test3-option3

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((60 275) (45 270) (30 275) (15 270) (60 87) (45 85) (30 93) (2 90) (65 225) (50 227) (35 220) (15 225) (65 40) (50 45) (35 48) (15 46) (2 45) (75 2) (75 180) (60 355) (60 180) (45 356) (45 178) (30 5) (30 180) (15 7) (15 182) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (DESCRIPTION WHEAT)
Positive training set is:- CT5-26 CT4-27 CT5-28 CT5-59 CT4-32 CT4-51 CT5-42
Negative training set is:- CT11-82 CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT8-56 CT11-58 CT2-50 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (30 275) (50 45)) T) score 0.6129 at wavelength 0.68
((GREATER-THAN (30 275) (45 85)) T) score 0.5346 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) NIL) score 0.5207 at wavelength 0.68
((GREATER-THAN (45 270) (45 85)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (45 270) (60 355)) T) score 0.5023 at wavelength 0.68
((GREATER-THAN (60 275) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (30 93) (50 45)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (50 227) (60 355)) T) score 0.4839 at wavelength 0.68
((GREATER-THAN (35 48) (0 0)) NIL) score 0.4839 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((FIRST-MAX (60 180)) NIL) 0.68)) T) with score 0.7742
(((((GREATER-THAN (30 275) (50 45)) T) 0.68) (((SECOND-MAX (75 180)) NIL) 0.68)) T) with score 0.7742
Score for unknown sample for class (DESCRIPTION WHEAT) is -1.0000
Cover types CT5-26 CT4-27 CT5-28 CT5-59 CT4-32 CT4-51 CT5-42 were correctly classified as belonging to this class.
Cover types CT7-23 CT2-28 CT3-27 CT3-35 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8400

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT2-28 CT2-30 CT3-27 CT3-63 CT3-35 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45
Negative training set is:- CT1-30 CT4-27 CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT4-32 CT4-51 CT1-45 CT5-42 CT6-45 CT7-41 CT9-46 CT10-49

Best single hypothesis scores are:-
((GREATER-THAN (60 275) (60 87)) NIL) score 0.6250 at wavelength 0.68
((GREATER-THAN (60 87) (60 355)) T) score 0.5000 at wavelength 0.68
((GREATER-THAN (60 275) (45 85)) NIL) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (75 2)) T) score 0.4875 at wavelength 0.68
((GREATER-THAN (60 87) (65 40)) T) score 0.4625 at wavelength 0.68
((GREATER-THAN (45 270) (45 356)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (30 93) (2 45)) T) score 0.4375 at wavelength 0.68
((GREATER-THAN (75 2) (75 180)) NIL) score 0.4375 at wavelength 0.68
((GREATER-THAN (15 270) (60 355)) T) score 0.4250 at wavelength 0.68
((GREATER-THAN (45 85) (65 40)) T) score 0.4250 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 87) (60 355)) T) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
(((((GREATER-THAN (60 275) (60 87)) NIL) 0.68) (((GREATER-THAN (30 93) (2 45)) T) 0.68)) T) with score 0.8125
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000
Cover types CT2-30 CT3-63 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8400

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 2) (60 180)) T) score 0.7333 at wavelength 0.68
((GREATER-THAN (60 275) (50 227)) T) score 0.7000 at wavelength 0.68
((GREATER-THAN (30 93) (60 355)) NIL) score 0.6667 at wavelength 0.68
((GREATER-THAN (75 2) (45 178)) T) score 0.6667 at wavelength 0.68
((GREATER-THAN (60 275) (60 87)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (60 87) (30 93)) T) score 0.6333 at wavelength 0.68
((GREATER-THAN (45 85) (60 355)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (65 225) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (50 227) (75 2)) NIL) score 0.6333 at wavelength 0.68
((GREATER-THAN (75 2) (30 180)) T) score 0.6333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 275) (60 87)) T) 0.68) (((GREATER-THAN (45 270) (0 0)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (75 2) (60 180)) T) 0.68) (((GREATER-THAN (15 225) (15 182)) NIL) 0.68)) T) with score 0.9000
Score for unknown sample for class (DESCRIPTION FOREST) is 0.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample
Cover types CT6-79 CT7-74 CT6-25 CT6-63 CT7-59 CT6-45 CT7-41 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8400

test4-run1-trace

Problem TRAINING.PROB.441 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) T) 0.91)) Overall score 0.7778 Positive 0.8889 Negative 0.1111
((((FIRST-MIN (60 315)) NIL) 0.91)) Overall score 0.7778 Positive 1.0000 Negative 0.2222
((((GREATER-THAN (60 315) (15 315)) T) 0.91)) Overall score 0.5556 Positive 0.6667 Negative 0.1111
((((GREATER-THAN (60 315) (0 0)) T) 0.91)) Overall score 0.5556 Positive 0.5556 Negative 0.0000

Problem TRAINING.PROB.440 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) NIL) 0.91)) Overall score 0.7778 Positive 0.8889 Negative 0.1111
((((FIRST-MIN (60 315)) T) 0.91)) Overall score 0.7778 Positive 0.7778 Negative 0.0000
((((GREATER-THAN (60 315) (15 315)) NIL) 0.91)) Overall score 0.5556 Positive 0.8889 Negative 0.3333
((((GREATER-THAN (60 315) (0 0)) NIL) 0.91)) Overall score 0.5556 Positive 1.0000 Negative 0.4444
Learning completed 1992/9/22 at time 13.11.13

test4-run1-option3

Solar Zenith Angle:- 35
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28
Negative training set is:- CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) T) score 1.0000 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) T) score 0.7778 at wavelength 0.91
((FIRST-MIN (60 315)) NIL) score 0.7778 at wavelength 0.91
((GREATER-THAN (60 315) (15 315)) T) score 0.5556 at wavelength 0.91
((GREATER-THAN (60 315) (0 0)) T) score 0.5556 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) with score 1.0000
Cover types CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32
Negative training set is:- CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) NIL) score 1.0000 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) NIL) score 0.7778 at wavelength 0.91
((FIRST-MIN (60 315)) T) score 0.7778 at wavelength 0.91
((GREATER-THAN (60 315) (15 315)) NIL) score 0.5556 at wavelength 0.91
((GREATER-THAN (60 315) (0 0)) NIL) score 0.5556 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) with score 1.0000
Cover types CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 1.0000

test4-run2-trace

Problem TRAINING.PROB.515 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.6364 Positive 1.0000 Negative 0.3636
((((GREATER-THAN (60 315) (30 315)) T) 0.91)) Overall score 0.5864 Positive 0.9500 Negative 0.3636
((((SECOND-MAX (15 135)) NIL) 0.91)) Overall score 0.4955 Positive 0.9500 Negative 0.4545
((((FIRST-MAX (30 135)) NIL) 0.91)) Overall score 0.4682 Positive 0.6500 Negative 0.1818
((((FIRST-MIN (60 315)) NIL) 0.91)) Overall score 0.4545 Positive 1.0000 Negative 0.5455

Problem TRAINING.PROB.514 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 0.6364 Positive 0.6364 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) NIL) 0.91)) Overall score 0.5864 Positive 0.6364 Negative 0.0500
((((SECOND-MAX (15 135)) T) 0.91)) Overall score 0.4955 Positive 0.5455 Negative 0.0500
((((FIRST-MAX (30 135)) T) 0.91)) Overall score 0.4682 Positive 0.8182 Negative 0.3500
((((FIRST-MIN (60 315)) T) 0.91)) Overall score 0.4545 Positive 0.4545 Negative 0.0000

Problem TRAINING.PROB.515 Level 2
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7182 Positive 0.9000 Negative 0.1818

Problem TRAINING.PROB.514 Level 2
Best scores
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) Overall score 0.6364 Positive 0.6364 Negative 0.0000

Problem TRAINING.PROB.515 Level 3
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((GREATER-THAN (15 135) (30 135)) NIL) 0.91)) T) Overall score 0.7591 Positive 0.8500 Negative 0.0909
Learning completed 1992/9/22 at time 13.15.12

test4-run2-option3

Solar Zenith Angle:- 70
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-28 CT10-28 CT7-41 CT5-42 CT6-45 CT8-42 CT11-45 CT10-49 CT5-59 CT6-79 CT7-59 CT8-56 CT11-82 CT11-58 CT6-63 CT7-74 CT8-70 CT10-76 CT10-63 CT11-71
Negative training set is:- CT1-30 CT2-28 CT2-30 CT4-32 CT3-35 CT1-45 CT9-46 CT2-50 CT4-51 CT3-63 CT9-68
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) T) score 0.6364 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) T) score 0.5864 at wavelength 0.91
((SECOND-MAX (15 135)) NIL) score 0.4955 at wavelength 0.91
((FIRST-MAX (30 135)) NIL) score 0.4682 at wavelength 0.91
((FIRST-MIN (60 315)) NIL) score 0.4545 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((GREATER-THAN (15 135) (30 135)) NIL) 0.91)) T) with score 0.7591
Cover types CT5-28 CT10-28 CT7-41 CT5-42 CT6-45 CT8-42 CT11-45 CT10-49 CT5-59 CT6-79 CT7-59 CT8-56 CT11-82 CT11-58 CT6-63 CT7-74 CT8-70 CT10-76 CT10-63 CT11-71 were correctly classified as belonging to this class.
Cover types CT9-46 CT4-51 CT3-63 CT9-68 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8710

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT4-32 CT3-35 CT1-45 CT9-46 CT2-50 CT4-51 CT3-63 CT9-68
Negative training set is:- CT5-28 CT10-28 CT7-41 CT5-42 CT6-45 CT8-42 CT11-45 CT10-49 CT5-59 CT6-79 CT7-59 CT8-56 CT11-82 CT11-58 CT6-63 CT7-74 CT8-70 CT10-76 CT10-63 CT11-71
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) NIL) score 0.6364 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) NIL) score 0.5864 at wavelength 0.91
((SECOND-MAX (15 135)) T) score 0.4955 at wavelength 0.91
((FIRST-MAX (30 135)) T) score 0.4682 at wavelength 0.91
((FIRST-MIN (60 315)) T) score 0.4545 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) with score 0.6364
Cover types CT1-30 CT2-28 CT2-30 CT4-32 CT3-35 CT1-45 CT2-50 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8710

test4-run3-trace

Problem TRAINING.PROB.642 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.5789 Positive 1.0000 Negative 0.4211
((((FIRST-MIN (60 315)) NIL) 0.91)) Overall score 0.3684 Positive 1.0000 Negative 0.6316
((((FIRST-MIN (45 315)) T) 0.91)) Overall score 0.3333 Positive 0.3333 Negative 0.0000
((((GREATER-THAN (15 315) (0 0)) T) 0.91)) Overall score 0.2865 Positive 0.4444 Negative 0.1579
((((SECOND-MIN (30 315)) T) 0.91)) Overall score 0.2865 Positive 0.4444 Negative 0.1579

Problem TRAINING.PROB.641 Level 1
Best scores
((((GREATER-THAN (45 315) (0 0)) T) 0.91)) Overall score 0.6333 Positive 0.8333 Negative 0.2000
((((GREATER-THAN (45 315) (30 315)) T) 0.91)) Overall score 0.5600 Positive 1.0000 Negative 0.4400
((((GREATER-THAN (45 315) (15 135)) T) 0.91)) Overall score 0.5533 Positive 0.8333 Negative 0.2800
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.4400 Positive 1.0000 Negative 0.5600
((((GREATER-THAN (60 315) (30 315)) T) 0.91)) Overall score 0.4400 Positive 1.0000 Negative 0.5600

Problem TRAINING.PROB.640 Level 1
Best scores
((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (60 315) (30 315)) NIL) 0.91)) Overall score 0.7778 Positive 0.8889 Negative 0.1111
((((FIRST-MIN (60 315)) T) 0.91)) Overall score 0.7778 Positive 0.7778 Negative 0.0000
((((GREATER-THAN (60 315) (15 315)) NIL) 0.91)) Overall score 0.5556 Positive 0.8889 Negative 0.3333
((((GREATER-THAN (60 315) (0 0)) NIL) 0.91)) Overall score 0.5556 Positive 1.0000 Negative 0.4444

Problem TRAINING.PROB.642 Level 2
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) Overall score 0.7368 Positive 1.0000 Negative 0.2632
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7368 Positive 1.0000 Negative 0.2632

Problem TRAINING.PROB.641 Level 2
Best scores
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) Overall score 0.7133 Positive 0.8333 Negative 0.1200

Problem TRAINING.PROB.642 Level 3
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91) (((SECOND-MIN (15 135)) NIL) 0.91)) T) Overall score 0.8421 Positive 1.0000 Negative 0.1579

(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) Overall score 0.8421 Positive 1.0000 Negative 0.1579

Problem TRAINING.PROB.641 Level 3
Best scores
(((((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (0 0)) T) 0.91) (((GREATER-THAN (15 315) (30 135)) NIL) 0.91)) T) Overall score 0.7533 Positive 0.8333 Negative 0.0800
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (0 0)) NIL) 0.91)) T) Overall score 0.7533 Positive 0.8333 Negative 0.0800
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) Overall score 0.7533 Positive 0.8333 Negative 0.0800

Problem TRAINING.PROB.642 Level 4
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91) (((SECOND-MIN (15 135)) NIL) 0.91)) T) Overall score 0.8947 Positive 1.0000 Negative 0.1053
Learning completed 1992/9/22 at time 13.21.43

test4-run3-option3

Solar Zenith Angle:- 35
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT7-59 CT8-56 CT7-23 CT10-49 CT6-25 CT6-45 CT7-41 CT8-42 CT10-28
Negative training set is:- CT5-59 CT11-58 CT2-50 CT4-51 CT9-46 CT9-23 CT1-26 CT1-45 CT3-27 CT4-27 CT5-26 CT11-45 CT1-30 CT2-28 CT2-30 CT5-28 CT5-42 CT3-35 CT4-32
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) T) score 0.5789 at wavelength 0.91
((FIRST-MIN (60 315)) NIL) score 0.3684 at wavelength 0.91
((FIRST-MIN (45 315)) T) score 0.3333 at wavelength 0.91
((GREATER-THAN (15 315) (0 0)) T) score 0.2865 at wavelength 0.91
((SECOND-MIN (30 315)) T) score 0.2865 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91) (((SECOND-MIN (15 135)) NIL) 0.91)) T) with score 0.8947
Cover types CT7-59 CT8-56 CT7-23 CT10-49 CT6-25 CT6-45 CT7-41 CT8-42 CT10-28 were correctly classified as belonging to this class.
Cover types CT5-59 CT11-58 CT11-45 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8750

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT5-59 CT11-58 CT5-26 CT11-45 CT5-28 CT5-42
Negative training set is:- CT3-63 CT6-63 CT10-63 CT7-59 CT8-56 CT2-50 CT4-51 CT7-23 CT9-46 CT9-23 CT10-49 CT1-26 CT1-45 CT3-27 CT4-27 CT6-25 CT6-45 CT1-30 CT2-28 CT2-30 CT7-41 CT8-42 CT10-28 CT3-35 CT4-32

Best single hypothesis scores are:-
((GREATER-THAN (45 315) (0 0)) T) score 0.6333 at wavelength 0.91
((GREATER-THAN (45 315) (30 315)) T) score 0.5600 at wavelength 0.91
((GREATER-THAN (45 315) (15 135)) T) score 0.5533 at wavelength 0.91
((GREATER-THAN (60 315) (45 315)) T) score 0.4400 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) T) score 0.4400 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (0 0)) T) 0.91) (((GREATER-THAN (15 315) (30 135)) NIL) 0.91)) T) with score 0.7533
(((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (0 0)) NIL) 0.91)) T) with score 0.7533 (((((GREATER-THAN (45 315) (0 0)) T) 0.91) (((GREATER-THAN (45 315) (15 135)) T) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) with score 0.7533
Cover types CT5-26 CT5-28 CT5-42 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8750

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32
Negative training set is:- CT5-26 CT6-25 CT6-45 CT11-45 CT5-28 CT5-42 CT7-41 CT8-42 CT10-28
Best single hypothesis scores are:-
((GREATER-THAN (60 315) (45 315)) NIL) score 1.0000 at wavelength 0.91
((GREATER-THAN (60 315) (30 315)) NIL) score 0.7778 at wavelength 0.91
((FIRST-MIN (60 315)) T) score 0.7778 at wavelength 0.91
((GREATER-THAN (60 315) (15 315)) NIL) score 0.5556 at wavelength 0.91
((GREATER-THAN (60 315) (0 0)) NIL) score 0.5556 at wavelength 0.91
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) T) with score 1.0000
Cover types CT1-26 CT1-45 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT3-35 CT4-32 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8750

test4-run4-trace

Problem TRAINING.PROB.952 Level 1
Best scores
((((FIRST-MIN (45 315)) T) 0.91)) Overall score 0.4667 Positive 0.5000 Negative 0.0333
((((GREATER-THAN (45 315) (15 135)) NIL) 0.91)) Overall score 0.3750 Positive 0.8750 Negative 0.5000
((((GREATER-THAN (60 315) (45 315)) T) 0.91)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (45 315) (0 0)) NIL) 0.91)) Overall score 0.3417 Positive 0.8750 Negative 0.5333
((((FIRST-MIN (15 315)) T) 0.91)) Overall score 0.3250 Positive 0.6250 Negative 0.3000

Problem TRAINING.PROB.951 Level 1
Best scores
((((FIRST-MIN (45 315)) NIL) 0.91)) Overall score 0.4667 Positive 0.9667 Negative 0.5000
((((GREATER-THAN (45 315) (15 135)) T) 0.91)) Overall score 0.3750 Positive 0.5000 Negative 0.1250

((((GREATER-THAN (60 315) (45 315)) NIL) 0.91)) Overall score 0.3667 Positive 0.3667 Negative 0.0000
((((GREATER-THAN (45 315) (0 0)) T) 0.91)) Overall score 0.3417 Positive 0.4667 Negative 0.1250
((((FIRST-MIN (15 315)) NIL) 0.91)) Overall score 0.3250 Positive 0.7000 Negative 0.3750

Problem TRAINING.PROB.952 Level 2
Best scores
(((((GREATER-THAN (45 315) (15 135)) NIL) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91)) T) Overall score 0.6750 Positive 0.8750 Negative 0.2000

Problem TRAINING.PROB.951 Level 2
Best scores
(((((FIRST-MIN (45 315)) NIL) 0.91) (((FIRST-MIN (15 315)) NIL) 0.91)) T) Overall score 0.6667 Positive 0.6667 Negative 0.0000

Problem TRAINING.PROB.952 Level 3
Best scores
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) Overall score 0.7417 Positive 0.8750 Negative 0.1333
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7417 Positive 0.8750 Negative 0.1333
(((((GREATER-THAN (45 315) (15 135)) NIL) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) Overall score 0.7417 Positive 0.8750 Negative 0.1333

Problem TRAINING.PROB.951 Level 3
Best scores
(((((FIRST-MIN (45 315)) NIL) 0.91) (((FIRST-MIN (15 315)) NIL) 0.91)) T) Overall score 0.6667 Positive 0.6667 Negative 0.0000
Learning completed 1992/9/22 at time 14.18.15

test4-run4-option3

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.91 data is ((60 315) (45 315) (30 315) (15 315) (0 0) (15 135) (30 135))

Class Definition:- (HEIGHT.CM (1000 3000))
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((FIRST-MIN (45 315)) T) score 0.4667 at wavelength 0.91
((GREATER-THAN (45 315) (15 135)) NIL) score 0.3750 at wavelength 0.91
((GREATER-THAN (60 315) (45 315)) T) score 0.3667 at wavelength 0.91
((GREATER-THAN (45 315) (0 0)) NIL) score 0.3417 at wavelength 0.91
((FIRST-MIN (15 315)) T) score 0.3250 at wavelength 0.91

Best compound hypothesis scores are:-
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((SECOND-MAX (60 315)) NIL) 0.91)) T) with score 0.7417
(((((GREATER-THAN (60 315) (45 315)) T) 0.91) (((GREATER-THAN (45 315) (0 0)) NIL) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) with score 0.7417
(((((GREATER-THAN (45 315) (15 135)) NIL) 0.91) (((GREATER-THAN (60 315) (45 315)) T) 0.91) (((FIRST-MIN (30 315)) NIL) 0.91)) T) with score 0.7417
Cover types CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41 were correctly classified as belonging to this class.
Cover types CT8-70 CT9-68 CT11-71 CT5-26 CT10-63 CT10-28 CT3-35 CT11-58 CT4-51 CT5-42 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7368

Class Definition:- (HEIGHT.CM (0 1000))
Positive training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Negative training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Best single hypothesis scores are:-
((FIRST-MIN (45 315)) NIL) score 0.4667 at wavelength 0.91
((GREATER-THAN (45 315) (15 135)) T) score 0.3750 at wavelength 0.91
((GREATER-THAN (60 315) (45 315)) NIL) score 0.3667 at wavelength 0.91
((GREATER-THAN (45 315) (0 0)) T) score 0.3417 at wavelength 0.91
((FIRST-MIN (15 315)) NIL) score 0.3250 at wavelength 0.91
Best compound hypothesis scores are:-
(((((FIRST-MIN (45 315)) NIL) 0.91) (((FIRST-MIN (15 315)) NIL) 0.91)) T) with score 0.6667
Cover types CT11-82 CT10-76 CT1-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT4-32 CT8-56 CT2-50 CT1-45 CT8-42 CT9-46 CT10-49 CT11-45 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7368

test4-run5-trace

Problem TRAINING.PROB.1106 Level 1
Best scores
((((GREATER-THAN (30 45) (60 45)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((FIRST-MIN (60 45)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((SECOND-MIN (60 45)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((SECOND-MIN (60 315)) NIL) 0.68)) Overall score 0.8258 Positive 0.9091 Negative 0.0833
((((GREATER-THAN (30 315) (60 315)) NIL) 0.68)) Overall score 0.8182 Positive 0.8182 Negative 0.0000

Problem TRAINING.PROB.1105 Level 1
Best scores
((((GREATER-THAN (30 45) (60 45)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909
((((FIRST-MIN (60 45)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909
((((SECOND-MIN (60 45)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909
((((SECOND-MIN (60 315)) T) 0.68)) Overall score 0.8258 Positive 0.9167 Negative 0.0909

((((GREATER-THAN (30 315) (60 315)) T) 0.68)) Overall score 0.8182 Positive 1.0000 Negative 0.1818

Problem TRAINING.PROB.1106 Level 2
Best scores
(((((SECOND-MIN (60 315)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000 (((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000

Problem TRAINING.PROB.1105 Level 2
Best scores
(((((SECOND-MIN (30 45)) NIL) 0.68) (((GREATER-THAN (30 225) (60 315)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((SECOND-MIN (30 45)) NIL) 0.68) (((SECOND-MIN (30 315)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((GREATER-THAN (60 135) (60 225)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((SECOND-MIN (30 45)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

Problem TRAINING.PROB.1106 Level 3
Best scores
(((((SECOND-MIN (60 315)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 315)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((SECOND-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000

(((((SECOND-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((FIRST-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) Overall score 0.9091 Positive 0.9091 Negative 0.0000

Problem TRAINING.PROB.1105 Level 3
Best scores
(((((SECOND-MIN (30 45)) NIL) 0.68) (((GREATER-THAN (30 225) (60 315)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((SECOND-MIN (30 45)) NIL) 0.68) (((SECOND-MIN (30 315)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((GREATER-THAN (60 135) (60 225)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((SECOND-MIN (30 45)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
Learning completed 1992/9/22 at time 14.36.6

test4-run5-option3

Solar Zenith Angle:- 40
View Angle Data:-
At wavelength 0.68 data is ((0 0) (30 45) (60 45) (30 135) (60 135) (30 225) (60 225) (30 315) (60 315))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Negative training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Best single hypothesis scores are:-
((GREATER-THAN (30 45) (60 45)) NIL) score 0.8258 at wavelength 0.68
((FIRST-MIN (60 45)) NIL) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 45)) NIL) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 315)) NIL) score 0.8258 at wavelength 0.68
((GREATER-THAN (30 315) (60 315)) NIL) score 0.8182 at wavelength 0.68
Best compound hypothesis scores are:-
(((((SECOND-MIN (60 315)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 315)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 315)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091

(((((SECOND-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((SECOND-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091
(((((FIRST-MIN (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((FIRST-MIN (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((FIRST-MIN (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((SECOND-MIN (60 135)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((GREATER-THAN (60 135) (30 315)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 45) (60 45)) NIL) 0.68) (((FIRST-MAX (30 135)) NIL) 0.68)) T) with score 0.9091
Cover types CT5-26 CT6-25 CT8-56 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.9565

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Negative training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Best single hypothesis scores are:-
((GREATER-THAN (30 45) (60 45)) T) score 0.8258 at wavelength 0.68
((FIRST-MIN (60 45)) T) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 45)) T) score 0.8258 at wavelength 0.68
((SECOND-MIN (60 315)) T) score 0.8258 at wavelength 0.68
((GREATER-THAN (30 315) (60 315)) T) score 0.8182 at wavelength 0.68
Best compound hypothesis scores are:-
(((((SECOND-MIN (30 45)) NIL) 0.68) (((GREATER-THAN (30 225) (60 315)) T) 0.68)) T) with score 0.9091
(((((SECOND-MIN (30 45)) NIL) 0.68) (((SECOND-MIN (30 315)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((GREATER-THAN (60 135) (60 225)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (30 315) (60 315)) T) 0.68) (((SECOND-MIN (30 45)) NIL) 0.68)) T) with score 0.9091
Cover types CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46 were correctly classified as belonging to this class.
Cover types CT5-28 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.9565

test4-run6-trace

Problem TRAINING.PROB.1200 Level 1
Best scores
((((GREATER-THAN (0 0) (10 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((GREATER-THAN (0 0) (15 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((GREATER-THAN (10 0) (15 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((FIRST-MAX (0 0)) NIL) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000
((((FIRST-MIN (0 0)) T) 0.68)) Overall score 0.1818 Positive 0.1818 Negative 0.0000

Problem TRAINING.PROB.1199 Level 1
Best scores
((((GREATER-THAN (0 0) (10 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((GREATER-THAN (0 0) (15 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((GREATER-THAN (10 0) (15 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((FIRST-MAX (0 0)) T) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182
((((FIRST-MIN (0 0)) NIL) 0.68)) Overall score 0.1818 Positive 1.0000 Negative 0.8182

Problem TRAINING.PROB.1200 Level 2
Best scores
(((((FIRST-MIN (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((FIRST-MAX (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((FIRST-MIN (0 0)) T) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((FIRST-MAX (0 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((GREATER-THAN (10 0) (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((GREATER-THAN (0 0) (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000
(((((GREATER-THAN (0 0) (10 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 0.1818 Negative 0.0000

Problem TRAINING.PROB.1199 Level 2
Best scores
(((((FIRST-MIN (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((FIRST-MAX (15 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((FIRST-MIN (0 0)) NIL) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((FIRST-MAX (0 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((GREATER-THAN (10 0) (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((GREATER-THAN (0 0) (15 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
(((((GREATER-THAN (0 0) (10 0)) T) 0.68)) T) Overall score 0.1818 Positive 1.0000 Negative 0.8182
Learning completed 1992/9/22 at time 14.38.10

test4-run6-opt3

Solar Zenith Angle:- 40
View Angle Data:-
At wavelength 0.68 data is ((0 0) (10 0) (15 0))

Class Definition:- (GROUND.COVER (0.31 1))
Positive training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Negative training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Best single hypothesis scores are:-
((GREATER-THAN (0 0) (10 0)) NIL) score 0.1818 at wavelength 0.68
((GREATER-THAN (0 0) (15 0)) NIL) score 0.1818 at wavelength 0.68
((GREATER-THAN (10 0) (15 0)) NIL) score 0.1818 at wavelength 0.68
((FIRST-MAX (0 0)) NIL) score 0.1818 at wavelength 0.68
((FIRST-MIN (0 0)) T) score 0.1818 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (15 0)) NIL) 0.68)) T) with score 0.1818
(((((FIRST-MAX (15 0)) T) 0.68)) T) with score 0.1818
(((((FIRST-MIN (0 0)) T) 0.68)) T) with score 0.1818
(((((FIRST-MAX (0 0)) NIL) 0.68)) T) with score 0.1818
(((((GREATER-THAN (10 0) (15 0)) NIL) 0.68)) T) with score 0.1818
(((((GREATER-THAN (0 0) (15 0)) NIL) 0.68)) T) with score 0.1818
(((((GREATER-THAN (0 0) (10 0)) NIL) 0.68)) T) with score 0.1818
Cover types CT10-49 CT8-42 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.6087

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46
Negative training set is:- CT5-26 CT6-25 CT8-56 CT5-28 CT10-49 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 CT8-42
Best single hypothesis scores are:-
((GREATER-THAN (0 0) (10 0)) T) score 0.1818 at wavelength 0.68
((GREATER-THAN (0 0) (15 0)) T) score 0.1818 at wavelength 0.68
((GREATER-THAN (10 0) (15 0)) T) score 0.1818 at wavelength 0.68
((FIRST-MAX (0 0)) T) score 0.1818 at wavelength 0.68
((FIRST-MIN (0 0)) NIL) score 0.1818 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (15 0)) T) 0.68)) T) with score 0.1818
(((((FIRST-MAX (15 0)) NIL) 0.68)) T) with score 0.1818
(((((FIRST-MIN (0 0)) NIL) 0.68)) T) with score 0.1818
(((((FIRST-MAX (0 0)) T) 0.68)) T) with score 0.1818 (((((GREATER-THAN (0 0) (15 0)) T) 0.68)) T) with score 0.1818
(((((GREATER-THAN (0 0) (10 0)) T) 0.68)) T) with score 0.1818
Cover types CT1-26 CT3-27 CT4-27 CT1-30 CT2-28 CT2-30 CT2-50 CT4-51 CT1-45 CT3-35 CT4-32 CT9-46 were correctly classified as belonging to this class.
Cover types CT5-26 CT6-25 CT8-56 CT5-28 CT10-28 CT6-45 CT11-45 CT5-42 CT7-41 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.6087

test5-run1-trace

Problem TRAINING.PROB.278 Level 1
Best scores
((((GREATER-THAN (60 355) (45 225)) T) 0.68)) Overall score 0.8182 Positive 1.0000 Negative 0.1818
((((GREATER-THAN (40 5) (0 0)) T) 0.68)) Overall score 0.8182 Positive 1.0000 Negative 0.1818
((((GREATER-THAN (40 85) (60 355)) NIL) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (20 92) (45 43)) NIL) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (60 355) (60 48)) T) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (60 355) (15 222)) T) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (60 48) (45 225)) T) 0.68)) Overall score 0.7273 Positive 1.0000 Negative 0.2727
((((GREATER-THAN (65 90) (60 355)) NIL) 0.68)) Overall score 0.7182 Positive 0.9000 Negative 0.1818
((((GREATER-THAN (60 355) (60 230)) T) 0.68)) Overall score 0.7091 Positive 0.8000 Negative 0.0909
((((GREATER-THAN (20 92) (60 355)) NIL) 0.68)) Overall score 0.6364 Positive 1.0000 Negative 0.3636

Problem TRAINING.PROB.277 Level 1
Best scores
((((SECOND-MAX (60 230)) T) 0.68)) Overall score 0.4917 Positive 0.6250 Negative 0.1333
((((FIRST-MAX (75 225)) T) 0.68)) Overall score 0.4750 Positive 0.8750 Negative 0.4000
((((GREATER-THAN (65 90) (60 355)) T) 0.68)) Overall score 0.4667 Positive 1.0000 Negative 0.5333
((((FIRST-MIN (40 5)) T) 0.68)) Overall score 0.4333 Positive 0.5000 Negative 0.0667
((((GREATER-THAN (40 5) (60 48)) NIL) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (75 225) (45 225)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (60 48) (45 43)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (60 355) (45 225)) NIL) 0.68)) Overall score 0.3417 Positive 0.8750 Negative 0.5333
((((GREATER-THAN (60 48) (45 225)) NIL) 0.68)) Overall score 0.3417 Positive 0.8750 Negative 0.5333
((((GREATER-THAN (65 90) (40 85)) T) 0.68)) Overall score 0.3333 Positive 1.0000 Negative 0.6667

Problem TRAINING.PROB.276 Level 1
Best scores
((((GREATER-THAN (65 90) (30 220)) NIL) 0.68)) Overall score 0.8444 Positive 0.9000 Negative 0.0556
((((GREATER-THAN (65 90) (45 225)) NIL) 0.68)) Overall score 0.8333 Positive 1.0000 Negative 0.1667
((((GREATER-THAN (60 355) (15 42)) NIL) 0.68))Overall score 0.7889 Positive 0.9000 Negative 0.1111

((((GREATER-THAN (60 48) (30 220)) NIL) 0.68)) Overall score 0.7889 Positive 0.9000 Negative 0.1111
((((GREATER-THAN (20 92) (60 48)) T) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 355) (40 5)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 355) (20 0)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 355) (45 43)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 48) (0 0)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556
((((GREATER-THAN (60 230) (45 225)) NIL) 0.68)) Overall score 0.7444 Positive 0.8000 Negative 0.0556

Problem TRAINING.PROB.278 Level 2
Best scores
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

Problem TRAINING.PROB.277 Level 2
Best scores
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68)) T) Overall score 0.8000 Positive 1.0000 Negative 0.2000

Problem TRAINING.PROB.276 Level 2
Best scores
(((((GREATER-THAN (60 48) (30 220)) NIL) 0.68) (((FIRST-MIN (40 5)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((SECOND-MIN (40 85)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MIN (20 92)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MAX (60 355)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (40 5)) T) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 230)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (20 92) (40 5)) T) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (0 0)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (45 225)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (15 222)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 48)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 222)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (30 220)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (60 355)) T) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 48) (30 220)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 42)) NIL) 0.68)) T) Overall score 0.9000 Positive 0.9000 Negative 0.0000

Problem TRAINING.PROB.278 Level 3
Best scores
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) Overall score 0.9091 Positive 1.0000 Negative 0.0909

Problem TRAINING.PROB.277 Level 3
Best scores
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000

Problem TRAINING.PROB.277 Level 4
Best scores
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68) (((FIRST-MAX (45 225)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((FIRST-MIN (30 45)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 92) (20 0)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (75 225)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 0) (15 222)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) Overall score 0.9333 Positive 1.0000 Negative 0.0667

Learning completed 1992/9/18 at time 4.2.45

test5-run1-option2

Solar Zenith Angle:- 71
View Angle Data:-
At wavelength 0.92 data is ((0 0))
At wavelength 0.68 data is ((65 90) (40 85) (20 92) (60 355) (40 5) (20 0) (75 225) (60 48) (60 230) (45 43) (45 225) (30 45) (30 220) (15 42) (15 222) (0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT10-49 CT6-45 CT8-56 CT10-63 CT7-59 CT6-79 CT6-63 CT10-76 CT8-70 CT7-74
Negative training set is:- CT11-45 CT9-46 CT1-45 CT4-51 CT2-50 CT11-58 CT11-82 CT5-59 CT3-63 CT11-71 CT9-68

Best single hypothesis scores are:-
((GREATER-THAN (60 355) (45 225)) T) score 0.8182 at wavelength 0.68
((GREATER-THAN (40 5) (0 0)) T) score 0.8182 at wavelength 0.68
((GREATER-THAN (40 85) (60 355)) NIL) score 0.7273 at wavelength 0.68
((GREATER-THAN (20 92) (45 43)) NIL) score 0.7273 at wavelength 0.68
((GREATER-THAN (60 355) (60 48)) T) score 0.7273 at wavelength 0.68
((GREATER-THAN (60 355) (15 222)) T) score 0.7273 at wavelength 0.68
((GREATER-THAN (60 48) (45 225)) T) score 0.7273 at wavelength 0.68
((GREATER-THAN (65 90) (60 355)) NIL) score 0.7182 at wavelength 0.68
((GREATER-THAN (60 355) (60 230)) T) score 0.7091 at wavelength 0.68
((GREATER-THAN (20 92) (60 355)) NIL) score 0.6364 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 48) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (20 92) (45 43)) NIL) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.6 8)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (40 5) (0 0)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((SECOND-MAX (15 222)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 230) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (30 220) (0 0)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (75 225) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (20 92) (30 220)) NIL) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (40 85)) T) 0.68)) T) with score 0.9091

(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (65 90) (15 222)) T) 0.68)) T) with score 0.9091
(((((GREATER-THAN (60 355) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (15 222)) T) 0.68)) T) with score 0.9091
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 0.5000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT5-28 CT5-26 CT5-42 CT11-45 CT11-58 CT11-82 CT5-59 CT11-71
Negative training set is:- CT10-28 CT9-23 CT7-23 CT6-25 CT4-27 CT3-27 CT2-28 CT 1-26 CT4-32 CT3-35 CT2-30 CT1-30 CT8-42 CT7-41 CT10-49 CT9-46 CT6-45 CT1-45 CT8 -56 CT4-51 CT2-50 CT10-63 CT7-59 CT6-79 CT6-63 CT3-63 CT10-76 CT9-68 CT8-70 CT7 -74
Best single hypothesis scores are:-
((SECOND-MAX (60 230)) T) score 0.4917 at wavelength 0.68
((FIRST-MAX (75 225)) T) score 0.4750 at wavelength 0.68
((GREATER-THAN (65 90) (60 355)) T) score 0.4667 at wavelength 0.68
((FIRST-MIN (40 5)) T) score 0.4333 at wavelength 0.68
((GREATER-THAN (40 5) (60 48)) NIL) score 0.3667 at wavelength 0.68
((GREATER-THAN (75 225) (45 225)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (60 48) (45 43)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (60 355) (45 225)) NIL) score 0.3417 at wavelength 0.68
((GREATER-THAN (60 48) (45 225)) NIL) score 0.3417 at wavelength 0.68
((GREATER-THAN (65 90) (40 85)) T) score 0.3333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((GREATER-THAN (60 48) (45 43)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68) (((FIRST-MAX (45 225)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((FIRST-MIN (30 45)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T)
with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 92) (20 0)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((SECOND-MAX (75 225)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (20 0) (15 222)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (40 5) (45 43)) NIL) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333

(((((GREATER-THAN (65 90) (60 355)) T) 0.68) (((GREATER-THAN (40 5) (60 48)) NIL) 0.68) (((GREATER-THAN (75 225) (45 225)) T) 0.68) (((SECOND-MAX (65 90)) NIL) 0.68)) T) with score 0.9333
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is -1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT4-32 CT3-35 CT2-30 CT1-30 CT9-46 CT1-45 CT4-51 CT2-50 CT3-63 CT9-68
Negative training set is:- CT8-42 CT7-41 CT5-42 CT11-45 CT10-49 CT6-45 CT8-56 CT11-58 CT11-82 CT10-63 CT7-59 CT6-79 CT6-63 CT5-59 CT11-71 CT10-76 CT8-70 CT7-74
Best single hypothesis scores are:-
((GREATER-THAN (65 90) (30 220)) NIL) score 0.8444 at wavelength 0.68
((GREATER-THAN (65 90) (45 225)) NIL) score 0.8333 at wavelength 0.68
((GREATER-THAN (60 355) (15 42)) NIL) score 0.7889 at wavelength 0.68
((GREATER-THAN (60 48) (30 220)) NIL) score 0.7889 at wavelength 0.68
((GREATER-THAN (20 92) (60 48)) T) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 355) (40 5)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 355) (20 0)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 355) (45 43)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 48) (0 0)) NIL) score 0.7444 at wavelength 0.68
((GREATER-THAN (60 230) (45 225)) NIL) score 0.7444 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 48) (30 220)) NIL) 0.68) (((FIRST-MIN (40 5)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((SECOND-MIN (40 85)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MIN (20 92)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((FIRST-MAX (60 355)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (40 5)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 230)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (20 92) (40 5)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (0 0)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (45 225)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 5) (15 222)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (60 48)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 222)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (30 220)) NIL) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (40 85) (60 355)) T) 0.68)) T) with score 0.9000
(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 48) (30 220)) NIL) 0.68)) T) with score 0.9000

(((((GREATER-THAN (65 90) (30 220)) NIL) 0.68) (((GREATER-THAN (60 355) (15 42)
) NIL) 0.68)) T) with score 0.9000
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -0.9333

test5-run1-option3

The system's classification performance score is 0.8929

test5-run2-trace

Problem TRAINING.PROB.5 Level 1
Best scores
((((GREATER-THAN (60 275) (50 227)) T) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (45 85) (60 355)) NIL) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (75 2) (45 178)) T) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (60 355) (15 182)) T) 0.68)) Overall score 0.9412 Positive 1.0000 Negative 0.0588
((((GREATER-THAN (50 227) (60 355)) NIL) 0.68)) Overall score 0.8889 Positive 0.8889 Negative 0.0000
((((GREATER-THAN (60 275) (30 93)) T) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176
((((GREATER-THAN (60 275) (35 220)) T) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176
((((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176
((((GREATER-THAN (60 87) (30 93)) T) 0.68)) Overall score 0.8824 Positive 1.0000 Negative 0.1176

Problem TRAINING.PROB.4 Level 1
Best scores
((((GREATER-THAN (60 275) (60 355)) T) 0.68)) Overall score 0.5000 Positive 1.0000 Negative 0.5000
((((GREATER-THAN (65 40) (45 356)) T) 0.68)) Overall score 0.4333 Positive 1.0000 Negative 0.5667
((((GREATER-THAN (65 40) (30 5)) T) 0.68)) Overall score 0.4333 Positive 1.0000 Negative 0.5667
((((GREATER-THAN (60 355) (45 356)) T) 0.68)) Overall score 0.4333 Positive 1.0000 Negative 0.5667
((((FIRST-MAX (75 180)) T) 0.68)) Overall score 0.4167 Positive 0.7500 Negative 0.3333
((((FIRST-MIN (45 356)) T) 0.68)) Overall score 0.3750 Positive 0.3750 Negative 0.0000
((((GREATER-THAN (75 2) (60 180)) NIL) 0.68)) Overall score 0.3750 Positive 0.8750 Negative 0.5000
((((SECOND-MAX (60 180)) T) 0.68)) Overall score 0.3667 Positive 0.5000 Negative 0.1333
((((GREATER-THAN (65 225) (50 227)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333
((((GREATER-THAN (65 225) (65 40)) T) 0.68)) Overall score 0.3667 Positive 1.0000 Negative 0.6333

Problem TRAINING.PROB.3 Level 1
Best scores
((((GREATER-THAN (30 275) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (65 40) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (50 45) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((GREATER-THAN (15 46) (75 2)) T) 0.68)) Overall score 1.0000 Positive 1.0000 Negative 0.0000
((((FIRST-MIN (75 2)) T) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (60 275) (75 2)) T) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (65 40) (45 356)) NIL) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (65 40) (30 5)) NIL) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (35 48) (75 2)) T) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000
((((GREATER-THAN (75 2) (60 355)) NIL) 0.68)) Overall score 0.9167 Positive 0.9167 Negative 0.0000

Problem TRAINING.PROB.5 Level 2
Best scores
(((((GREATER-THAN (60 355) (0 0)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) Overall score 1.0000 Positive 1.0000 Negative 0.0000
"

Total of 91 similar compound hypotheses each with discrimination score 1.0000
"

(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) T) Overall score 1.0000 Positive 1.0000 Negative 0.0000

Problem TRAINING.PROB.4 Level 2
Best scores
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (60 355) (45 356)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (30 5)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (45 356)) T) 0.68)) T) Overall score 0.9000 Positive 1.0000 Negative 0.1000

Problem TRAINING.PROB.4 Level 3
Best scores
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (60 355) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) Overall score 0.9667 Positive 1.0000 Negative 0.0333
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (30 5)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) Overall score 0.9667 Positive 1.0000 Negative 0.0333
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) Overall score 0.9667 Positive 1.0000 Negative 0.0333
Learning completed 1992/9/21 at time 21.40.8

test5-run2-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.92 data is ((0 0))
At wavelength 0.68 data is ((60 275) (45 270) (30 275) (15 270) (60 87) (45 85) (30 93) (2 90) (65 225) (50 227) (35 220) (15 225) (65 40) (50 45) (35 48) (15 46) (2 45) (75 2) (75 180) (60 355) (60 180) (45 356) (45 178) (30 5) (30 180) (15 7) (15 182) (0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT10-28 CT10-63 CT7-59 CT6-63 CT8-56 CT10-49 CT8-42 CT7-41 CT6-45
Negative training set is:- CT5-59 CT5-28 CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT11-58 CT4-32 CT3-35 CT4-51 CT2-50 CT11-45 CT9-46 CT5-42 CT1-45
Best single hypothesis scores are:-
((GREATER-THAN (60 275) (50 227)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (45 85) (60 355)) NIL) score 0.9412 at wavelength 0.68
((GREATER-THAN (30 93) (60 355)) NIL) score 0.9412 at wavelength 0.68
((GREATER-THAN (75 2) (45 178)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (60 355) (15 182)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (50 227) (60 355)) NIL) score 0.8889 at wavelength 0.68
((GREATER-THAN (60 275) (30 93)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 275) (35 220)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (30 275) (50 45)) NIL) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 87) (30 93)) T) score 0.8824 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 355) (0 0)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (0 0)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (15 225) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (15 225) (60 355)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (65 40)) NIL) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (35 220) (65 40)) NIL) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (50 227) (65 40)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (50 227) (65 40)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (65 225) (75 2)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (65 225) (75 2)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (2 90) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (2 90) (60 355)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000

SYSTEMS INC

(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((SECOND-MIN (45 356)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (45 356) (15 7)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (60 355) (0 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (15 225) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (50 227) (65 40)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (65 225) (75 2)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 87) (30 93)) T) 0.68) (((GREATER-THAN (2 90) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (0 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (2 45)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (2 90)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (15 225)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (45 85)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (45 356) (30 5)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (60 275) (15 182)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (45 356) (15 7)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (75 2) (15 182)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (75 2) (30 180)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (15 46) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (50 45) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (50 45)) NIL) 0.68) (((GREATER-THAN (50 227) (75 2)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (35 220)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (35 220)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (30 93)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000

(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 355) (15 182)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 2) (45 178)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((SECOND-MIN (45 356)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (45 356) (15 7)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (60 355) (0 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (15 225) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (35 220) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (35 220) (65 40)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (50 227) (65 40)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (65 225) (75 2)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (2 90) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (60 275) (35 220)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (60 355) (15 182)) T) 0.68)) T) with score 1.0000

(((((GREATER-THAN (30 93) (60 355)) NIL) 0.68) (((GREATER-THAN (75 2) (45 178)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (45 85) (60 355)) NIL) 0.68) (((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 275) (30 180)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (50 227)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 275) (35 220)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (45 270) (30 275)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (45 85) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (60 87) (30 93)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 275) (50 45)) NIL) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 275) (50 227)) T) 0.68) (((GREATER-THAN (30 93) (60 355)) NIL) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 1.0000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT11-82 CT11-71 CT5-26 CT5-59 CT5-28 CT11-58 CT11-45 CT5-42
Negative training set is:- CT6-79 CT10-76 CT7-74 CT9-68 CT8-70 CT9-23 CT7-23 CT6-25 CT1-26 CT10-28 CT10-63 CT7-59 CT6-63 CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT8-56 CT4-32 CT3-35 CT4-51 CT2-50 CT10-49 CT9-46 CT8-42 CT7-41 CT6-45 CT1-45
Best single hypothesis scores are:-
((GREATER-THAN (60 275) (60 355)) T) score 0.5000 at wavelength 0.68
((GREATER-THAN (65 40) (45 356)) T) score 0.4333 at wavelength 0.68
((GREATER-THAN (65 40) (30 5)) T) score 0.4333 at wavelength 0.68

((GREATER-THAN (60 355) (45 356)) T) score 0.4333 at wavelength 0.68
((FIRST-MAX (75 180)) T) score 0.4167 at wavelength 0.68
((FIRST-MIN (45 356)) T) score 0.3750 at wavelength 0.68
((GREATER-THAN (75 2) (60 180)) NIL) score 0.3750 at wavelength 0.68
((SECOND-MAX (60 180)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (65 225) (50 227)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (65 225) (65 40)) T) score 0.3667 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (60 355) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) with score 0.9667
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (30 5)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) with score 0.9667
(((((GREATER-THAN (60 275) (60 355)) T) 0.68) (((GREATER-THAN (65 40) (45 356)) T) 0.68) (((GREATER-THAN (65 225) (60 180)) NIL) 0.68)) T) with score 0.9667
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is 1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT4-32 CT3-35 CT4-51 CT2-50 CT9-46 CT1-45
Negative training set is:- CT10-28 CT10-63 CT7-59 CT6-63 CT5-59 CT5-28 CT11-58 CT8-56 CT11-45 CT10-49 CT8-42 CT7-41 CT6-45 CT5-42
Best single hypothesis scores are:-
((GREATER-THAN (30 275) (75 2)) T) score 1.0000 at wavelength 0.68
((GREATER-THAN (65 40) (75 2)) T) score 1.0000 at wavelength 0.68
((GREATER-THAN (50 45) (75 2)) T) score 1.0000 at wavelength 0.68
((GREATER-THAN (15 46) (75 2)) T) score 1.0000 at wavelength 0.68
((FIRST-MIN (75 2)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (60 275) (75 2)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (65 40) (45 356)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (65 40) (30 5)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (35 48) (75 2)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 2) (60 355)) NIL) score 0.9167 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (15 46) (75 2)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (50 45) (75 2)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (65 40) (75 2)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (30 275) (75 2)) T) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -1.0000

test5-run2-option3

The system's classification performance score is 0.9655

test6-run1-auto

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((60 180) (60 0) (45 180) (45 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT2-28 CT2-30 CT3-27 CT3-63 CT3-35 CT8-56 CT11-58 CT2-50 CT8-42 CT11-45
Negative training set is:- CT1-30 CT4-27 CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT4-32 CT4-51 CT1-45 CT5-42 CT6-45 CT7-41 CT9-46 CT10-49
Best single hypothesis scores are:-
((SECOND-MAX (45 180)) T) score 0.2625 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.2000 at wavelength 0.68
((SECOND-MAX (60 180)) NIL) score 0.2000 at wavelength 0.68
((GREATER-THAN (60 0) (45 180)) NIL) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) NIL) score 0.1875 at wavelength 0.68
((FIRST-MIN (60 0)) T) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 180) (60 0)) T) score 0.1250 at wavelength 0.68
((FIRST-MAX (60 0)) NIL) score 0.1250 at wavelength 0.68
((SECOND-MIN (45 180)) NIL) score 0.1250 at wavelength 0.68
((SECOND-MIN (45 0)) T) score 0.1250 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (60 0)) T) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MIN (60 0)) T) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000
Cover types CT2-28 CT2-30 CT3-63 CT2-50 CT11-45 were correctly classified as belonging to this class.
No cover types were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7222

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41
Negative training set is:- CT11-82 CT10-76 CT8-70 CT9-68 CT11-71 CT1-26 CT5-26 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT8-42 CT9-46 CT10-49 CT11-45
Best single hypothesis scores are:-
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MIN (45 0)) NIL) score 0.4333 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) T) score 0.4000 at wavelength 0.68
((FIRST-MIN (60 0)) NIL) score 0.3667 at wavelength 0.68
((SECOND-MIN (60 0)) T) score 0.2500 at wavelength 0.68

((GREATER-THAN (60 180) (45 180)) T) score 0.2083 at wavelength 0.68
((FIRST-MAX (45 180)) NIL) score 0.2083 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.1833 at wavelength 0.68
((SECOND-MIN (45 180)) T) score 0.1833 at wavelength 0.68
((GREATER-THAN (45 180) (45 0)) T) score 0.1333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (60 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((FIRST-MIN (45 180)) NIL) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((SECOND-MAX (45 0)) NIL) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((FIRST-MIN (60 0)) NIL) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((SECOND-MIN (45 0)) NIL) 0.68)) T) with score 0.5000
Score for unknown sample for class (DESCRIPTION FOREST) is 1.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample
Cover types CT6-79 CT7-74 CT6-25 CT7-23 CT6-63 CT7-59 CT6-45 CT7-41 were correctly classified as belonging to this class.
Cover types CT3-27 CT3-35 CT8-56 CT11-58 CT8-42 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.7222

test6-run1-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.92 data is ((0 0))
At wavelength 0.68 data is ((60 180) (60 0) (45 180) (45 0))

Class Definition:- (DESCRIPTION GRASS)
Positive training set is:- CT3-63 CT3-27 CT2-30 CT2-28 CT11-58 CT8-56 CT3-35 CT2-50 CT11-45 CT8-42
Negative training set is:- CT10-28 CT10-63 CT7-59 CT6-63 CT5-59 CT5-28 CT4-27 CT1-30 CT4-32 CT4-51 CT10-49 CT9-46 CT7-41 CT6-45 CT5-42 CT1-45
Best single hypothesis scores are:-
((SECOND-MAX (45 180)) T) score 0.2625 at wavelength 0.68
((SECOND-MAX (60 180)) NIL) score 0.2000 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.2000 at wavelength 0.68
((FIRST-MIN (60 0)) T) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 0) (45 180)) NIL) score 0.1875 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) NIL) score 0.1875 at wavelength 0.68
((SECOND-MIN (45 180)) NIL) score 0.1250 at wavelength 0.68
((SECOND-MIN (45 0)) T) score 0.1250 at wavelength 0.68
((FIRST-MAX (60 0)) NIL) score 0.1250 at wavelength 0.68
((GREATER-THAN (60 180) (60 0)) T) score 0.1250 at wavelength 0.68

Best compound hypothesis scores are:-
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((GREATER-THAN (60 0) (45 0)) NIL) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MIN (60 0)) T) 0.68) (((GREATER-THAN (60 180) (45 180)) T) 0.68)) T) with score 0.2750
(((((FIRST-MIN (60 0)) T) 0.68) (((FIRST-MAX (45 180)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((FIRST-MAX (60 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((SECOND-MAX (60 180)) NIL) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((SECOND-MIN (45 0)) T) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) NIL) 0.68)) T) with score 0.2750
(((((SECOND-MAX (45 180)) T) 0.68) (((FIRST-MIN (60 0)) T) 0.68)) T) with score 0.2750
Score for unknown sample for class (DESCRIPTION GRASS) is -1.0000

Class Definition:- (DESCRIPTION FOREST)
Positive training set is:- CT6-79 CT7-74 CT7-23 CT6-25 CT7-59 CT6-63 CT7-41 CT6-45
Negative training set is:- CT11-82 CT10-76 CT11-71 CT9-68 CT8-70 CT9-23 CT5-26 CT1-26 CT10-28 CT10-63 CT5-59 CT5-28 CT4-27 CT3-63 CT3-27 CT2-30 CT2-28 CT1-30 CT11-58 CT8-56 CT4-32 CT3-35 CT4-51 CT2-50 CT11-45 CT10-49 CT9-46 CT8-42 CT5-42 CT1-45
Best single hypothesis scores are:-
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MIN (45 0)) NIL) score 0.4333 at wavelength 0.68
((GREATER-THAN (60 0) (45 0)) T) score 0.4000 at wavelength 0.68
((FIRST-MIN (60 0)) NIL) score 0.3667 at wavelength 0.68
((SECOND-MIN (60 0)) T) score 0.2500 at wavelength 0.68
((FIRST-MAX (45 180)) NIL) score 0.2083 at wavelength 0.68
((GREATER-THAN (60 180) (45 180)) T) score 0.2083 at wavelength 0.68
((SECOND-MIN (45 180)) T) score 0.1833 at wavelength 0.68
((FIRST-MAX (60 180)) T) score 0.1833 at wavelength 0.68
((GREATER-THAN (45 180) (45 0)) T) score 0.1333 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (60 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((FIRST-MIN (45 180)) NIL) 0.68)) T) with score 0.5000
(((((GREATER-THAN (60 0) (45 0)) T) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (60 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((SECOND-MAX (45 0)) NIL) 0.68)) T) with score 0.5000
(((((SECOND-MIN (45 0)) NIL) 0.68) (((GREATER-THAN (45 180) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((FIRST-MIN (60 0)) NIL) 0.68)) T) with score 0.5000

(((((FIRST-MIN (45 0)) T) 0.68) (((GREATER-THAN (60 0) (45 0)) T) 0.68)) T) with score 0.5000
(((((FIRST-MIN (45 0)) T) 0.68) (((SECOND-MIN (45 0)) NIL) 0.68)) T) with score 0.5000
Score for unknown sample for class (DESCRIPTION FOREST) is 1.0000
The class (DESCRIPTION FOREST) is the best class for this unknown sample

test6-run1-option3

The system's classification performance score is 0.7222

test6-run2-auto

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((75 0) (75 180) (60 0) (60 180) (45 0) (45 180) (30 0) (30 180) (15 0) (15 180) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT6-45 CT7-41 CT8-42 CT10-49
Negative training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT3-35 CT4-32 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT9-46 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (45 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (75 0) (60 180)) T) score 0.8889 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (0 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (45 0) (30 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (30 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (60 0) (15 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (45 0) (15 0)) T) score 0.8235 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (45 0) (30 0)) T) 0.68) (((GREATER-THAN (60 0) (15 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (0 0)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (15 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 0) (45 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 1.0000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample
Cover types CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT6-45 CT7-41 CT8-42 CT10-49 were correctly classified as belonging to this class.
Cover types CT11-82 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8966

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT11-82 CT11-71 CT5-26 CT5-28 CT5-59 CT11-58 CT5-42 CT11-45

Negative training set is:- CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT2-50 CT4-51 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (60 180)) NIL) score 0.5000 at wavelength 0.68
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MAX (60 180)) T) score 0.4583 at wavelength 0.68
((FIRST-MAX (75 180)) T) score 0.4167 at wavelength 0.68
((GREATER-THAN (60 180) (45 180)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (75 0) (60 0)) T) score 0.3417 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 0) (30 0)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 180) (60 180)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (45 0) (0 0)) NIL) score 0.3083 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((FIRST-MIN (75 0)) NIL) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (30 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (45 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (60 0)) T) 0.68)) T) with score 0.8083
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is 1.0000
Cover types CT5-26 CT5-28 CT5-59 CT11-58 CT5-42 CT11-45 were correctly classified as belonging to this class.
Cover types CT9-46 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8966

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 CT9-46
Negative training set is:- CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT11-58 CT5-42 CT6-45 CT7-41 CT8-42 CT10-49 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (60 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (15 0)) NIL) score 0.9167 at wavelength 0.68
((FIRST-MIN (75 0)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (0 0)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 0) (45 180)) NIL) score 0.7143 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) NIL) score 0.7143 at wavelength 0.68
Best compound hypothesis scores are:-
(((((FIRST-MIN (75 0)) T) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (15 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (30 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (45 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (60 0)) NIL) 0.68)) T) with score 0.9167
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -1.0000
Cover types CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 were correctly classified as belonging to this class.

Cover types CT11-71 were incorrectly classified as belonging to this class.
The system's classification performance score is 0.8966

test6-run2-option2

Solar Zenith Angle:- 45
View Angle Data:-
At wavelength 0.68 data is ((75 0) (75 180) (60 0) (60 180) (45 0) (45 180) (30 0) (30 180) (15 0) (15 180) (0 0))
At wavelength 0.92 data is ((0 0))

Class Definition:- (GROUND.COVER (0.71 1))
Positive training set is:- CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT6-45 CT7-41 CT8-42 CT10-49
Negative training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT5-28 CT5-59 CT3-35 CT4-32 CT11-58 CT2-50 CT4-51 CT1-45 CT5-42 CT9-46 CT11-45
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (45 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) T) score 0.9412 at wavelength 0.68
((GREATER-THAN (75 0) (60 180)) T) score 0.8889 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (0 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (45 0) (30 0)) T) score 0.8824 at wavelength 0.68
((GREATER-THAN (60 0) (30 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (60 0) (15 0)) T) score 0.8235 at wavelength 0.68
((GREATER-THAN (45 0) (15 0)) T) score 0.8235 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (45 0) (30 0)) T) 0.68) (((GREATER-THAN (60 0) (15 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (0 0)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (60 0) (15 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
(((((GREATER-THAN (75 0) (45 180)) T) 0.68) (((GREATER-THAN (45 0) (30 0)) T) 0.68)) T) with score 1.0000
Score for unknown sample for class (GROUND.COVER (0.71 1)) is 1.0000
The class (GROUND.COVER (0.71 1)) is the best class for this unknown sample

Class Definition:- (GROUND.COVER (0.31 0.7))
Positive training set is:- CT11-82 CT11-71 CT5-26 CT5-28 CT5-59 CT11-58 CT5-42 CT11-45
Negative training set is:- CT6-79 CT7-74 CT10-76 CT8-70 CT9-68 CT1-26 CT6-25 CT7-23 CT9-23 CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT6-63 CT7-59 CT10-63 CT10-28 CT3-35 CT4-32 CT8-56 CT2-50 CT4-51 CT1-45 CT6-45 CT7-41 CT8-42 CT9-46 CT10-49
Best single hypothesis scores are:-
((GREATER-THAN (75 0) (60 180)) NIL) score 0.5000 at wavelength 0.68
((FIRST-MIN (45 0)) T) score 0.4667 at wavelength 0.68
((SECOND-MAX (60 180)) T) score 0.4583 at wavelength 0.68
((FIRST-MAX (75 180)) T) score 0.4167 at wavelength 0.68
((GREATER-THAN (60 180) (45 180)) T) score 0.3667 at wavelength 0.68
((GREATER-THAN (75 0) (60 0)) T) score 0.3417 at wavelength 0.68
((FIRST-MIN (75 0)) NIL) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) T) score 0.3083 at wavelength 0.68

((GREATER-THAN (75 0) (30 0)) T) score 0.3083 at wavelength 0.68
((GREATER-THAN (75 180) (60 180)) T) score 0.3083 at wavelength 0.68

Best compound hypothesis scores are:-
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (30 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (45 0)) T) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((FIRST-MIN (75 0)) NIL) 0.68)) T) with score 0.8083
(((((GREATER-THAN (75 0) (60 180)) NIL) 0.68) (((GREATER-THAN (75 0) (60 0)) T) 0.68)) T) with score 0.8083
Score for unknown sample for class (GROUND.COVER (0.31 0.7)) is 1.0000

Class Definition:- (GROUND.COVER (0 0.3))
Positive training set is:- CT1-30 CT2-28 CT2-30 CT3-27 CT3-63 CT4-27 CT3-35 CT4-32 CT2-50 CT4-51 CT1-45 CT9-46
Negative training set is:- CT5-28 CT5-59 CT6-63 CT7-59 CT10-63 CT10-28 CT8-56 CT11-58 CT5-42 CT6-45 CT7-41 CT8-42 CT10-49 CT11-45
Best single hypothesis scores are:-
((FIRST-MIN (75 0)) T) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (60 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (45 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (15 0)) NIL) score 0.9167 at wavelength 0.68
((GREATER-THAN (75 0) (30 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (15 180)) NIL) score 0.7857 at wavelength 0.68
((GREATER-THAN (75 0) (0 0)) NIL) score 0.7738 at wavelength 0.68
((GREATER-THAN (75 0) (45 180)) NIL) score 0.7143 at wavelength 0.68
((GREATER-THAN (60 0) (15 180)) NIL) score 0.7143 at wavelength 0.68
Best compound hypothesis scores are:-
(((((GREATER-THAN (75 0) (15 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (30 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (45 0)) NIL) 0.68)) T) with score 0.9167
(((((GREATER-THAN (75 0) (60 0)) NIL) 0.68)) T) with score 0.9167
(((((FIRST-MIN (75 0)) T) 0.68)) T) with score 0.9167
Score for unknown sample for class (GROUND.COVER (0 0.3)) is -1.0000

test6-run2-option3

The system's classification performance score is 0.8966

JJM SYSTEMS INC

APPENDIX C

LISTING OF FUNCTIONS REQUIRED BY THE NEW VERSION OF THE BROWSER

```
;;;-----------------------------------------------------------------------
;;; Functions Required by the Browser
;;;-----------------------------------------------------------------------

(defun show (unit kb-name)
"Draws a graph consisting of the unit and its immediate children."
 (let ((unit (unit unit (get.value 'browser 'current.kb)))
       (kb (kb kb-name)))
   (put.value 'browser 'message "")
   (put.value 'browser 'current.unit unit)
   (graph-unit unit
               :unit-predicate
               #'(lambda (subunits)
                   (cond ((eq subunits unit) unit)
                         ((and (or (member unit
                                          (unit.parents subunits 'subclass))
                                   (member unit
                                          (unit.parents subunits 'member)))
                               (eq (unit.kb subunits) kb))
                          unit)
                         (t nil))))))

(defun down (unit)
"Calls the function show to draw the graph consisting of the unit's child and
its children, if any.  Allows the user to select which child's tree is to be
displayed from a menu.  If the unit has no children, the function get-tree is
called so that the user can select one of the top level units in the kb."
  (let* ((kb (get.value 'browser 'current.kb))
         (children (get-children unit kb))
         (len (length children)))
    (cond ((= len 0)
           (get-tree kb))
          ((= len 1)
           (show (first children) kb))
          (t (let* ((menu
                     (make-cascading-menu :pop-up children))
                    (choice (choose-from-menu menu :position
                                              (make-position :x 800 :y 400))))
               (show choice kb))))))
```

```
(defun up (unit)
"Calls the function show to graph the subtree including the parent of the unit
and all the other units that have the same parent.  If the unit has more than
one parent, it allows the user to select the required parent from a menu."
  (let* ((kb (get.value 'browser 'current.kb))
         (parents (get-parents unit kb)))
    (cond ((= (length parents) 1) (show (first parents) kb))
          (t
           (let ((menu
                  (make-cascading-menu :pop-up
                                       (mapcar #'(lambda (unit)
                                                   (unit.name unit))
                                               (get-parents unit kb)))))
             (let ((choice (choose-from-menu menu :position
                                             (make-position :x 800 :y 400))))
               (if choice
                   (show choice kb))))))))

(defun get-parents (unit kb)
"Returns a list of the parents, both member and subclass, of the unit.  Adds to
the list the unit that is at the top of the tree containing the unit in the kb.
If the unit has no parents, a list of the top level units in the kb is
returned."
  (let ((lis ())
        (tree-top (unit (get.value 'browser 'tree.top))))
    (dolist (n (append (unit.parents unit 'subclass)
                       (unit.parents unit 'member))
               (cond ((member tree-top lis) lis)
                     ((null lis)(get-top-level-units kb))
                     (t (cons tree-top lis))))
      (if (and n (eq (unit.kb n) (kb kb)))
          (push n lis)))))

(defun get-children (unit kb)
"Returns the children, both member and subclass, of a unit."
  (let ((children (mapcar #'(lambda (unit) (unit.name unit))
                          (append (unit.children unit 'subclass)
                                  (unit.children unit 'member))))
        (result ()))
    (dolist (child children result)
      (when (eq (unit.kb child) (kb kb))
        (push child result)))))
```

```
(defun show-slots (thisunit)
"Allows the user to select a slot from a menu and then displays the value of
the slot."
  (let ((lis (set-difference
              (unit.slot.names thisunit)
              (unit.slot.names 'classes))))
    (cond ((null lis)
           (put.value 'browser 'message (format ()
                  "The unit ~S has no slots" (unit.name thisunit)))
           nil)
          (t (let* ((men (make-cascading-menu :pop-up lis))
                    (choice (choose-from-menu men :position
                                              (make-position :x 800 :y 400))))
               (when (member choice lis)
                 (put.value 'browser 'message
                   (format () "~%The value of ~s = ~s"
                           choice
                           (get.values thisunit choice)))
                 (show-slots thisunit)))))))

(defun browse-kb ()
"Allows the user to select a different kb for the browser."
  (let* ((kbs (unit.children 'knowledgebases 'member))
         (menu
          (make-cascading-menu :pop-up
                               (mapcar #'(lambda (unit)
                                           (unit.name unit))
                                       kbs)))
         (choice (choose-from-menu menu :position
                                   (make-position :x 800 :y 400))))
    (when choice
      (put.value 'browser 'current.kb choice)
      (put.value 'browser 'top.level.units
            (get-top-level-units choice))
      (get-tree choice))))
```

```
(defun get-tree (kb)
"Allows the user to select which tree to be displayed if more than one tree is
available from the top level in the kb."
  (let ((possible-units (get-unit-names
                          (get.value 'browser 'top.level.units))))
    (cond ((null possible-units)
           (put.value 'browser 'message (format ()
          "It is not possible to browse the kb ~S. Select a different kb."
             kb))
           (browse-kb))
          ((= (length possible-units) 1)
           (let ((tree-top (first possible-units)))
             (put.value 'browser 'tree.top tree-top)
             (show tree-top kb)))
          (t (let* ((menu
                     (make-cascading-menu :pop-up possible-units))
                    (choice (choose-from-menu menu :position
                                   (make-position :x 800 :y 400))))
               (cond (choice
                      (show choice kb)
                      (put.value 'browser 'tree.top choice))
                     (t (let ((tree-top (first possible-units)))
                          (put.value 'browser 'tree.top tree-top)
                          (show tree-top kb)))))))))

(defun get-top-level-units (kb)
"Returns a list of the top level units in the kb.  Note that this function
excludes activevalue and activeimage units and any other units that are members
of a class other than classes or entities from another kb."
  (let ((classes (unit.children 'classes 'member))
        (result ()))
    (dolist (class classes result)
      (when (and (eq (kb.name (unit.kb class)) kb)        ;In correct kb
                 (= (length (unit.parents class 'member)) 1);Classes only
                 (null                                ;member parent
                  (remove 'classes                    ;Entities & classes only
                          (remove 'entities           ;class parent
                                  (mapcar #'unit.name
                                          (unit.parents class 'subclass))))))
        (push class result)))))

;;; BROWSER CONTROL

(defparameter outw
  (create-kee-output-window :region (make-region :left 550 :bottom 220
                                                 :width 500 :height 500)
                            :title "BROWSER"
                            :border 5
                            :activate-p nil)
  "Defines the output display window for the browser.")
```

```
(defun browse-entire-system ()
"Initializes the slots in the unit browser. Opens the browser window. Then
activates and exposes the output window for the browser and graphs the tree
including the unit workbench and its immediate children from the kb veg."
  (remove.all.values 'browser 'browser.menu)
  (put.value 'browser 'current.kb 'veg)
  (put.value 'browser 'top.level.units (get-top-level-units 'veg))
  (put.value 'browser 'tree.top 'workbench)
  (unitmsg 'viewport-browser.1 'open-panel!)
  (activate outw)
  (expose outw)
  (show 'workbench 'veg))
```

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

An Expert System Shell for Inferring Vegetation Characteristics - The Learning System (Tasks C and D)

5. Report Date

October 1992

6. Performing Organization Code

7. Author(s)

P. Ann Harrison
Patrick R. Harrison

8. Performing Organization Report No.

B921014-U-2R03

10. Work Unit No.

462-61-14

9. Performing Organization Name and Address

JJM Systems, Inc.
One Ivybrook Blvd., Suite 190
Ivyland, PA 18974

11. Contract or Grant No.

NAS5-30127

13. Type of Report and Period Covered

Task Report for Tasks C & D
July - October 1992

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546-0001
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

14. Sponsoring Agency Code

15. Supplementary Notes

The Lisp and KEE code for this work is available on a Sun Cartridge Tape.

16. Abstract

This report describes the implementation of a learning system that uses a data base of historical cover type reflectance data taken at different solar zenith angles and wavelengths to learn class descriptions of classes of cover types. It has been integrated with the VEG system and requires that the VEG system be loaded to operate. VEG is the NASA VEGetation workbench - an expert system for inferring vegetation characteristics from reflectance data. The learning system provides three basic options. Using option one, the system learns class descriptions of one or more classes. Using option two, the system learns class descriptions of one or more classes and then uses the learned classes to classify an unknown sample. Using option three, the user can test the system's classification performance. The learning system can also be run in an automatic mode. In this mode, options two and three are executed on each sample from an input file. The system was developed using KEE. It is menu driven and contains a sophisticated window and mouse driven interface which guides the user through various computations. Input and output file management and data formatting facilities are also provided.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
EXPERT SYSTEM, ARTIFICIAL INTELLIGENCE, REMOTE SENSING, LEARNING, DISCRIMINATION	UNCLASSIFIED - UNLIMITED

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
UNCLASSIFIED	UNCLASSIFIED		

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171

B921020-U-2R00

JJM SYSTEMS INC

APPENDIX D

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - IMPLEMENTATION OF ADDITIONAL TECHNIQUES (TASK E)

JJM SYSTEMS INC

B921019-U-2R04

AN EXPERT SYSTEM SHELL FOR INFERRING VEGETATION CHARACTERISTICS - IMPLEMENTATION OF ADDITIONAL TECHNIQUES (TASK E)

28 October 1992

Prepared for:

National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt, MD 20771

Prepared by:

JJM Systems, Inc.
One Ivybrook Boulevard, Suite 190
Ivyland, PA 18974

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF ACRONYMS

KEE	Knowledge Engineering Environment
VEG	VEGetation Workbench

SECTION 1.0

INTRODUCTION

The NASA VEGetation Workbench (VEG) infers vegetation characteristics from reflectance data. For a detailed description of VEG, see references 1 and 2. A number of subgoals are available in VEG. In the previous version of VEG, the subgoals DESCRIPTION.OF. SYSTEM, SPECTRAL.HEMISPHERICAL.REFLECTANCE, TOTAL.AND.SPECTRAL. HEMISPHERICAL.REFLECTANCE, and VIEW.ANGLE.EXTENSION were fully implemented. The subgoal LEARN.CLASS.DESCRIPTIONS was implemented in tasks C and D of the current contract. The basic framework for the subgoal PROPORTION.GROUND.COVER was developed as part of the previous version of VEG but no techniques were available. Several techniques to estimate the proportion ground cover were included in VEG. These were all dummy functions that returned the value one irrespective of the values of the arguments to the functions.

Three techniques that infer the proportion ground cover of a sample using data at a single wavelength have been fully implemented in VEG. In addition, the framework for the VEG subgoal PROPORTION.GROUND.COVER has been extended so that techniques can be applied to the data for a sample at multiple wavelengths. Two such techniques have also been implemented.

VEG can be operated in two different modes. In the "Research Mode," the scientist must separately execute each step in the processing of unknown cover type data. This mode allows the scientist to study the intermediate results in detail. VEG can also be operated in the "Automatic Mode." In this mode, the scientist selects the operations to be carried out. The cover type data is read from a file, processed and the results are written to another file without any further intervention from the user. The subgoal PROPORTION.GROUND.COVER was originally implemented in the VEG "Research Mode." This subgoal has now been implemented in the VEG "Automatic Mode."

Task E has been completed. Additional techniques to infer proportion ground cover have been implemented. The VEG subgoal PROPORTION.GROUND.COVER including the additional techniques is described in detail in this report. The code for the Lisp methods involved is included in Appendix A. A Sun cartridge tape containing these Lisp methods and the current version of VEG including the completed subgoal PROPORTION.GROUND.COVER has been delivered to the NASA GSFC technical representative.

JJM SYSTEMS INC

SECTION 2.0

DESCRIPTION OF THE SUBGOAL PROPORTION.GROUND.COVER IN THE VEG "RESEARCH MODE"

The VEG subgoal that estimates the proportion ground cover is selected by left clicking on PROPORTION.GROUND.COVER in the VEG "Research Mode" top level menu. When this option is selected, the menu shown in Figure 2-1 is displayed. This menu enables the user to invoke the steps involved in processing unknown cover type data to estimate the proportion ground cover and estimate the error in the calculation. Before each step is carried out, a check is made to make sure that the necessary prerequisite steps have been carried out. For example, the results cannot be output before the techniques have been executed. If any prerequisite steps have not been carried out, a message is displayed and the user is prompted to complete the necessary prerequisite steps.

New rules and new methods for the techniques that estimate the proportion ground cover have been developed. However, many of the methods and screens that were originally developed for the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE, have been re-used for the subgoal PROPORTION.GROUND.COVER. The steps involved in estimating the proportion ground cover in the VEG "Research Mode" are briefly described in this section.

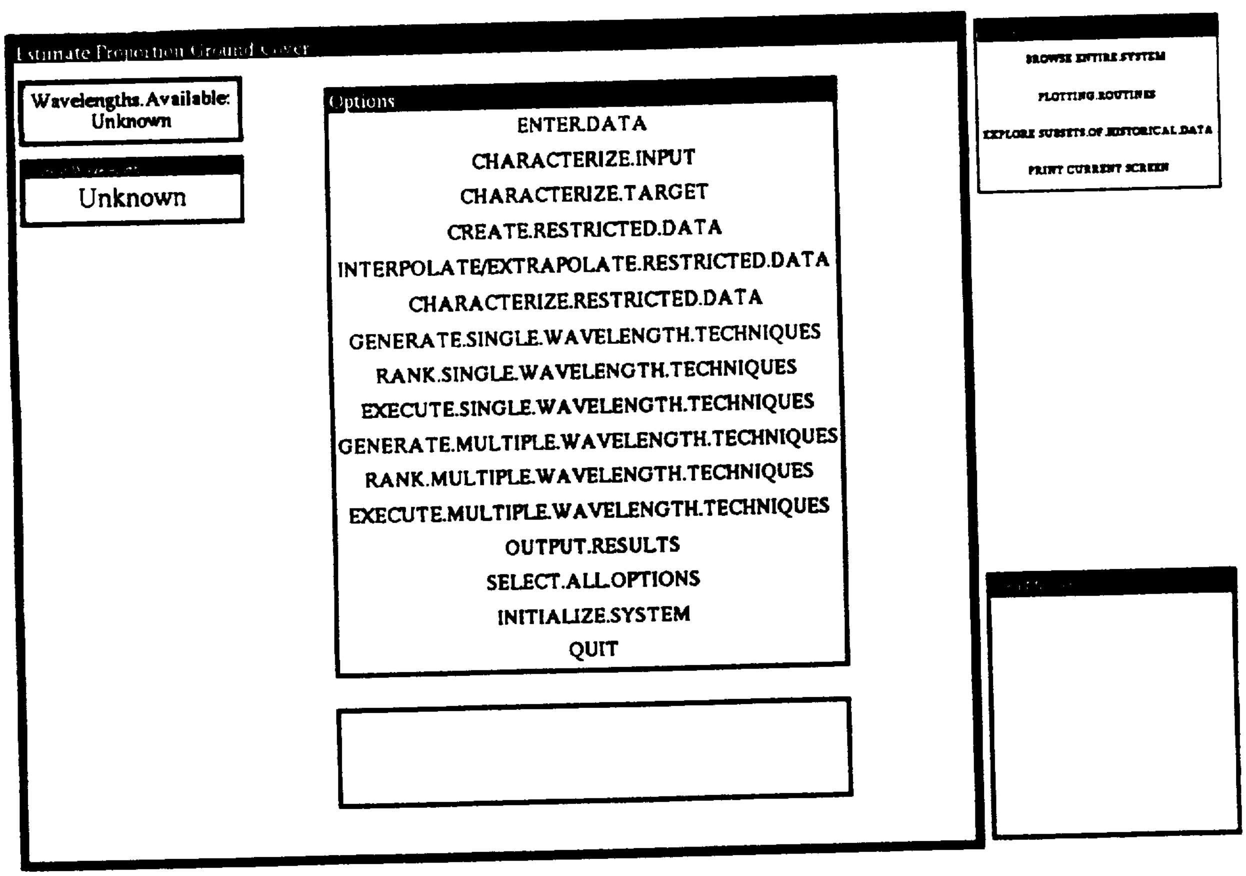


Figure 2-1

Menu for the VEG Subgoal PROPORTION.GROUND.COVER

2.1 ENTER DATA

The code and interface that were originally developed for the step ENTER.DATA for the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE have been modified for re-use in this step. When the user selects the step ENTER.DATA, an interface opens. This interface allows the user to either enter a new original set of data for an unknown cover type or select one of a number of samples of unknown cover type data already stored in VEG. If the user chooses to enter original data, another interface opens as shown in Figure 2-2. This interface allows the user to enter data for the new sample. Each value is checked as soon as it has been entered to make sure that it is of the correct type and is in the valid range for the data item it represents. The user can left click on the menu button "SAVE.DATA," at the bottom of the screen in Figure 2-2 to store the data. Before a set of cover type data is stored, the system checks that at least the solar zenith angle, wavelength and reflectance data have been entered. If any of these items is missing, the user is prompted to supply the missing items before storing the data. If the user elects to use one of the sets of sample cover type data already stored in VEG, a different screen is opened. This screen displays the name and a brief description of each available sample. The user can left click on the name of the sample to select it. Each set of unknown cover type data whether entered by the user or selected from the samples already in VEG can contain reflectance data at one or more wavelengths.

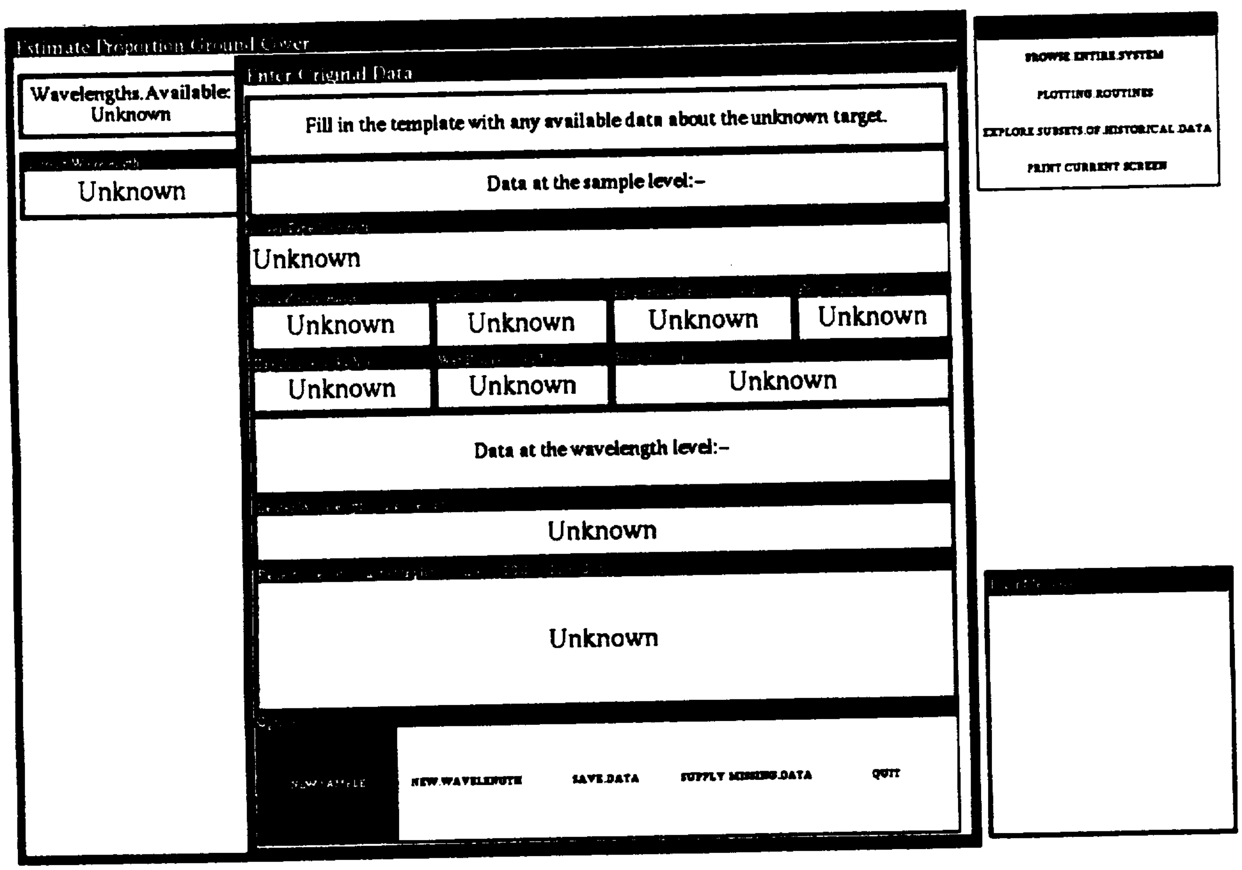


Figure 2-2

The Screen for Entering Original Cover Type Data

2.2 CHARACTERIZE INPUT

The unknown cover type data at each wavelength is characterized using code that was developed for the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE. Sets of view angles in the same azimuthal plane are identified as "strings." Strings are characterized as full-strings if they contain both forwardscatter and backscatter data and half-strings if they contain either backscatter or forwardscatter data.

2.3 CHARACTERIZE TARGET

If the sample data does not contain a value for ground cover or leaf area index, a crude estimation of these values is made. The code developed for the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE is re-used for this purpose.

2.4 CREATE RESTRICTED DATA

VEG contains a data base of historical cover type data. This data base is organized as a hierarchy of units that are subclasses and members of the unit HISTORICAL.COVER.TYPES. These units contain the results of previous experiments including values for the spectral hemispherical reflectance and the proportion ground cover. They are used in VEG to generate the coefficients needed in many of the technique methods. In several of the VEG subgoals, an estimate of the error term involved in applying a technique to the unknown cover type data is calculated. This is accomplished by applying the same technique to samples in the data base of historical cover types and measuring the error involved in each calculation. The error terms obtained by applying the technique to a number of historical data base samples are collected together and the root mean square value of the error terms is calculated. This value gives a measure of the error involved in applying the technique to the unknown cover type data.

In the step "CREATE.RESTRICTED.DATA," the data base of historical cover types is searched to find the sets of historical data that match the unknown cover type. This set of cover types is referred to as the "restricted data set." The selection of the restricted data set can either be made automatically by the system, or it can be made by the user. In the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE, the reflectance data at each wavelength is considered separately when searching the data base of historical cover types. However, some of the techniques for estimating proportion ground cover are multiple wavelength techniques. These techniques involve performing calculations on data in the red and near-infrared bands and then combining the results together to calculate the result. These techniques require a sample to have data in both bands. If these techniques are to be applied to historical cover type data, the data selected from the data base must also have data at multiple wavelengths. Thus the restricted data set is selected by considering the data at all the wavelengths together. The code originally developed for the VEG subgoal TOTAL.HEMISPHERICAL.REFLECTANCE is re-used for this purpose.

If the user elects to have the restricted data set selected automatically by the system, the data base of historical cover types is searched to find the cover types that best match the unknown cover type sample. The subset of historical cover types that have data that match all the wavelengths of the unknown cover type sample is first identified. From this subset, the cover types whose ground cover and solar zenith angle are within ten percent of the values for the unknown sample are then identified and pushed onto a list. If the list contains insufficient values, the search is then widened to include cover types whose sun angles and proportion ground cover are within 20 percent of the values for unknown cover type sample. The search criteria are progressively widened until either sufficient cover types have been identified or all cover types whose sun angle and proportion ground cover are within 100 percent of the values in the unknown cover type sample have been collected. In the search for the best matching cover types, the value for the proportion ground cover calculated in the step CHARACTERIZE.TARGET is used as the proportion ground cover of the unknown cover type sample.

The user can also manually select the restricted data set. In this case, the screen shown in Figure 2-3 is opened. This screen allows the user to enter the maximum and minimum values to be considered for parameters such as height and solar zenith angle. When the user left clicks on "MATCH.DATA," the data base of historical cover types is searched to find the cover types that match the criteria entered by the user and that also contain data at all the wavelengths present in the unknown cover type sample. The user can then select the matched cover types, enter new maximum and minimum values and match the data again or select a subset of the matched data using the screen shown in Figure 2-4.

When the search for the restricted data set has been completed, either manually or automatically, the cover type names are stored. The names of the historical cover types at the sun angle level (e.g. CT11-58) are stored in the slot R.D.S of the unknown cover type unit (e.g. the unit SAMPLE6). The names of the corresponding cover type units at the wavelength level such as CT11-58-1 and CT11-58-2 are stored in the slot R.D.S of the unknown cover types units at the wavelength level, for example W11 and W12.

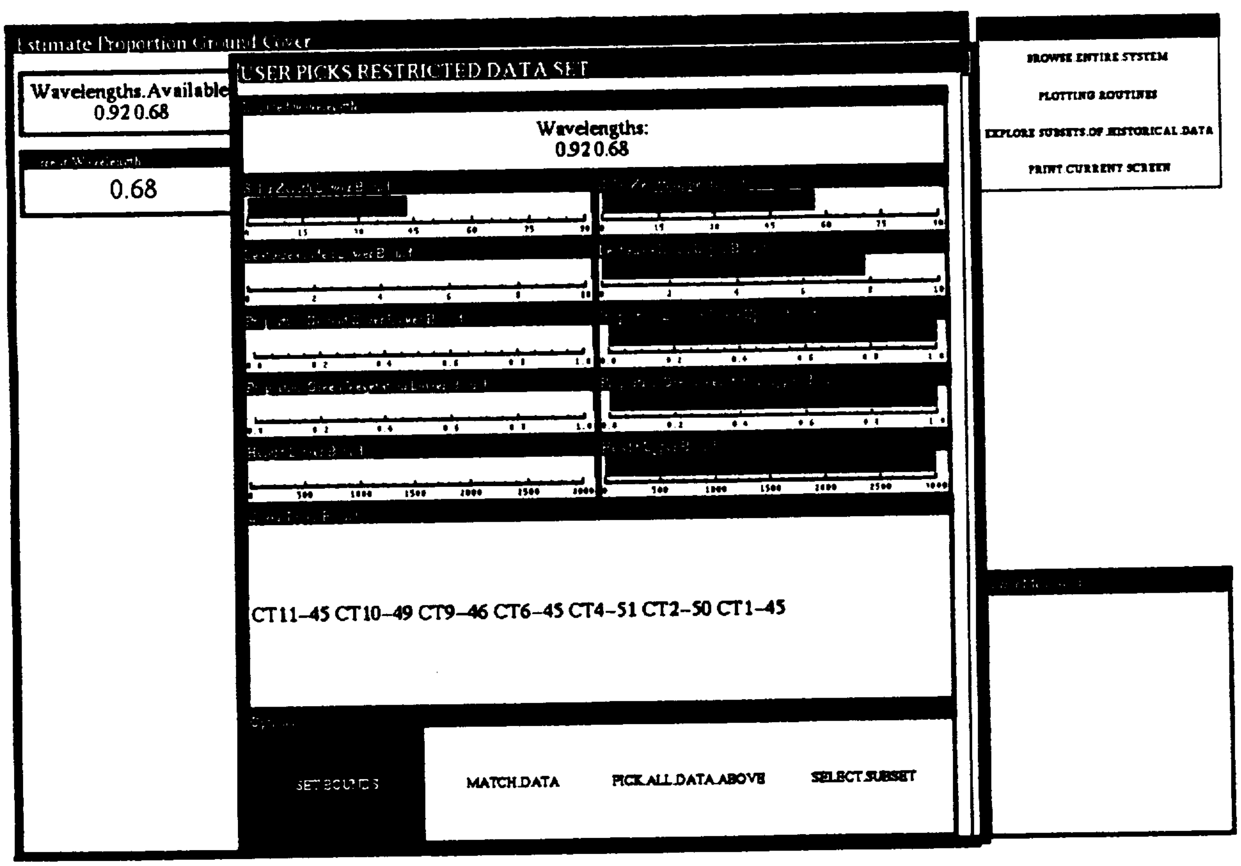


Figure 2-3

The Screen for Picking the Restricted Data Set

JJM SYSTEMS INC

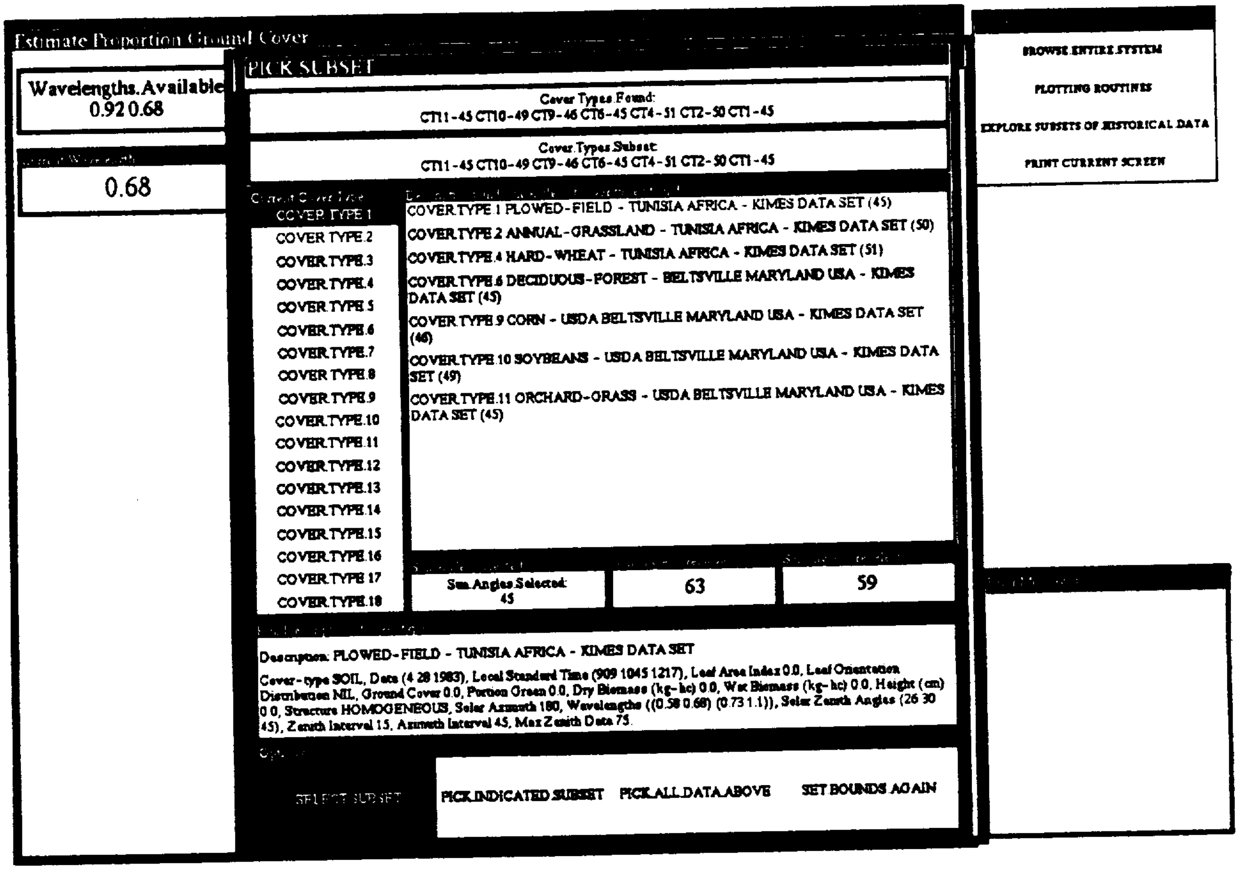


Figure 2-4

The Screen for Picking the Subset of the Matched Data

2.5 INTERPOLATE/EXTRAPOLATE RESTRICTED DATA

The code for this step in the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE is re-used for the subgoal PROPORTION.GROUND.COVER. In this step, the raw reflectance data for each cover type in the restricted data set is interpolated and extrapolated so that the view angles exactly match at each wavelength the view angles in the unknown cover type data. A unit such as RHD.SAMPLE6 is created as a subclass of the unit RESTRICTED.HISTORICAL.DATA. A hierarchy of units is set up as subclasses and members of the unit RHD.SAMPLE6 to store the data corresponding to SAMPLE6. A subclass unit such as RHD.SAMPLE6-CT11-58 is set up to correspond to each restricted data set cover type at the sun angle level. Member units of this unit such as R-4 and R-15 are created to store the reflectance data interpolated and extrapolated from the units CT11-58-1 and CT11-58-2. The class-member relationship between the units such as RHD.SAMPLE6-CT11-58, R-4 and R-15 is the same as the class member relationship between the units SAMPLE6, W11 and W12. This organization of units allows the same methods to be applied to the restricted historical data units and to the sample units.

2.6 CHARACTERIZE RESTRICTED DATA

The data in the restricted historical data units are characterized in the same way as the sample of unknown cover type data was characterized in the step CHARACTERIZE.INPUT. The code from the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE is re-used for this step.

2.7 GENERATE SINGLE WAVELENGTH TECHNIQUES

The single wavelength techniques can be generated automatically by the system or selected by the user. The code for generating techniques for the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE was copied and modified for this step towards the goal PROPORTION.GROUND.COVER. A new screen was created to allow the user to select the single wavelength proportion ground cover techniques, but many existing functions were used to operate this screen.

If the user elects to have the system generate the techniques, the rules in the rulebase PROPORTION.GROUND.COVER.SINGLE.WAVELENGTH.RULES are run. The rules operate on the unknown sample data at the wavelength level. All the techniques that are suitable for estimating the proportion ground cover of a sample at a particular wavelength are stored in the TECHNIQUES slot of the wavelength level unit. The rules are listed in Figure 2-5. The rule PGCSWR.10 selects the technique PGC.NEAR.NADIR if the reflectance data contains at least one view angle. The rule PGCSWR.11 fires and the technique PGC.NORMAN.PLUS is selected if the reflectance data contains three or more view angles. Rules PGCSWR.12, PGCSWR.13 and PGCSWR.14 fire if the reflectance data contains 2, 3 or 4 view angles respectively. These rules select the appropriate PGC.2.OFF.NADIR.ANGLE techniques, according to the number of view angles available.

If the user elects to choose the techniques manually, the screen shown in Figure 2-6 is opened. When the user left clicks on the name of a technique, a brief description of the technique is displayed in the box labelled "Description of Technique." A function is called to check whether the technique is suitable for the sample. For example, if the technique PGC.NORMAN.PLUS is selected, the function pgc.norman.plus.ok is called. This function returns T if the sample has at least 3 view angles and NIL otherwise. It the technique is suitable for the sample, the message "Technique is suitable for this sample" is displayed in the box labelled "Error Message," and the technique is selected. Otherwise an error message is displayed in the same box and the technique is not selected. When the user left clicks on PICK.SELECTED.TECHNIQUES, the selected techniques are stored in the TECHNIQUES slot of the wavelength level unknown cover type unit.

The NASA GSFC technical representative specified the single wavelength techniques that were to be implemented for estimating proportion ground cover. The maximum possible value for proportion ground cover is 1.0. It is possible for a technique to return a value for proportion ground cover that is greater than 1.0. In these cases, the value of the proportion ground cover was reported as 1.0.

The technique PGC.NEAR.NADIR uses the nearest view angle to the nadir and applies a simple linear regression (least squares) technique to calculate the proportion ground cover. This technique is similar to the technique NADIR which is used for estimating spectral hemispherical reflectance. It can be applied to any sample that has at least one view angle.

```
(IF (THE CURRENT.SAMPLE.WAVELENGTHS
          OF ESTIMATE.HEMISPHERICAL.REFLECTANCE IS ?X)
     (LISP (CONSP (GET.VALUE ?X 'REFLECTANCE.DATA)))
     THEN
     (LISP (ADD.VALUES ?X
                         'TECHNIQUES
                         '(PGC.NEAR.NADIR))))
RULE : PGCSWR.10

(IF (THE CURRENT.SAMPLE.WAVELENGTHS
          OF ESTIMATE.HEMISPHERICAL.REFLECTANCE IS ?X)
     (THE NUMBER.VIEW.ANGLES OF ?X IS ?Y)
     (LISP (AND (NUMBERP ?Y) (>= ?Y 3)))
     THEN
     (LISP (ADD.VALUES ?X
                         'TECHNIQUES
                         '(PGC.NORMAN.PLUS))))
RULE : PGCSWR.11

(IF (THE CURRENT.SAMPLE.WAVELENGTHS
          OF ESTIMATE.HEMISPHERICAL.REFLECTANCE IS ?X)
     (THE NUMBER.VIEW.ANGLES OF ?X IS 2)
     THEN
     (LISP (ADD.VALUES ?X
                         'TECHNIQUES
                         '(PGC.2OFF.NADIR.ANGLE.0))))
RULE : PGCSWR.12

(IF (THE CURRENT.SAMPLE.WAVELENGTHS
          OF ESTIMATE.HEMISPHERICAL.REFLECTANCE IS ?X)
     (THE NUMBER.VIEW.ANGLES OF ?X IS 3)
     THEN
     (LISP
     (ADD.VALUES ?X
                   'TECHNIQUES
                   (PGC.2OFF.NADIR.ANGLE.0 PGC.2OFF.NADIR.ANGLE.1
                    PGC.2OFF.NADIR.ANGLE.2))))
RULE : PGCSWR.13

(IF (THE CURRENT.SAMPLE.WAVELENGTHS
          OF ESTIMATE.HEMISPHERICAL.REFLECTANCE IS ?X)
     (THE NUMBER.VIEW.ANGLES OF ?X IS 4)
     THEN
     (LISP
     (ADD.VALUES ?X
                   'TECHNIQUES
                   '(PGC.2OFF.NADIR.ANGLE.0 PGC.2OFF.NADIR.ANGLE.1
                     PGC.2OFF.NADIR.ANGLE.2 PGC.2OFF.NADIR.ANGLE.3
                     PGC.2OFF.NADIR.ANGLE.4 PGC.2OFF.NADIR.ANGLE.5))))
RULE : PGCSWR.14
```

B921019G1

Figure 2-5

The Proportion Ground Cover Single Wavelength Rules

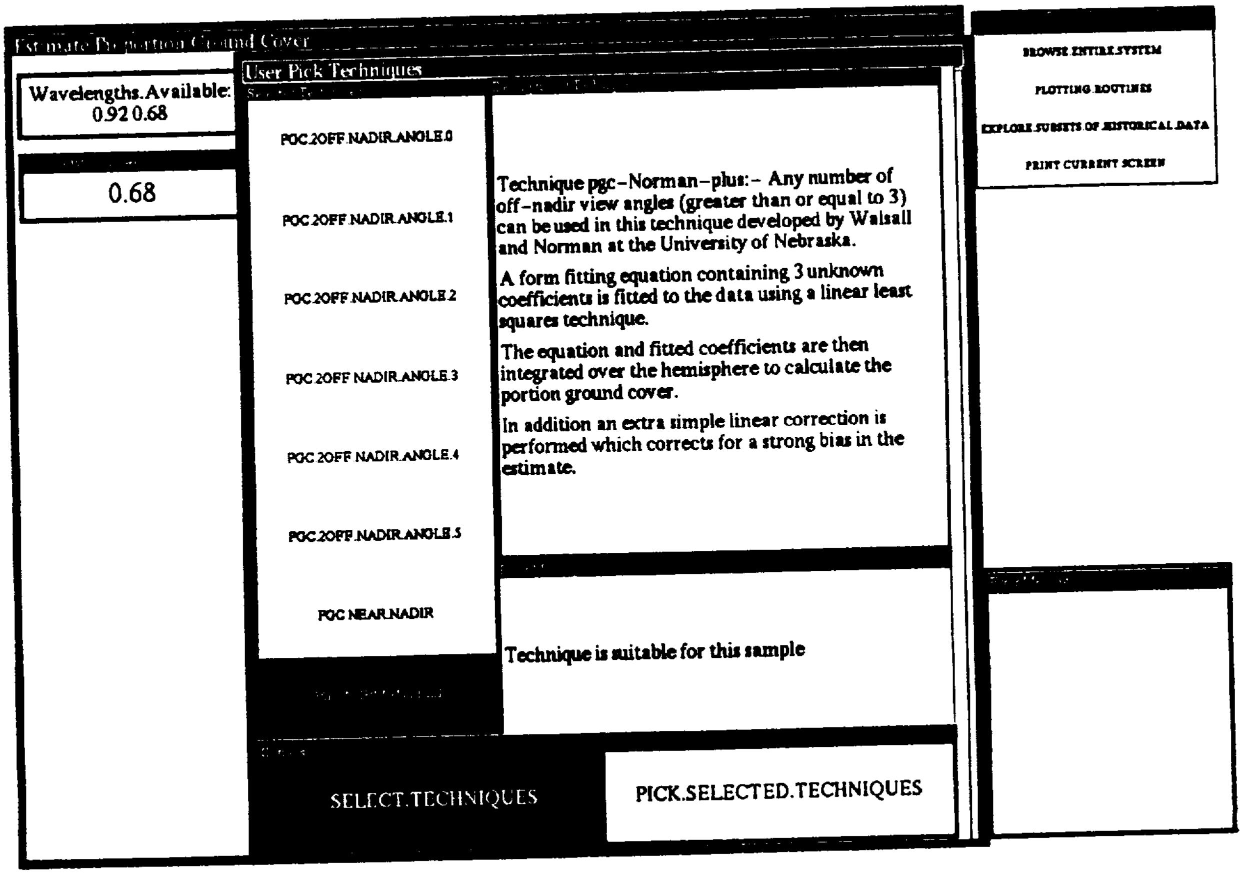


Figure 2-6

The Screen for Selecting the Single Wavelength Proportion Ground Cover Techniques

The technique PGC.2.OFF.NADIR.ANGLE can be applied to any two view angles in a set of reflectance data. A multiple regression (least squares) technique is applied to determine the proportion ground cover. This technique is similar to the technique 2.OFF.NADIR angle that was used to estimate the spectral hemispherical reflectance. The NASA GSFC technical representative advised that the technique should be applied to all samples with two, three or four view angles and it should be applied to every possible pair of angles in each suitable sample.

The technique PGC.NORMAN.PLUS is applied to samples with three or more view angles. The NORMAN technique is applied to the reflectance data to estimate the hemispherical reflectance. Then a simple linear regression is applied to estimate the proportion ground cover. This technique is similar to the technique NORMAN.PLUS for estimating spectral hemispherical reflectance.

2.8 RANK SINGLE WAVELENGTH TECHNIQUES

The code from this step in the subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE was re-used for this step. The techniques are ranked according to a simple weighting scheme and the ranked techniques at each wavelength are displayed on the screen. The user can select the best one, two or three techniques for each wavelength, pick all the selected techniques or repeat the previous step and generate the techniques again.

2.9 EXECUTE SINGLE WAVELENGTH TECHNIQUES

The code providing the framework for this step from the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE was re-used for this step. However, the methods for generating the coefficients and calculating the proportion ground cover for each technique are new. The code for these methods is included in Appendix A. When the step EXECUTE.SINGLE.WAVELENGTH.TECHNIQUES is selected, the techniques are applied to the data at each wavelength in the unknown cover type sample. If a technique requires coefficients, the user is asked whether all or half the restricted data set should be used for generating the coefficients and estimating the error. The appropriate coefficient methods are applied as necessary. The techniques are applied to the restricted historical data and the difference between the calculated proportion ground cover and the correct value for the proportion ground cover stored in the data base is calculated. Using the error measurements from several historical cover types, the root mean square error is calculated. This provides an estimate of the error involved in applying the technique to the sample of unknown cover type data. A hierarchy of units is set up to hold the results for each cover type in the restricted historical data and to hold the calculated proportion ground cover, error estimate and coefficients for each technique.

2.10 GENERATE MULTIPLE WAVELENGTH TECHNIQUES

The subgoal PROPORTION.GROUND.COVER includes techniques which are applied to a sample at multiple wavelengths. No other VEG subgoal has techniques of this type. The subgoal TOTAL.HEMISPHERICAL.REFLECTANCE has a technique which combines the spectral hemispherical reflectance results for several wavelengths to determine the total hemispherical reflectance. However, the multiple wavelength techniques for calculating the proportion ground cover operate on the raw reflectance data for the sample in the red (wavelength 0.63 μm - 0.68 μm) and near-infrared (wavelength 0.76 μm - 1.1 μm) bands. Because no techniques of this type had been included in the previous version of VEG, it was necessary to build the framework for these techniques. This included building the interfaces for generating, ranking and executing multiple wavelength techniques. The design of these interfaces was based on the interfaces that had previously been developed for the single wavelength techniques. Some of the existing code for the single wavelength techniques was copied and modified for use with the multiple wavelength techniques.

When the user selects the step GENERATE.MULTIPLE.WAVELENGTH.TECHNIQUES from the PROPORTION.GROUND.COVER main menu, the screen shown in Figure 2-7 is displayed. This screen enables the user to choose the method of selecting the techniques. The techniques can either be selected automatically by the system or selected manually by the user.

If the user elects to have the techniques selected by the system, the rules in the rulebase PROPORTION.GROUND.COVER.MULTIPLE.WAVELENGTH.RULES are run. These rules are shown in Figure 2-8. The rule PGCMWR1 fires if the unknown cover type has reflectance data in both the red and near-infrared bands. If this rule fires, the techniques PGC.NEAR.NADIR.ND is added to the slot MULTIPLE.WAVELENGTH.TECHNIQUES of the unit PROPORTION. GROUND.COVER. If the unknown cover type sample has reflectance data with at least three view angles in the red and near-infrared bands, the rule PGCMWR2 fires and the technique PGC.NORMAN.PLUS.ND is added to the slot MULTIPLE.WAVELENGTH.TECHNIQUES of the unit PROPORTION.GROUND.COVER. When the rules have run, the selected techniques are displayed on the screen as shown in Figure 2-7.

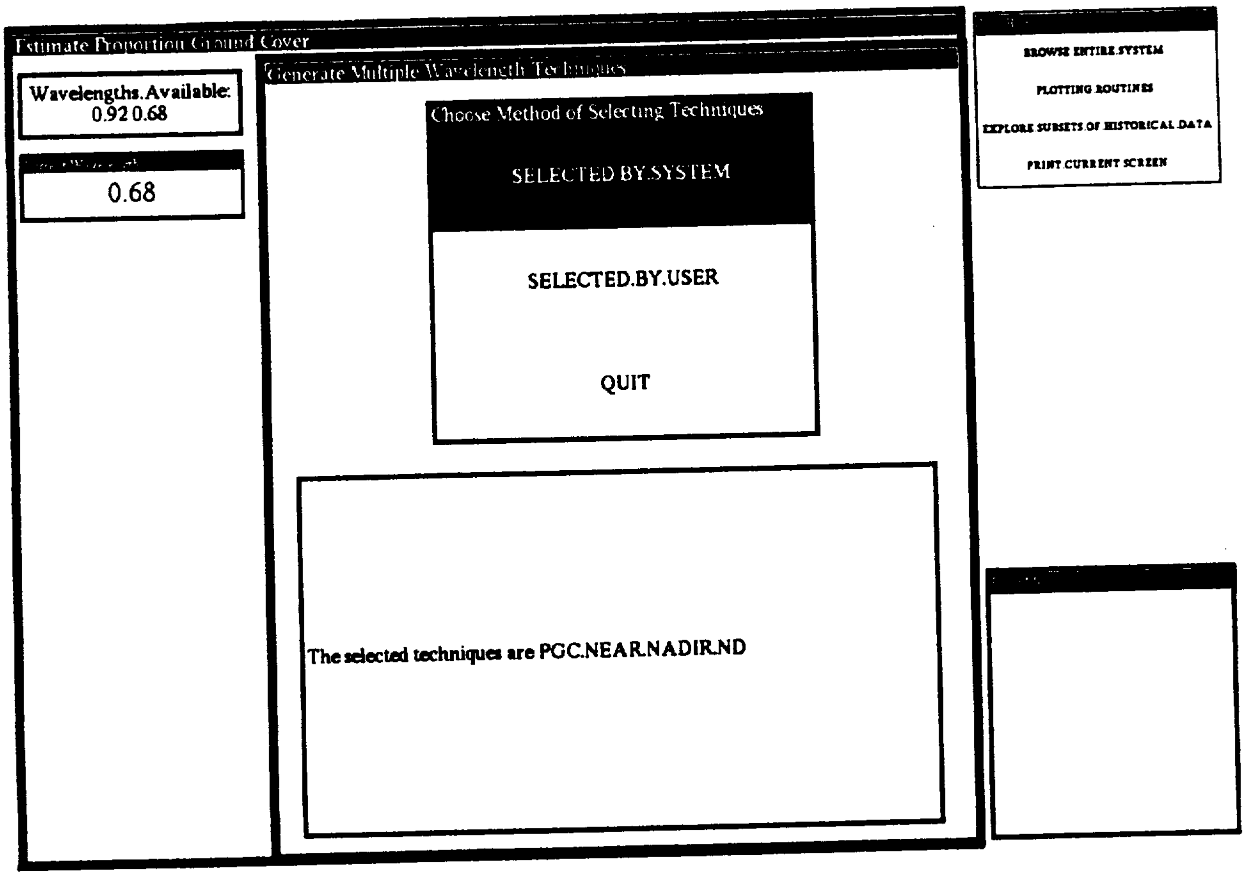


Figure 2-7

Choosing the Method of Selecting the Multiple Wavelength Techniques

The technique PGC.NEAR.NADIR.ND uses the reflectance value at the nearest view angle to the nadir in the red and near-infrared bands to calculate the normalized difference. A simple linear regression is then applied to calculate the proportion ground cover.

The technique PGC.NORMAN.PLUS.ND applies the Norman technique to three or more view angles in the red and near-infrared bands to estimate the spectral hemispherical reflectance in each band. The normalized difference is then calculated using these estimates. Finally, a simple linear regression is applied to calculate the proportion ground cover.

```
          (IF
            (THE CURRENT.SAMPLE.WAVELENGTHS
                  OF ESTIMA  TE.HEMISPHERICAL.REFLECTANCE IS ?X)
            (THE WAVELENGTH OF ?X IS ?RED)
            (LISP (AND (>= ?RED 0.63)
                       (<= ?RED 0.69)))
            (LISP (CONSP  (GET.VALUE ?X 'REFLECTANCE.DATA)))
            (THE CURRENT.SAMPLE.WAVELENGTHS
                  OF ESTIMATE.HEMISPHERICAL.REFLECTANCE IS ?Y)
            (THE WAVELENGTH OF ?Y IS ?NIR)
            (LISP (AND (>= ?NIR 0.76)
                       (<= ?NIR 1.1)))
            (LISP (CONSP  (GET.VALUE ?Y 'REFLECTANCE.DATA)))
            THEN
            (LISP
              (ADD.V ALUE  'PROPORTION.GROUND.COVER
                           'MUL TIPLE.WAVELENGTH.TECHNIQUES
                           'PGC.NEAR.NADIR.ND)))
RULE :  PGCMWR1


          (IF
            (THE CURRENT.SAMPLE.WAVELENGTHS
                  OF ESTIMA  TE.HEMISPHERICAL.REFLECTANCE IS ?X)
            (THE WAVELENGTH OF ?X IS ?RED)
            (LISP (AND (>= ?RED 0.63)
                       (<= ?RED 0.69)))
            (THE NUMBER.VIEW.ANGLES OF ?X IS ?RED-VIEW)
            (LISP (>= ?RED-VIEW 3))
            (THE CURRENT.SAMPLE.WAVELENGTHS
                  OF ESTIMA  TE.HEMISPHERICAL.REFLECTANCE IS ?Y)
            (THE WAVELENGTH OF ?Y IS ?NIR)
            (LISP (AND (>= ?NIR 0.76)
                       (<= ?NIR 1.1)))
            (THE NUMBER.VIEW.ANGLES OF ?Y IS ?NIR-VIEW)
            (LISP (>= ?NIR-VIEW 3))
            THEN
            (LISP
              (ADD.V ALUE  'PROPORTION.GROUND.COVER
                           'MUL TIPLE.WAVELENGTH.TECHNIQUES
                           'PGC.NORMAN.PLUS.ND)))
RULE :  PGCMWR2
```

B921019G2

Figure 2-8

The Proportion Ground Cover Multiple Wavelength Rules

If the user elects to choose the multiple wavelength techniques manually, the screen shown in Figure 2-9 is opened. Left clicking on the name of a technique causes a brief description of the technique to be displayed in the box labelled "Description of Technique." A function such as the function pgc.near.nadir.nd.ok is called to determine if the sample has all the data required by the technique. In the case of the technique PGC.NEAR.NORMAN.ND, the sample must have reflectance data at the red and near-infrared bands. If the technique is suitable for the sample, the words "Technique is suitable for this sample" are displayed in the box labelled "Error Message," and the technique name is highlighted to show that the technique has been selected. Otherwise, an error message is displayed and the technique is not selected. The user must left click on "PICK.TECHNIQUES" to finish picking the techniques and close the screen. The message "Finished selecting techniques" is then displayed in the lower box on the screen shown in Figure 2-7. Left clicking on "QUIT" returns the user to the main menu for the subgoal PROPORTION.GROUND.COVER.

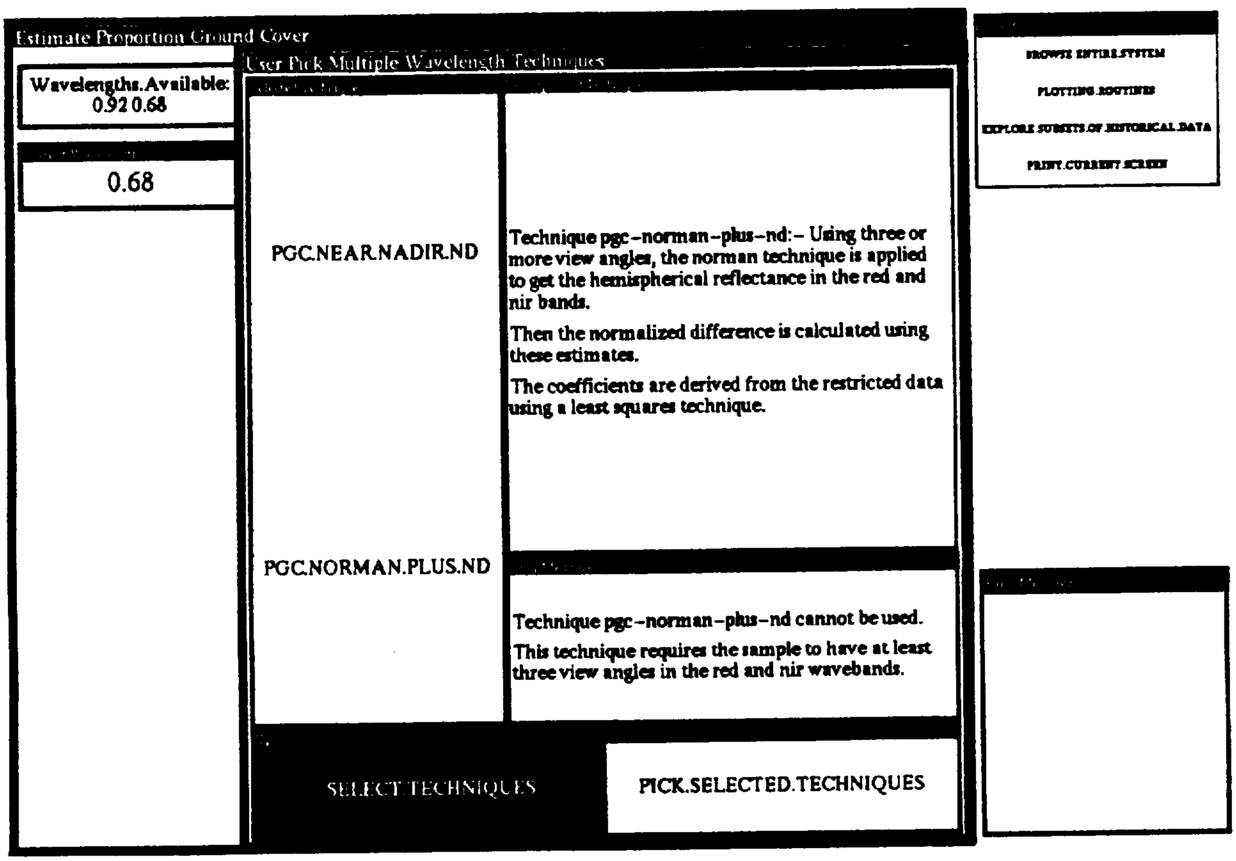


Figure 2-9

The Screen for Selecting the Multiple Wavelength Techniques

2.11 RANK MULTIPLE WAVELENGTH TECHNIQUES

The RANK.MULTIPLE.WAVELENGTH.TECHNIQUES interface has been implemented so that it is similar to the interface used for ranking the single wavelength techniques. When the user selects this step, the screen shown in Figure 2-10 is opened. The techniques that were generated in the previous step are ranked using a simple weighting scheme and then displayed on the screen. Even though VEG currently contains only two proportion ground cover multiple wavelength techniques, the interface has been implemented to allow for the addition of more multiple wavelength techniques at a later date. The user can choose the one, two, three best techniques, all the selected techniques or open the interface to generate the techniques again. If the user chooses the three best techniques and only two techniques have been generated, both techniques are used. When the user left clicks on a selection, the interface is closed and the main menu for the subgoal PROPORTION.GROUND.COVER is once again visible.

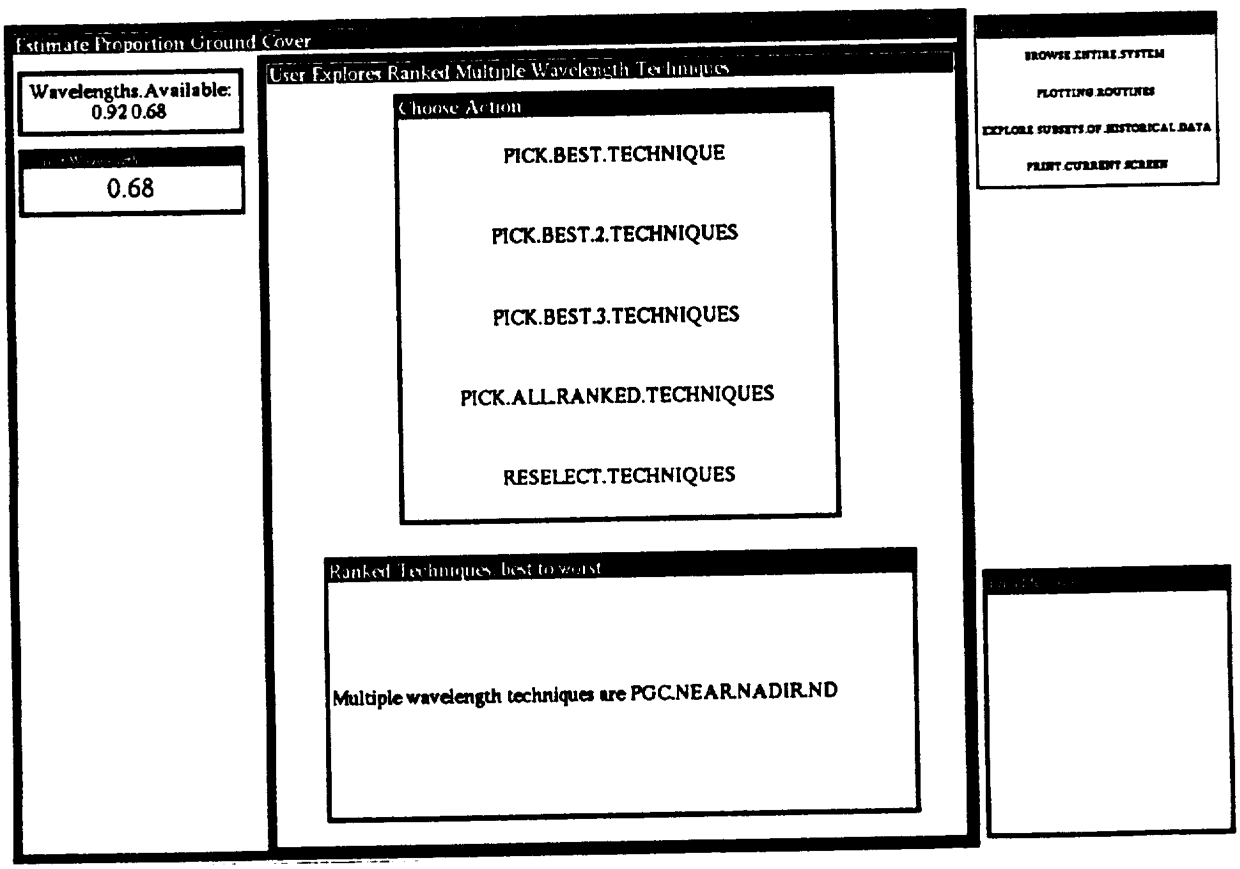


Figure 2-10

The RANK.MULTIPLE.WAVELENGTH.TECHNIQUES Interface

2.12 EXECUTE MULTIPLE WAVELENGTH TECHNIQUES

The Lisp code used for executing the single wavelength techniques and was modified to produce the code for executing the multiple wavelength techniques. When the user selects this step, the interface shown in Figure 2-11 is opened and the message "Executing multiple wavelength techniques - please wait........" is displayed. A check is made to make sure at least one multiple wavelength technique has been generated. If no techniques have been generated, an error message is displayed. Otherwise, the techniques are executed one at a time. If a technique requires coefficients, the user is asked whether all or half of the historical data units in the restricted data set should be used for calculating the coefficients and the error estimate. This step uses the screen shown in Figure 2-12. If the user left clicks on "HALF.OF. HISTORICAL.DATA," the list of restricted historical data units is divided into two lists. One list of units is then used in the function for calculating the coefficients for the technique. The other list is used for calculating the error estimate. The coefficients are calculated by applying the coefficient method for the technique to the appropriate list of restricted historical data units. Then the technique is applied to the unknown cover type data using the calculated coefficients. Next, the technique method is applied to each restricted historical data unit in the appropriate list. For each unit, the difference between the calculated value and the correct value for the ground cover (previously stored in the unit) is calculated. The root mean square of the error terms for all the appropriate restricted historical data units is calculated. This number provides an estimate of the error term involved in applying the technique to the sample.

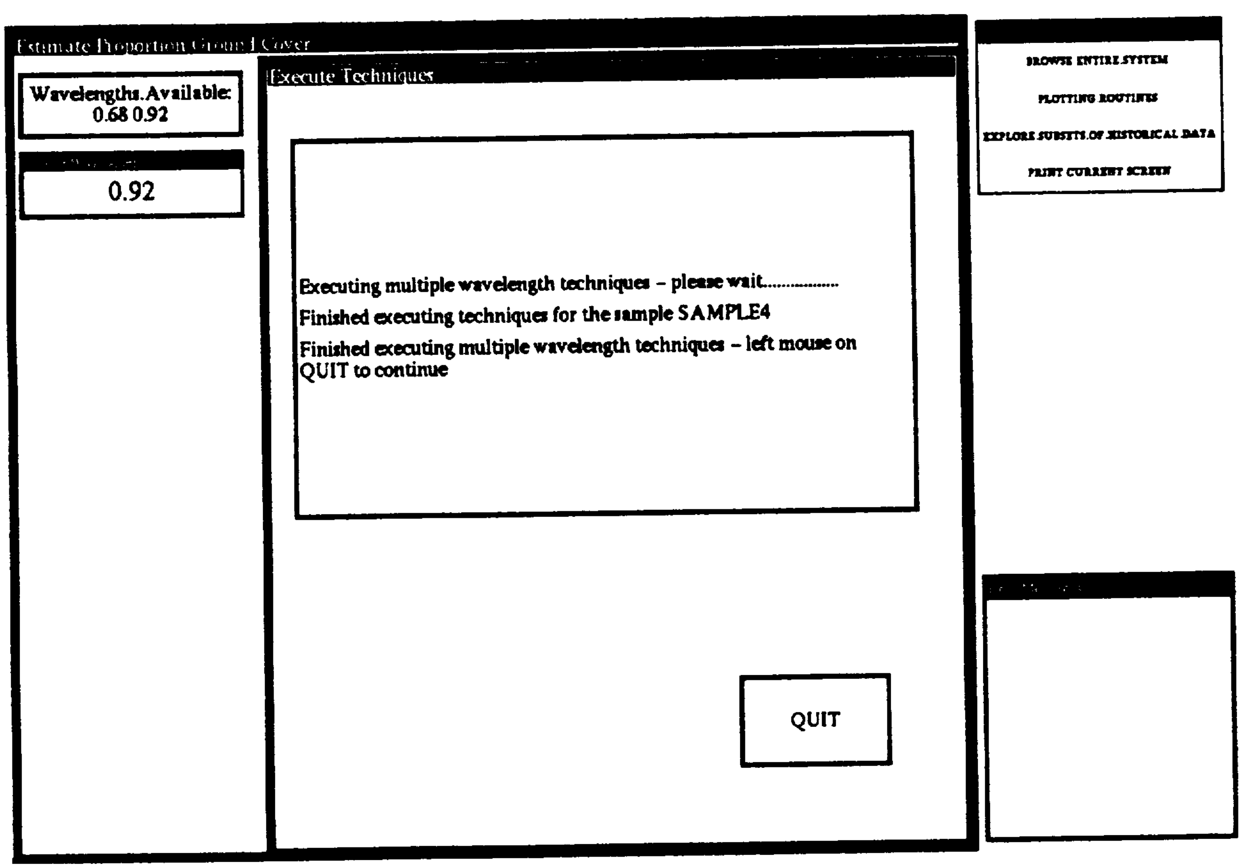


Figure 2-11

The EXECUTE.MULTIPLE.WAVELENGTH.TECHNIQUES Interface

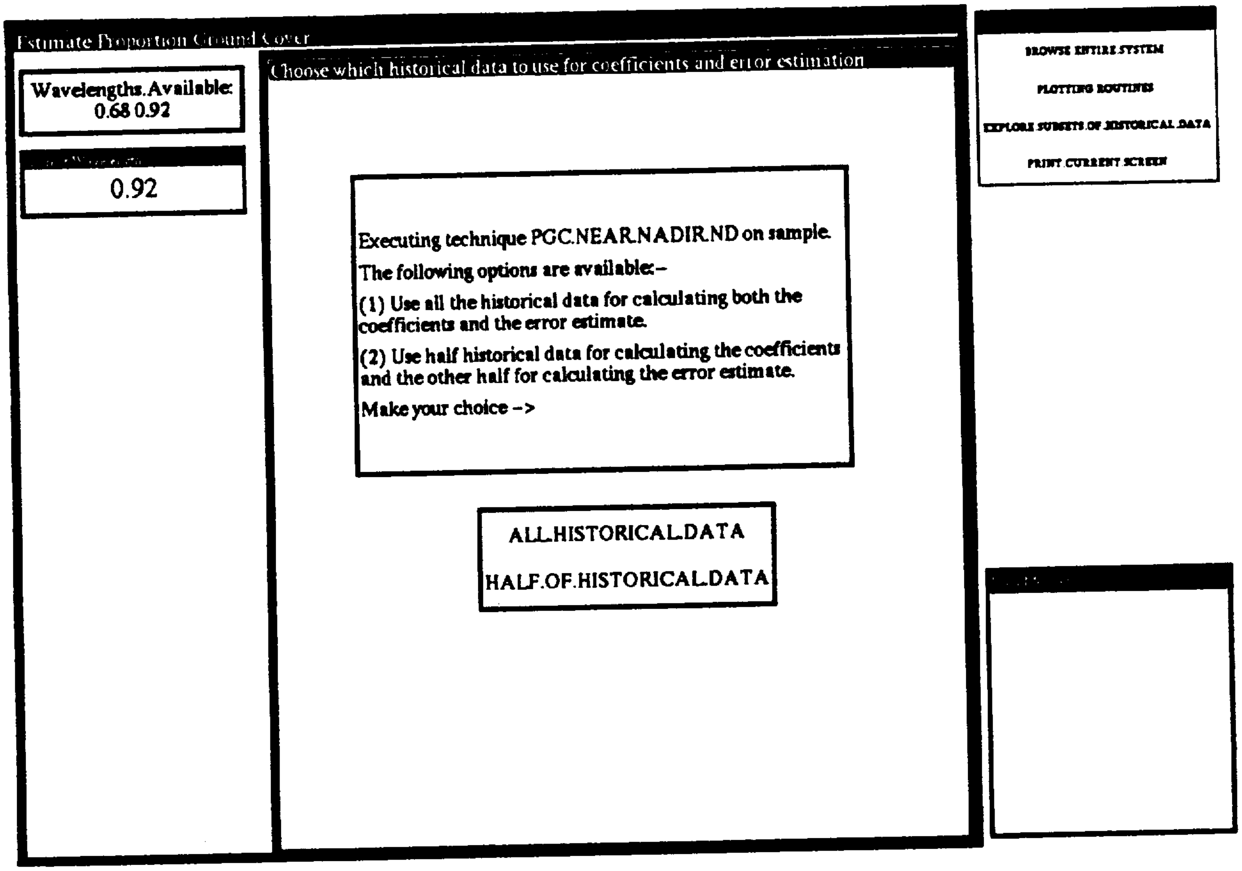


Figure 2-12

Choosing which Historical Data will be Used for Coefficients and Error Estimation

A hierarchy of units is created as subclasses and members of the unit RESULTS to hold the results of processing the multiple wavelength techniques. The results for each restricted historical data unit are stored as well as the results for the sample. This is to allow the scientist to study the intermediate results of processing the data in detail.

Each selected multiple wavelength technique is applied to the sample and the results are stored. When the execution of the multiple wavelength techniques has been completed, a message is displayed on the screen, and the user is prompted to left click on the box labelled "QUIT" to return to the main menu.

2.13 OUTPUT RESULTS

The results are displayed on the screen as shown in Figure 2-13. This screen was originally constructed for the VEG subgoal SPECTRAL.HEMISPHERICAL.REFLECTANCE. The title has been changed to "Proportion Ground Cover Results." The multiple wavelength results are displayed in the box labelled, "Sample Results." For each multiple wavelength technique, the estimate of the proportion ground cover, the error estimates and the coefficients are displayed. The multiple wavelength results apply to the entire sample, and they do not change when the user selects either the next or the previous wavelength. The single wavelength results are displayed in the box labelled, "Wavelength Results." For each single wavelength technique, the name of the technique is displayed together with the estimate of proportion ground cover, the error estimate and the coefficients. If the user left clicks on "NEXT.WAVELENGTH" or "PREVIOUS. WAVELENGTH," the wavelength level results for a different wavelength are displayed. When the user left clicks on "QUIT," another screen is opened. The user is asked whether the results should be written to a file. A detailed description of the interface for writing results from VEG was provided in the JJM Systems report B921016-U-2R02. The results for all the VEG subgoals, including the subgoal PROPORTION.GROUND.COVER, can be written to a file. Appendix B contains the listings of files that were produced when the subgoal PROPORTION.GROUND.COVER was tested.

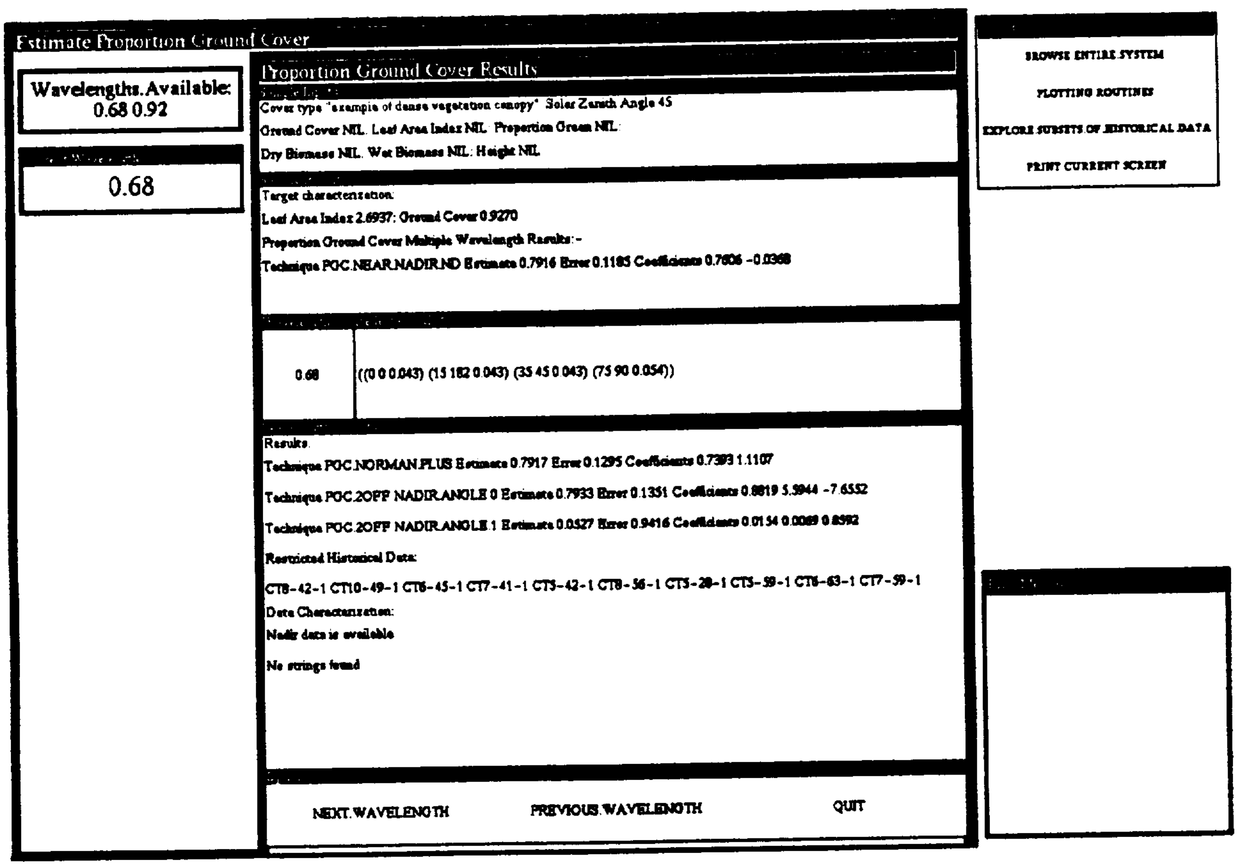


Figure 2-13

Displaying the Results of the VEG Subgoal PROPORTION.GROUND.COVER

SECTION 3.0

DESCRIPTION OF THE SUBGOAL PROPORTION.GROUND.COVER IN THE VEG "AUTOMATIC MODE"

When the user left clicks on the option PROPORTION.GROUND.COVER on the VEG "Automatic Mode" top level screen, additional boxes are opened, as shown in Figure 3-1. This screen enables the user to enter the name of the input file and specify the format for the file. When the user enters the name of the output file, he/she is prompted to specify the parameters to be written to the file and the format to be used. The interfaces that enable the user to specify the format of the input and output files were described in detail in the JJM Systems reports B921015-U-2R01 and B921016-U-2R02. The user can specify how many single wavelength techniques are to be applied to the sample at each wavelength and how many multiple wavelength techniques are to be applied to the entire sample by clicking on the required option in the box labelled, "How Many Techniques To Test." It should be noted that the interface does not allow the user to select a different number of single and multiple wavelength techniques. The user can also specify whether all or half the restricted historical data units should be used for calculating both the error term and the coefficients when the techniques are applied to the samples of unknown cover type data.

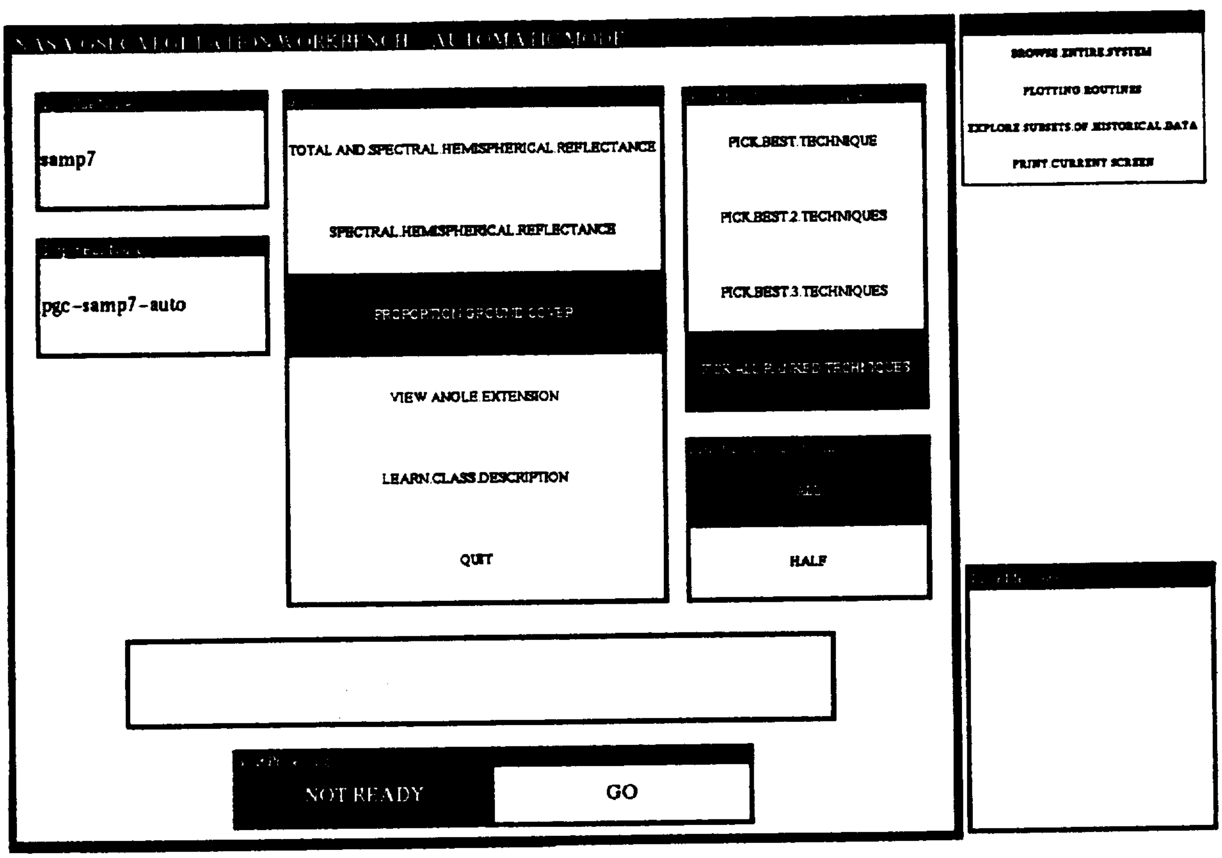


Figure 3-1

Selecting the Subgoal PROPORTION.GROUND.COVER from the VEG "Automatic Mode" Top Level Screen

When the user left clicks on "GO", the unknown cover type data is read from the file. The data is processed using the same sequence of steps as in the VEG "Research Mode." The results are written to the named file using the specified format.

SECTION 4.0

TESTING AND RESULTS

All the options of the VEG subgoal PROPORTION.GROUND.COVER were tested. These included testing the generation, ranking and execution of all the new single wavelength and multiple wavelength techniques. The selection of techniques both by the user and by the system were tested. In every test,the restricted data set was created automatically by the system. The tests included the subgoal PROPORTION.GROUND.COVER in the VEG "Automatic Mode" as well as the "Research Mode." All the tests were successful, showing that the system was working correctly. The tests are described in detail in this section. The output files produced by the test runs are presented in Appendix B.

4.1 TEST 1

The first test was designed to test the overall framework of the VEG subgoal PROPORTION.GROUND.COVER, and the operation of the rules to generate the single wavelength and multiple wavelength techniques. SAMPLE4 from the data base of unknown cover types within VEG was selected as the sample for this run. SAMPLE4 has four random view angles at wavelength 0.68 μm and a nadir view angle at wavelength 0.92 μm. All the single wavelength techniques are applicable to a sample with four view angles. This sample was selected so that the maximum number of techniques could be generated.

The data was processed by carrying out the steps in the PROPORTION.GROUND.COVER menu, as shown in Figure 2-1 and described in Section 2. The restricted data set was created automatically by the system.

The user opted to have the system generate the single wavelength techniques. The rules in the rulebase PROPORTION.GROUND.COVER.SINGLE.WAVELENGTH. RULES were run. The techniques PGC.NEAR.NADIR, PGC.NORMAN.PLUS, PGC.2OFF.NADIR.ANGLE.0, PGC.2OFF.NADIR.ANGLE.1, PGC.2OFF.NADIR.ANGLE.2, PGC.2OFF.NADIR. ANGLE.3, PGC.2OFF.NADIR.ANGLE.4 and PGC.2OFF.NADIR.ANGLE.5 were correctly chosen for the data at wavelength 0.68 μm. The technique PGC.NEAR.NADIR was the only technique selected for the data at wavelength 0.92 μm. This was as expected.

The single wavelength techniques were ranked, and the user chose to select the best three techniques at each wavelength. The best three techniques at wavelength 0.68 μm were PGC.NORMAN.PLUS, PGC.2OFF.NADIR.ANGLE.0 and PGC.2OFF.NADIR.ANGLE.1. The technique PGC.NEAR.NADIR was the only technique generated for the wavelength 0.92-μm. It was the only technique selected at this wavelength.

The techniques were executed by the system. In response to the question for each technique, the user indicated that half the restricted data set should be used for generating the coefficients and the other half should be used for estimating the error term.

The user elected to have the multiple wavelength techniques selected by the system. The rules in the rulebase PROPORTION.GROUND.COVER.MULTIPLE.WAVELENGTH.RULES were run. The technique PGC.NEAR.NADIR.ND was the only multiple wavelength technique generated for SAMPLE4. The technique PGC.NORMAN.PLUS.ND was not selected because the sample had only one view angle at wavelength 0.92 μm. The option PICK.BEST.TECHNIQUE was selected from the rank multiple wavelength techniques interface although selecting one, two, three or all techniques would have made no difference since only one technique was generated.

The technique PGC.NEAR.NADIR was executed on the sample. Half the restricted data set was used for generating the coefficients and the other half was used for estimating the error term.

The results were output on the screen. Figure 2-13 shows some of the results of this test. The proportion ground cover was estimated as 0.7916, with an error estimate of 0.1185 by the multiple wavelength technique PGC.NEAR.NADIR.ND. At wavelength 0.68 µm, the techniques PGC.NORMAN.PLUS, PGC.2OFF.NADIR.ANGLE.0 and PGC.2.OFF.NADIR.ANGLE.1 provided estimates of proportion ground cover of 0.7917, 0.7933 and 0.0527 with error estimate 0.1295, 0.1351 and 0.9416, respectively. At wavelength 0.92 using the technique PGC.NEAR.NADIR the proportion ground cover was estimated as 0.6124 with an error estimate of 0.2002. The NASA GSFC technical representative advised that errors as high as 45% are not uncommon in the measuring of reflectance data. Given the inherent inaccuracy of the reflectance data, most of the results were acceptable. The exception was the estimate of 0.0527 from applying the technique PGC.2OFF.NADIR.ANGLE1 to the data at wavelength 0.68 µm. Careful examination did not reveal any errors in coding. The result appears to have been calculated correctly. It has been referred to the NASA GSFC technical representative for interpretation. Test 1 provided evidence that all the steps in calculating the proportion ground cover were working correctly.

4.2 TEST 2

In test 1, all the single and multiple wavelength techniques were generated automatically by the system, by running rules. The purpose of test 2 was to test the interfaces that allow the user to select the techniques manually.

SAMPLE3 was chosen as the sample for test 2. This sample has three full strings at wavelength 0.68 µm and a nadir value at wavelength 0.92 µm. All the steps in the subgoal PROPORTION.GROUND.COVER were selected in turn. The "SELECTED.BY.USER" option was chosen for generating both the single wavelength and multiple wavelength techniques. In each case, the user attempted to select all the techniques. The system permitted the user to select the single wavelength techniques PGC.NORMAN.PLUS and PGC.NEAR.NADIR at wavelength 0.68 µm. It displayed an error message instead of selecting the technique when the user left clicked on each of the PGC.2OFF.NADIR.ANGLE techniques. These techniques were unsuitable for the sample because it had more than four view angles so the system was operating correctly. The user was only permitted to select the technique PGC.NEAR.NADIR at wavelength 0.92 µm, because the sample had only one view angle. The system allowed the user to select the multiple wavelength technique PGC.NEAR.NADIR.ND. It did not allow the user to select the technique PGC.NORMAN.PLUS.ND, because the sample had less than three view angles in the near-infrared band.

The results were displayed on the screen and output to a file which is listed in Appendix B. All the results were acceptable. This test showed that the interfaces for selecting the techniques manually were operating correctly.

4.3 TEST 3

This test was designed to test the subgoal PROPORTION.GROUND.COVER in the VEG "Automatic Mode." The proportion ground cover for SAMPLE7 was calculated using the VEG "Research Mode" and then using the VEG "Automatic Mode." The results were compared.

In the first part of this test, the VEG "Research Mode" was selected and SAMPLE7 was picked as the unknown cover type to be investigated. Each step in the VEG subgoal PROPORTION.GROUND.COVER was selected in turn.

A crude estimate of the proportion ground cover of a sample is calculated in the step CHARACTERIZE.TARGET. However, SAMPLE7 contains insufficient data for this crude estimate to be made. The restricted data set was created automatically. This operation uses the crude estimate of the proportion ground cover calculated in the previous step. When the estimate of ground cover is not available, the user is prompted to enter an estimate of the value. In test 3, the user entered the value 0.5 in response to this prompt. This value was chosen so that a direct comparison could be made between the results of this part of the test and running the same test in the VEG "Automatic Mode." If VEG is running in "Automatic Mode," and an estimate of proportion ground cover is required but it is not available, VEG uses the value 0.5.

Both the single wavelength and multiple wavelength techniques were generated by the system. In each case, all the ranked techniques were selected. The techniques were executed using half the restricted historical data units to calculate the coefficients and half the restricted historical data units to calculate the error estimate. The results were displayed on the screen and written to the file "pgc-samp7" which is listed in Appendix B.

In the second part of this test, VEG was run in the "Automatic Mode." The file "samp7" was selected as the input file. This file contains the same data as the VEG units SAMPLE7, W13 and W14. The output file was named "pgc-samp7-auto." The options "PICK.ALL.RANKED.TECHNIQUES" and "HALF" (see Figure 3-1) were selected. The user clicked on "GO," and the data was processed.

At the end of the run, the user compared the files "pgc-samp7" and "pgc-samp7-auto." These files were found to be the same. It was concluded that the subgoal PROPORTION.GROUND.COVER was working correctly in the VEG "Automatic Mode" since it gave the same results as the VEG "Research Mode."

4.4 TEST 4

This test was also run in both VEG modes as further proof that they were operating correctly. The test was designed primarily to test the multiple wavelength technique PGC.NORMAN.PLUS.ND. This technique was not generated in any of the previous tests. A new sample called SAMPLE10 was added to VEG. This sample contained data at more than four view angles in both the red and the near-infrared bands. The multiple wavelength technique PGC.NORMAN.PLUS.ND was suitable for this sample.

The test was run first in the VEG "Research Mode" and then in the "Automatic Mode." In both runs, the techniques were generated automatically, all the generated techniques were selected, and all the restricted historical data units were used for calculating both the coefficients and the error estimates. The results of the first and second runs were written to the files "pgc-samp10" and "pgc-samp10-auto," respectively.

In both runs, the multiple wavelength technique PGC.NORMAN.PLUS.ND was selected. Using this technique, the proportion ground cover was estimated as 0.7691 with an error estimate of 0.1438. This result compared with a proportion ground cover estimate of 0.7724 and an error estimate of 0.1182 using the multiple wavelength technique PGC.NEAR.NADIR.ND. This result was also similar to the results obtained using the single wavelength techniques on the same sample. It was concluded that the technique PGC.NORMAN.PLUS.ND was operating correctly. Both runs in this test produced the same results. This was further evidence that the subgoal PROPORTION.GROUND.COVER was working correctly in the VEG "Research Mode."

SECTION 5.0

CONCLUSIONS

The VEG subgoal PROPORTION.GROUND.COVER was completed in both the VEG "Research Mode" and the VEG "Automatic Mode." Several single wavelength techniques for estimating proportion ground cover were implemented. The framework to incorporate multiple wavelength techniques into VEG was constructed. Two multiple wavelength techniques for estimating ground cover were implemented.

All the options in the VEG subgoal PROPORTION.GROUND.COVER were tested. The tests were successful, showing that the system was working correctly.

REFERENCES

1. Kimes, D. S., Harrison, P. R. and Ratcliffe, P. A. 1991. A Knowledge-Based Expert System for Inferring Vegetation Characteristics. International Journal of Remote Sensing: Vol 12, 10, pp. 1987-2020.

2. Kimes, D. S., Harrison, P. A. and Harrison, P. R. 1992. New Developments of a Knowledge Based System (VEG) for Inferring Vegetation Characteristics. International Geoscience and Remote Sensing Symposium, Houston, Texas, May 1992.

APPENDIX A

LISP CODE FOR THE VEG SUBGOAL PROPORTION.GROUND.COVER

```
;;; veg-methods3.lisp
;;;
;;; Written by Ann Harrison
;;; Created 10th March 1992
;;; Last Modified 28th October 1992

(in-package 'kee)

(defun pgc.p ()
"Returns t if the subgoal PROPORTION.GROUND.COVER has been selected and nil
otherwise."
  (or (eq 'PROPORTION.GROUND.COVER (get.value 'methods 'goals))
     (eq 'PROPORTION.GROUND.COVER (get.value 'automatic.process 'auto.goals))))

(defun proportion-ground-cover ()
"Opens the panel to display the main menu for the subgoal
PROPORTION.GROUND.COVER."
 (remove.all.values 'proportion.ground.cover 'pgc.menu)
 (put.value 'proportion.ground.cover 'error.message "")
 (unitmsg 'viewport-portion.ground.cover.1 'open-panel!))

;;;---------------------------------------------------------------------------
;;; Methods for generating appropriate single wavelength techniques for
;;; proportion ground cover
;;;---------------------------------------------------------------------------

(defun user-pick-pgc-techniques ()
"Opens the panel that controls the selection of the single wavelength
techniques by the user. The panel contains a pushbutton. When the user left
clicks on the pushbutton, the system moves on to allow the user to select the
techniques at the next wavelength."
 (unitmsg 'viewport-6.generate.techniques.3 'open-panel!)
 (dolist (thisunit (get.values 'estimate.hemispherical.reflectance
                      'current.sample.wavelengths)
         (all-generate-techniques-finished-message))
  (put.value 'estimate.hemispherical.reflectance
        'current.wavelength thisunit)
  (user-pick-pgc-techniques-aux)
  (remove.all.values '6.generate.techniques 'push.button)
  (wait-for-mouse-gt)))

(defun user-pick-pgc-techniques-aux ()
"Calls a function to reset the values of images on the screen. Then opens the
screen that allows the user to select the single wavelength proportion ground
cover techniques."
 (reset-initial-values-pick-pgc-techniques)
 (unitmsg 'viewport-portion.ground.cover.5 'open-panel!))
```

```
(defun reset-initial-values-pick-pgc-techniques ()
"Resets the values of slots required by the screen that allows the user to
select the single wavelength proportion ground cover techniques."
  (remove.all.values 'proportion.ground.cover 'selected.techniques)
  (put.value '6.generate.techniques 'error.message "")
  (put.value '6.generate.techniques 'description.of.technique "")
  (put.value 'proportion.ground.cover 'action.on.selecting.techniques
     'select.techniques))

(defun pick-selected-values-pgc ()
"Stores the selected techniques in the current wavelength level unit. Displays
a list of selected techniques on the screen. Not - these techniques are
displayed until the user left clicks on the pushbutton."
  (let ((techs (get.values 'proportion.ground.cover 'selected.techniques))
        (current-wave (get.value 'estimate.hemispherical.reflectance
                          'current.wavelength)))
   (unless (null current-wave)
     (put.values current-wave 'techniques techs))
   (remove.all.values '6.generate.techniques 'automatic.or.manual)
   (tech-message (format ()
             "Techniques selected for sample at wavelength ~S are :-~{ ~S~}"
                (wav current-wave)(get-unit-names techs)))))

(defun pgc.near.nadir.ok (sample)
"Returns t if the techniques pgc.near.nadir is suitable for the sample and nil
otherwise. The technique is suitable if the sample has at least one view
angle."
     (>= (get.value sample 'number.view.angles) 1))

(defun pgc.norman.plus.ok (sample)
"Returns t if the techniques pgc.nornam.plus is suitable for the sample and nil
otherwise. The technique is suitable if the sample has at least three view
angles."
     (>= (get.value sample 'number.view.angles) 3))

;;;---------------------------------------------------------------------------
;;; Basic functions required for calculating error proportion in proportion
;;; ground cover calculations
;;;---------------------------------------------------------------------------

(defun get-pgc-error-prop (tech-method this-rhd coeffs)
"Returns the calculated result and error proportion after applying a proportion
ground cover technique to a sample of restricted historical data. Returns the
value 0 for the error proportion if the true result is zero."
  (let ((true-result (find-true-pgc this-rhd))
        (calc-result (funcall tech-method this-rhd coeffs)))
   (values calc-result
           (if (zerop true-result)
               0
               (/ (- true-result calc-result) true-result)))))
```

```lisp
(defun find-true-pgc (this-rhd)
"Returns the true proportion ground cover for a restricted historical data
sample."
  (get.value (get.value this-rhd 'cover.type) 'ground.cover))

;;;-----------------------------------------------------------------------
;;; Techniques to calculate proportion ground cover
;;;-----------------------------------------------------------------------

(defun tech-pgc-near-nadir (thisunit coeffs)
"Method for the technique pgc.near.nadir. The argument thisunit is a unit
containing reflectance data for an unknown cover type at the wavelength."
 (put-in-range (+ (aref coeffs 0)
                  (* (third (get-nearest-to-nadir
                              (get.value thisunit 'reflectance.data)))
                     (aref coeffs 1)))))

(defun coeffs-pgc-near-nadir (data)
"Method for calculating the coefficients for the technique pgc.near.nadir.
The argument data is a list of rhd units to be used for calculating the
coefficients."
 (let* ((nearest-to-nadir-position
          (get-nearest-to-nadir-position (first data)))
         (the-a-data
        (mapcar #'(lambda (unit)
                   (third (nth nearest-to-nadir-position
                                (get.value unit 'reflectance.data))))
                 data))
         (the-b-data (get-true-pgc-values data)))
   (least-squares (make-a-matrix the-a-data)
                  (make-array (length the-b-data)
                               :initial-contents the-b-data))))

(defun get-nearest-to-nadir-position (this-unit)
"Returns the position in the reflectance data of the nearest view angle to the
nadir."
 (let ((reflectance-data (get.value this-unit 'reflectance.data)))
   (position (get-nearest-to-nadir reflectance-data) reflectance-data
             :test #'equal)))

(defun get-nearest-to-nadir (reflectance-data)
"Returns the nearest view angle to the nadir in a set of reflectance data."
 (let* ((first-point (first reflectance-data))
        (best-distance (sqrt (+ (square (first first-point))
                                (square (second first-point)))))
        (best-point first-point))
   (dolist (point (rest reflectance-data) best-point)
    (let ((this-distance (sqrt (+ (square (first point))
                                  (square (second point))))))
        (cond ((zerop this-distance)
              (return-from get-nearest-to-nadir point))
             ((< this-distance best-distance)
              (setf best-distance this-distance)
              (setf best-point point))))))
```

```
;;; The technique pgc-norman-plus uses the function tech-norman-plus from the
;;; file veg-methods.lisp, in the section containing the code for the
;;; techniques for estimating spectral hemispherical reflectance.

(defun tech-pgc-norman-plus (thisunit coeffs)
"Technique norman plus for proportion ground cover."
  (put-in-range (tech-norman-plus thisunit coeffs)))

(defun coeffs-pgc-norman-plus (data)
"Returns the coefficients for the technique norman plus for proportion ground
cover. The argument data is a list of rhd units to be used for calculating the
coefficients."
  (let* ((straight-norman-estimates (get-norman-estimates data))
       (the-b-data (get-true-pgc-values data)))
    (least-squares (make-a-matrix straight-norman-estimates)
                   (make-array (length the-b-data)
                               :initial-contents the-b-data))))


;;; The techniques pgc-2off-nadir-angle-* use the function tech-2off-nadir
;;; and related functions which can be found in the file veg-methods.lisp,
;;; in the section containing the code for the techniques for estimating
;;; spectral hemispherical reflectance.

(defun coeffs-pgc-2off-nadir (data m n)
"Returns the coefficients for a 2off nadir technique for calculating
proportion ground cover. The argument data is a list of rhd units to be used
for calculating the coefficients. The arguments m and n are the positions in
the list of reflectance data to be used in this technique."
   (let ((vector-1
        (mapcar #'(lambda (unit)
                (third (nth  m (get.value unit 'reflectance.data))))
                data))
      (vector-2
        (mapcar #'(lambda (unit)
                (third (nth n (get.value unit 'reflectance.data))))
                data))
      (the-b-data (get-true-pgc-values data)))
     (least-squares (make-a-matrix-3 vector-1 vector-2)
        (make-array (length the-b-data) :initial-contents the-b-data))))

(defun get-true-pgc-values (list-of-units)
"Returns a list of true proportion ground cover values for a list of units."
 (mapcar #'find-true-pgc list-of-units))

(defun tech-pgc-2off-nadir-angle-0 (thisunit coeffs)
"Technique 2 off nadir angle for proportion ground cover."
  (put-in-range (tech-2off-nadir-angle-0 thisunit coeffs)))

(defun coeffs-pgc-2off-nadir-angle-0 (data)
"Calls the function coeffs-pgc-2off-nadir to find the coefficients for applying
the function pgc-2off-nadir to the first and second view angles."
 (coeffs-pgc-2off-nadir data 0 1))
```

```
(defun tech-pgc-2off-nadir-angle-1 (thisunit coeffs)
"Technique 2 off nadir angle for proportion ground cover."
  (put-in-range (tech-2off-nadir-angle-1 thisunit coeffs)))

(defun coeffs-pgc-2off-nadir-angle-1 (data)
"Calls the function coeffs-pgc-2off-nadir to find the coefficients for applying
the function pgc-2off-nadir to the first and third view angles."
 (coeffs-pgc-2off-nadir data 0 2))

(defun tech-pgc-2off-nadir-angle-2 (thisunit coeffs)
"Technique 2 off nadir angle for proportion ground cover."
  (put-in-range (tech-2off-nadir-angle-2 thisunit coeffs)))

(defun coeffs-pgc-2off-nadir-angle-2 (data)
"Calls the function coeffs-pgc-2off-nadir to find the coefficients for applying
the function pgc-2off-nadir to the second and third view angles."
 (coeffs-pgc-2off-nadir data 1 2))

(defun tech-pgc-2off-nadir-angle-3 (thisunit coeffs)
"Technique 2 off nadir angle for proportion ground cover."
  (put-in-range (tech-2off-nadir-angle-3 thisunit coeffs)))

(defun coeffs-pgc-2off-nadir-angle-3 (data)
"Calls the function coeffs-pgc-2off-nadir to find the coefficients for applying
the function pgc-2off-nadir to the first and fourth view angles."
 (coeffs-pgc-2off-nadir data 0 3))

(defun tech-pgc-2off-nadir-angle-4 (thisunit coeffs)
"Technique 2 off nadir angle for proportion ground cover."
  (put-in-range (tech-2off-nadir-angle-4 thisunit coeffs)))

(defun coeffs-pgc-2off-nadir-angle-4 (data)
"Calls the function coeffs-pgc-2off-nadir to find the coefficients for applying
the function pgc-2off-nadir to the second and fourth view angles."
 (coeffs-pgc-2off-nadir data 1 3))

(defun tech-pgc-2off-nadir-angle-5 (thisunit coeffs)
"Technique 2 off nadir angle for proportion ground cover."
  (put-in-range (tech-2off-nadir-angle-5 thisunit coeffs)))

(defun coeffs-pgc-2off-nadir-angle-5 (data)
"Calls the function coeffs-pgc-2off-nadir to find the coefficients for applying
the function pgc-2off-nadir to the third and fourth view angles."
 (coeffs-pgc-2off-nadir data 2 3))

;;; -----------------------------------------------------------------------
;;; Multiple wavelength techniques
;;; -----------------------------------------------------------------------

;;;------------------------------------------------------------------------
;;; Generating Multiple Wavlength Techniques
;;;------------------------------------------------------------------------
```

```
(defun pgc.near.nadir.nd.ok (sample)
"Returns t if the selected sample is suitable for the technique
pgc.near.nadir.nd and nil otherwise.  A sample is suitable if it has
reflectance data in both the red and near infrared bands."
   (declare (ignore sample))
   (let ((waves (mapcar #'(lambda (unit) (get.value unit 'wavelength))
                        (get.values 'estimate.hemispherical.reflectance
                          'current.sample.wavelengths))))
     ; Check required data in red band is available
     (dolist (wav waves
             (return-from pgc.near.nadir.nd.ok nil))
         (when (and (>= wav 0.63)(<= wav 0.69))
          (return-from nil)))
     ; Check required data in nir band is available
     (dolist (wav waves nil)
         (when (and (>= wav 0.76)(<= wav 1.1))
          (return-from pgc.near.nadir.nd.ok t)))))

(defun pgc.norman.plus.nd.ok (sample)
"Returns t if the selected sample is suitable for the technique
pgc.norman.plus.nd and nil otherwise.  A sample is suitable if it has
at least three view angles in both the red and near infrared bands."
   (declare (ignore sample))
   (let ((wave-units (get.values 'estimate.hemispherical.reflectance
                        'current.sample.wavelengths)))
     ; Check required data in red band is available
     (dolist (wav-unit wave-units
             (return-from pgc.norman.plus.nd.ok nil))
         (let ((wav (get.value wav-unit 'wavelength)))
          (when (and (>= wav 0.63)(<= wav 0.69)
                  (>= (get.value wav-unit 'number.view.angles) 3))
           (return-from nil))))
     ; Check required data in nir band is available
     (dolist (wav-unit wave-units nil)
         (let ((wav (get.value wav-unit 'wavelength)))
          (when (and (>= wav 0.76)(<= wav 1.1)
                  (>= (get.value wav-unit 'number.view.angles) 3))
           (return-from pgc.norman.plus.nd.ok t))))))

(defun pick-selected-pgc-mw-techniques ()
"Puts the selected techniques into the slot multiple.wavelength.techniques of
the unit proportion.ground.cover."
  (put.values 'proportion.ground.cover 'multiple.wavelength.techniques
     (get.values 'proportion.ground.cover 'selected.mw.techniques))
  (put.value 'proportion.ground.cover 'message
     "Finished selecting techniques"))
```

```
(defun user-pick-mw-techniques ()
"Initializes values and opens the interface that allows the user to select
the multiple wavelength, proportion ground cover techniques."
  (put.value 'proportion.ground.cover 'action.on.selecting.mw.techniques
     'select.techniques)
  (remove.all.values 'proportion.ground.cover 'selected.mw.techniques)
  (put.value '6.generate.techniques 'description.of.technique "")
  (put.value '6.generate.techniques 'error.message "")
  (unitmsg 'viewport-portion.ground.cover.2 'open-panel!))

(defun open-generate-mw-techniques-interface ()
"Opens the interface for generating the multiple wavelength techniques."
   (remove.all.values 'proportion.ground.cover 'gen.mw.tech.auto.or.manual)
   (put.value 'proportion.ground.cover 'message "")
   (unitmsg 'viewport-portion.ground.cover.3 'open-panel!))

(defun display-selected-mw-techniques ()
"Displays the multiple wavelength, proportion ground cover techniques selected
by the rules."
  (put.value 'proportion.ground.cover 'message (format ()
     "The selected techniques are~{ ~S~}"
     (get-unit-names
         (get.values 'proportion.ground.cover
            'multiple.wavelength.techniques)))))

;;;----------------------------------------------------------------------
;;; Methods for Ranking the Multiple Wavelength Techniques
;;;----------------------------------------------------------------------

(defun open-rank-mw-techniques-interface ()
"Opens the interface that displays the ranked multiple wavelength techniques
and allows the user to specify how many techniques are to be used."
  (remove.all.values 'proportion.ground.cover 'action.on.ranking.mw.techniques)
  (unitmsg 'proportion.ground.cover 'evaluate.samples)
  (put.value 'proportion.ground.cover 'message (format ()
          "Multiple wavelength techniques are~{ ~S~}"
          (get-unit-names
           (get.values 'proportion.ground.cover
              'multiple.wavelength.techniques))))
  (unitmsg 'viewport-portion.ground.cover.4 'open-panel!))

(defun select-best-mw-techniques (num)
"Selects the best multiple wavelength techniques by reducing the set of
selected techniques as necessary."
  (let ((mw-techniques (get.values 'proportion.ground.cover
                          'multiple.wavelength.techniques)))
   (when (> (length mw-techniques) num)
     (put.values 'proportion.ground.cover 'multiple.wavelength.techniques
          (get-best num mw-techniques)))))

;;;----------------------------------------------------------------------
;;; Methods for Executing the Multiple Wavelength Techniques
;;;----------------------------------------------------------------------
```

```
(defun open-execute-mw-techniques-interface ()
"Opens the interface for executing the multiple wavelength techniques."
  (put.value '8.execute.techniques 'message "")
  (remove.all.values '8.execute.techniques 'push.button)
  (et-princ
"Executing multiple wavelength techniques - please wait................  ")
  (unitmsg 'viewport-8.execute.techniques.1 'open-panel!)
  (execute-mw-techniques)
  (et-princ
"Finished executing multiple wavelength techniques
- left mouse on QUIT to continue"))

(defun execute-mw-techniques()
"Displays an error message if no multiple wavelength techniques have been
selected.  Otherwise calls a function to execute the techniques."
  (let ((techs
          (get.values 'proportion.ground.cover
             'multiple.wavelength.techniques)))
    (if (null techs)
         (et-princ "No multiple wavelength techniques specified  ")
         (exe-mw-techniques techs))))

(defun exe-mw-techniques (techs)
"Creates the subclass unit to store the results for this sample - results for
different techniques will be subclasses of this unit. Controls the execution of
the different techniques on the sample."
  (let* ((current-sample (get.value 'estimate.hemispherical.reflectance
                              'current.sample))
          (parent (e.t.find-parent current-sample))
          (sample-name (unit.name current-sample)))
    (if (not (unitp parent))
         (et-princ (format ()
                         "Results have already been stored for the sample ~S"
                         sample-name))
         (let ((thisunit-name (string sample-name)))
          (dolist (this-tech techs)
            (let* ((new-unit-name
                       (string-append thisunit-name "-"
                                (string (unit.name this-tech))))
                      (new-unit (intern new-unit-name)))
              (create.unit new-unit 'veg parent)
              (apply-mw-tech this-tech current-sample new-unit)))
          (et-princ (format ()
                    "Finished executing techniques for the sample ~S  "
                    sample-name))))))
```

```
(defun apply-mw-tech (tech thisunit result-unit)
"Applies a multiple wavelength technique to a sample."
  (multiple-value-bind (i.e.rhd-use coeffs)
                 (get-mw-coefficients tech thisunit)
    (let* ((tech-method (get.value tech 'technique.method))
         (result (funcall tech-method thisunit coeffs))
         (error-term (calc-mw-error tech-method i.e.rhd-use coeffs
                       result-unit)))
      (store-mw-results result error-term coeffs result-unit))))

(defun calc-mw-error (tech-method ie.rhd coeffs result-unit)
"Function to calculate the error term for all rhd samples and then the rms
error for the sample for the particular technique. The results for each rhd
sample are also stored in new units."
  (let ((error-terms ()))
     (dolist (this-rhd ie.rhd)
         (let* ((ct (get.value this-rhd 'cover.type))
              (ct-name (unit.name ct))
              (new-unit
               (create.unit (gentemp (string ct-name)) 'veg nil
                         result-unit)))
            (multiple-value-bind (calc-result error-prop)
               (get-pgc-error-prop tech-method this-rhd coeffs)
              (push error-prop error-terms)
              (put.value new-unit 'cover.type ct)
              (put.value new-unit 'calc.spectral.hem.result
                    calc-result)
              (put.value new-unit 'shr.error.prop error-prop))))
     (calc-rms-error error-terms)))

(defun store-mw-results (result error-estimate coeffs result-unit)
"Function to store the main results for a sample and technique in the result
unit."
  (put.value result-unit 'calc.spectral.hem.refl (round-to-4-dp result))
  (put.value result-unit 'coeffs coeffs)
  (put.value result-unit 'shr.error.estimate (round-to-4-dp error-estimate)))

(defun put-in-range (result)
"Returns one if the value of the result is greater than one."
  (if (> result 1)
     1
     result))
```

```
(defun get-mw-coefficients (tech sample-unit)
"Method to calculate the coefficients for a technique and sample. Returns
the list of cover-types to be used as historical data, and the appropriate
ceofficients for this technique."
  (let* ((coeffs-p (get.value tech 'coeffs.p))
       (i.e.rhd (get.values sample-unit 'sample.level.i.e.rhd)))
      (if (not coeffs-p)          ;No coefficients for this technique
         (values i.e.rhd nil)     ;Use all rhd for calculations
         (let ((coeff-method (get.value tech 'coeff.method)))
            (multiple-value-bind (i.e.rhd-set coeffs-set)
               (ask-user-about-mw-coeffs i.e.rhd
                                (unit.name tech))
               (values i.e.rhd-set
                   (funcall coeff-method coeffs-set)))))))

(defun ask-user-about-mw-coeffs (i.e.rhd tech-name)
"Asks the user which historical data to use for coefficients and error term
estimation."
(cond ((and
        (eq (get.value 'methods 'processing.mode) 'research) ;research mode
        (> (length i.e.rhd) 3))                       ;large enough rds
     (unitmsg 'viewport-8.execute.techniques.2 'open-panel!)
     (remove.all.values '8.execute.techniques 'reply)
     (put.value '8.execute.techniques 'prompt
          (format ()
              "Executing technique ~S on sample.  The following options are available:-  (1) Use all
the historical data for calculating both the coefficients and the error estimate.  (2) Use half historical
data for calculating the coefficients and the other half for calculating the error estimate.  Make your
choice ->"
              tech-name))
     (et-wait-for-mouse)
     (if (eq 'all.historical.data (get.value '8.execute.techniques 'reply))
          (values i.e.rhd i.e.rhd)
          (split-i.e.rhd i.e.rhd)))
    ((and (eq (get.value 'methods 'processing.mode) 'automatic)
           (eq (get.value 'automatic.process 'auto.all.or.half) 'half))
     (split-i.e.rhd i.e.rhd))
    (t (values i.e.rhd i.e.rhd)))) ;use all if <=3 or selected in auto mode

(defun tech-pgc-near-nadir-nd (thisunit coeffs)
"Function for the proportion ground cover, near nadir, normalized difference
technique ."
 (put-in-range (+ (aref coeffs 0)
                 (* (near-nadir-nd thisunit)
                   (aref coeffs 1)))))

(defun coeffs-pgc-near-nadir-nd (data)
"Function to calculate the coefficients for the proportion ground cover, near
nadir, normalized difference technique ."
 (let ((the-a-data (mapcar #'near-nadir-nd data))
       (the-b-data (get-true-pgc-values data)))
   (least-squares (make-a-matrix the-a-data)
                 (make-array (length the-b-data)
                            :initial-contents the-b-data))))
```

```
(defun near-nadir-nd (sample)
"Returns the near nadir normalized difference."
 (multiple-value-bind (red-unit nir-unit)
    (get-red-and-nir-units sample)
   (let ((red-near-nadir (third (get-nearest-to-nadir
                                (get.value red-unit 'reflectance.data))))
         (nir-near-nadir (third (get-nearest-to-nadir
                                (get.value nir-unit 'reflectance.data)))))
    (/ (- red-near-nadir nir-near-nadir)
       (+ red-near-nadir nir-near-nadir)))))

(defun get-red-and-nir-units (sample)
"Returns the names of the units that are members of the sample and contain
data in the red and nir bands."
 (let ((red-unit nil)
       (nir-unit nil))
   (dolist (uni (unit.children sample 'member))
    (let ((wave (get.value uni 'wavelength)))
       (cond ((and (>= wave 0.63)(<= wave 0.68))
              (setf red-unit uni))
             ((and (>= wave 0.76)(<= wave 1.1))
              (setf nir-unit uni)))))
   (values red-unit nir-unit)))

(defun tech-pgc-norman-plus-nd (thisunit coeffs)
"Function for the proportion ground cover, norman plus, normalized difference
technique ."
  (put-in-range (+ (aref coeffs 0)
                   (* (norman-plus-nd thisunit)
                     (aref coeffs 1)))))

(defun coeffs-pgc-norman-plus-nd (data)
"Function to calculate the coefficients for the proportion ground cover, norman
plus, normalized difference technique ."
 (let ((the-a-data (mapcar #'norman-plus-nd data))
       (the-b-data (get-true-pgc-values data)))
   (least-squares (make-a-matrix the-a-data)
                  (make-array (length the-b-data)
                              :initial-contents the-b-data))))

(defun norman-plus-nd (sample)
"Returns the norman plus normalized difference."
 (multiple-value-bind (red-unit nir-unit)
    (get-red-and-nir-units sample)
   (let ((red-norman (apply-norman
                     (get.value red-unit 'reflectance.data)))
         (nir-norman (apply-norman
                     (get.value nir-unit 'reflectance.data))))
    (/ (- red-norman nir-norman)
       (+ red-norman nir-norman)))))
```

```
;;;-------------------------------------------------------------------------
;;; Method for Forming the Multiple Wavelength Results into a String Ready for
;;; Output
;;;-------------------------------------------------------------------------

(defun get-pgc-mw-results (sample)
"Returns a string containing the multiple wavelength proportion ground cover
results."
  (let ((sample-name (unit.name sample))
        (results "Proportion Ground Cover Multiple Wavelength Results:-  "))
    (dolist (tech (get.values 'proportion.ground.cover
                              'multiple.wavelength.techniques)
             results)
      (let* ((tech-name (unit.name tech))
             (result-unit (intern (string-append (string sample-name)
                                                 "-" (string tech-name)))))
        (setf results (string-append results
     (format () "Technique ~S Estimate ~,4F Error ~,4F Coefficients ~A  "
             tech-name
             (get.value result-unit 'calc.spectral.hem.refl)
             (get.value result-unit 'shr.error.estimate)
             (get-coeff-values result-unit)))))))
```

APPENDIX B

LISTINGS OF FILES PRODUCED BY THE TEST RUNS

Listing of the File pgc-samp4

Results for sample SAMPLE4
Sample input data:
Cover type "example of dense vegetation canopy": Solar Zenith Angle 45:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270 Proportion Ground Cover Multiple Wavelength Results:- Technique PGC.NEAR.NADIR.ND Estimate 0.7916 Error 0.1185 Coefficients 0.7606 -0.0368

Wavelength 0.68
Reflectance data ((0 0 0.043) (15 182 0.043) (35 45 0.043) (75 90 0.054))
Results:
Technique PGC.NORMAN.PLUS Estimate 0.7917 Error 0.1295 Coefficients 0.7393 1.1107
Technique PGC.2OFF.NADIR.ANGLE.0 Estimate 0.7933 Error 0.1351 Coefficients 0.8819 5.5944 -7.6552
Technique PGC.2OFF.NADIR.ANGLE.1 Estimate 0.0527 Error 0.9416 Coefficients 0.0154 0.0069 0.8592
Restricted Historical Data:
CT8-42-1 CT10-49-1 CT6-45-1 CT7-41-1 CT5-42-1 CT8-56-1 CT5-28-1 CT5-59-1 CT6-63-1 CT7-59-1
Data Characterization:
Nadir data is available
No strings found
Wavelength 0.92
Reflectance data ((0 0 0.5))
Results:
Technique PGC.NEAR.NADIR Estimate 0.6124 Error 0.2002 Coefficients 1.1142 -1.0037
Restricted Historical Data:
CT8-42-2 CT10-49-2 CT6-45-2 CT7-41-2 CT5-42-2 CT8-56-2 CT5-28-2 CT5-59-2 CT6-63-2 CT7-59-2
Data Characterization:
Nadir data is available
No strings found

Listing of the File pgc-samp3

Results for sample SAMPLE3
Sample input data:
Cover type "example of dense vegetation canopy": Solar Zenith Angle 45:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 2.6937: Ground Cover 0.9270 Proportion Ground Cover Multiple Wavelength Results:- Technique PGC.NEAR.NADIR.ND Estimate 0.7916 Error 0.1185 Coefficients 0.7606 -0.0368

Wavelength 0.92
Reflectance data ((0 0 0.5))
Results:
Technique PGC.NEAR.NADIR Estimate 0.6124 Error 0.2002 Coefficients 1.1142 -1.0037

Restricted Historical Data:
CT8-42-2 CT10-49-2 CT6-45-2 CT7-41-2 CT5-42-2 CT8-56-2 CT5-28-2 CT5-59-2 CT6-63-2 CT7-59-2
Data Characterization:
Nadir data is available
No strings found
Wavelength 0.68
Reflectance data ((0 0 0.043) (15 182 0.043) (15 7 0.043) (30 180 0.054) (30 5 0.043) (45 178 0.066) (45 356 0.044) (60 180 0.076) (60 355 0.054) (75 180 0.089) (75 2 0.067) (2 45 0.01) (15 46 0.03) (35 48 0.04) (50 45 0.05) (65 40 0.06) (15 225 0.02) (35 220 0.03) (50 227 0.04) (65 225 0.05) (2 90 0.01) (30 93 0.02) (45 85 0.03) (60 87 0.04) (15 270 0.02) (30 275 0.03) (45 270 0.05) (60 275 0.06))
Results:
Technique PGC.NORMAN.PLUS Estimate 0.7908 Error 0.1647 Coefficients 0.6443 3.0988
Technique PGC.NEAR.NADIR Estimate 0.7808 Error 0.1156 Coefficients 0.8480 -1.5625
Restricted Historical Data:
CT8-42-1 CT10-49-1 CT6-45-1 CT7-41-1 CT5-42-1 CT8-56-1 CT5-28-1 CT5-59-1 CT6-63-1 CT7-59-1
Data Characterization:
Nadir data is available
Strings:
COMPLETE FULL-string with 0 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth
INCOMPLETE FULL-string with 90 degrees azimuth

Listing of the File pgc-samp7

Results for sample SAMPLE7
Sample input data:
Cover type "Dense green vegetation canopy": Solar Zenith Angle 45:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
No target characterization Proportion Ground Cover Multiple Wavelength Results:- Technique PGC.NEAR.NADIR.ND Estimate 0.6463 Error 0.3210 Coefficients -0.2311 -1.1676

Wavelength 0.92
Reflectance data ((0 0 0.31))
Results:
Technique PGC.NEAR.NADIR Estimate 0.6697 Error 0.6921 Coefficients 0.4351 0.7567
Restricted Historical Data:
CT11-45-2 CT11-58-2 CT5-28-2 CT5-42-2 CT5-59-2 CT5-26-2 CT7-23-2 CT7-41-2 CT7-59-2 CT9-46-2
Data Characterization:
Nadir data is available
No strings found
Wavelength 0.68
Reflectance data ((45 0 0.044) (45 180 0.066) (60 0 0.054) (60 180 0.076))
Results:
Technique PGC.NORMAN.PLUS Estimate 0.6914 Error 0.4951 Coefficients 0.9188 -3.9599
Technique PGC.2OFF.NADIR.ANGLE.0 Estimate 0.7412 Error 0.4693 Coefficients 0.9112 4.2767 -5.4267
Technique PGC.2OFF.NADIR.ANGLE.1 Estimate 0.0645 Error 0.8630 Coefficients 0.0064 1.4208 -0.0817

Technique PGC.2OFF.NADIR.ANGLE.2 Estimate 0.7297 Error 0.4872 Coefficients 0.8634 -4.6006 3.1480
Technique PGC.2OFF.NADIR.ANGLE.3 Estimate 0.7312 Error 0.8346 Coefficients 0.8264 2.3740 -2.6269
Technique PGC.2OFF.NADIR.ANGLE.4 Estimate 0.6956 Error 0.1260 Coefficients 0.9684 -7.2136 2.6751
Technique PGC.2OFF.NADIR.ANGLE.5 Estimate 0.7821 Error 1.0843 Coefficients 0.7050 7.7256 -4.4745
Technique PGC.NEAR.NADIR Estimate 0.6633 Error 0.5509 Coefficients 0.8403 -4.0226
Restricted Historical Data:
CT11-45-1 CT11-58-1 CT5-28-1 CT5-42-1 CT5-59-1 CT5-26-1 CT7-23-1 CT7-41-1 CT7-59-1 CT9-46-1
Data Characterization:
Nadir data is not available
Strings:
NIL NIL-string with 0 degrees azimuth

Listing of the File pgc-samp7-auto

Results for sample FILE-SAMPLE-399
Sample input data:
Cover type NIL: Solar Zenith Angle 45:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
No target characterization Proportion Ground Cover Multiple Wavelength Results:- Technique PGC.NEAR.NADIR.ND Estimate 0.6463 Error 0.3210 Coefficients -0.2311 -1.1676

Wavelength 0.68
Reflectance data ((45 0 0.044) (45 180 0.066) (60 0 0.054) (60 180 0.076))
Results:
Technique PGC.NORMAN.PLUS Estimate 0.6914 Error 0.4951 Coefficients 0.9188 -3.9599
Technique PGC.2OFF.NADIR.ANGLE.0 Estimate 0.7412 Error 0.4693 Coefficients 0.9112 4.2767 -5.4267
Technique PGC.2OFF.NADIR.ANGLE.1 Estimate 0.0645 Error 0.8630 Coefficients 0.0064 1.4208 -0.0817
Technique PGC.2OFF.NADIR.ANGLE.2 Estimate 0.7297 Error 0.4872 Coefficients 0.8634 -4.6006 3.1480
Technique PGC.2OFF.NADIR.ANGLE.3 Estimate 0.7312 Error 0.8346 Coefficients 0.8264 2.3740 -2.6269
Technique PGC.2OFF.NADIR.ANGLE.4 Estimate 0.6956 Error 0.1260 Coefficients 0.9684 -7.2136 2.6751
Technique PGC.2OFF.NADIR.ANGLE.5 Estimate 0.7821 Error 1.0843 Coefficients 0.7050 7.7256 -4.4745
Technique PGC.NEAR.NADIR Estimate 0.6633 Error 0.5509 Coefficients 0.8403 -4.0226
Restricted Historical Data:
CT11-45-1 CT11-58-1 CT5-28-1 CT5-42-1 CT5-59-1 CT5-26-1 CT7-23-1 CT7-41-1 CT7-59-1 CT9-46-1
Data Characterization:
Nadir data is not available
Strings:
NIL NIL-string with 0 degrees azimuth
Wavelength 0.92
Reflectance data ((0 0 0.31))

Results:
Technique PGC.NEAR.NADIR Estimate 0.6697 Error 0.6921 Coefficients 0.4351 0.7567
Restricted Historical Data:
CT11-45-2 CT11-58-2 CT5-28-2 CT5-42-2 CT5-59-2 CT5-26-2 CT7-23-2 CT7-41-2 CT7-59-2 CT9-46-2
Data Characterization:
Nadir data is available
No strings found

Listing of the File pgc-samp10

Results for sample SAMPLE10
Sample input data:
Cover type "NIL": Solar Zenith Angle 42:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 1.6186: Ground Cover 0.8256 Proportion Ground Cover Multiple Wavelength Results:- Technique PGC.NORMAN.PLUS.ND Estimate 0.7691 Error 0.1438 Coefficients -0.0316 -1.0491 Technique PGC.NEAR.NADIR.ND Estimate 0.7724 Error 0.1182 Coefficients 0.0124 -0.9994

Wavelength 0.68
Reflectance data ((0 0 0.0425) (15 0 0.0433) (15 45 0.0443) (15 90 0.0461) (15 135 0.0443) (15 180 0.0424) (15 225 0.0397) (15 270 0.0415) (15 315 0.0406) (30 0 0.0433) (30 45 0.0461) (30 90 0.0461) (30 135 0.0516) (30 180 0.0535) (30 225 0.0507) (30 270 0.0461) (30 315 0.0443) (45 0 0.0443) (45 45 0.048) (45 90 0.0507) (45 135 0.0553) (45 180 0.0664) (45 225 0.0599) (45 270 0.0507) (45 315 0.0461) (60 0 0.0535) (60 45 0.0581) (60 90 0.0627) (60 135 0.063599996) (60 180 0.0756) (60 225 0.0655) (60 270 0.059) (60 315 0.0535) (75 0 0.0673) (75 45 0.0738) (75 90 0.0775) (75 135 0.0821) (75 180 0.0894) (75 225 0.0793) (75 270 0.0728) (75 315 0.0784))
Results:
Technique PGC.NORMAN.PLUS Estimate 0.7637 Error 0.1896 Coefficients 0.9323 -3.0073
Technique PGC.NEAR.NADIR Estimate 0.7813 Error 0.1476 Coefficients 1.0488 -6.2942
Restricted Historical Data:
CT6-45-1 CT7-41-1 CT5-42-1 CT8-42-1 CT10-49-1 CT5-26-1 CT5-28-1 CT8-56-1 CT10-28-1 CT11-58-1
Data Characterization:
Nadir data is available
Strings:
COMPLETE FULL-string with 0 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth
COMPLETE FULL-string with 90 degrees azimuth
Wavelength 0.92
Reflectance data ((0 0 0.3124) (15 0 0.3236) (15 45 0.3275) (15 90 0.31689999) (15 135 0.3002) (15 180 0.3242) (15 225 0.3258) (15 270 0.3125) (15 315 0.3242) (30 0 0.3426) (30 45 0.3593) (30 90 0.3549) (30 135 0.3515) (30 180 0.3699) (30 225 0.3683) (30 270 0.3549) (30 315 0.3482) (45 0 0.3805) (45 45 0.4028) (45 90 0.4129) (45 135 0.39) (45 180 0.4592) (45 225 0.4252) (45 270 0.4006) (45 315 0.39) (60 0 0.4586) (60 45 0.4726) (60 90 0.4771) (60 135 0.4854) (60 180 0.5451) (60 225 0.4938) (60 270 0.462) (60 315 0.4536) (75 0 0.5278) (75 45 0.5451) (75 90 0.5401) (75 135 0.5546) (75 180 0.5741) (75 225 0.5479) (75 270 0.5022) (75 315 0.5122))

Results:
Technique PGC.NORMAN.PLUS Estimate 0.7926 Error 0.1734 Coefficients 0.4582 0.8010
Technique PGC.NEAR.NADIR Estimate 0.7625 Error 0.1803 Coefficients 0.5751 0.6000
Restricted Historical Data:
CT6-45-2 CT7-41-2 CT5-42-2 CT8-42-2 CT10-49-2 CT5-26-2 CT5-28-2 CT8-56-2 CT10-28-2 CT11-58-2
Data Characterization:
Nadir data is available
Strings:
COMPLETE FULL-string with 0 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth
COMPLETE FULL-string with 90 degrees azimuth

Listing of the File pgc-samp10-auto

Results for sample FILE-SAMPLE-629
Sample input data:
Cover type NIL: Solar Zenith Angle 42:
Ground Cover NIL: Leaf Area Index NIL: Proportion Green NIL:
Dry Biomass NIL: Wet Biomass NIL: Height NIL
Target characterization: Leaf Area Index 1.6186: Ground Cover 0.8256 Proportion Ground Cover Multiple Wavelength Results:- Technique PGC.NORMAN.PLUS.ND Estimate 0.7691 Error 0.1438 Coefficients -0.0316 -1.0491 Technique PGC.NEAR.NADIR.ND Estimate 0.7724 Error 0.1182 Coefficients 0.0124 -0.9994

Wavelength 0.92
Reflectance data ((0 0 0.3124) (15 0 0.3236) (15 45 0.3275) (15 90 0.31689999) (15 135 0.3002) (15 180 0.3242) (15 225 0.3258) (15 270 0.3125) (15 315 0.3242) (30 0 0.3426) (30 45 0.3593) (30 90 0.3549) (30 135 0.3515) (30 180 0.3699) (30 225 0.3683) (30 270 0.3549) (30 315 0.3482) (45 0 0.3805) (45 45 0.4028) (45 90 0.4129) (45 135 0.39) (45 180 0.4592) (45 225 0.4252) (45 270 0.4006) (45 315 0.39) (60 0 0.4586) (60 45 0.4726) (60 90 0.4771) (60 135 0.4854) (60 180 0.5451) (60 225 0.4938) (60 270 0.462) (60 315 0.4536) (75 0 0.5278) (75 45 0.5451) (75 90 0.5401) (75 135 0.5546) (75 180 0.5741) (75 225 0.5479) (75 270 0.5022) (75 315 0.5122))
Results:
Technique PGC.NORMAN.PLUS Estimate 0.7926 Error 0.1734 Coefficients 0.4582 0.8010
Technique PGC.NEAR.NADIR Estimate 0.7625 Error 0.1803 Coefficients 0.5751 0.6000
Restricted Historical Data:
CT6-45-2 CT7-41-2 CT5-42-2 CT8-42-2 CT10-49-2 CT5-26-2 CT5-28-2 CT8-56-2 CT10-28-2 CT11-58-2
Data Characterization:
Nadir data is available
Strings:
COMPLETE FULL-string with 0 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth
COMPLETE FULL-string with 90 degrees azimuth

Wavelength 0.68
Reflectance data ((0 0 0.0425) (15 0 0.0433) (15 45 0.0443) (15 90 0.0461) (15 135 0.0443) (15 180 0.0424) (15 225 0.0397) (15 270 0.0415) (15 315 0.0406) (30 0 0.0433) (30 45 0.0461) (30 90 0.0461) (30 135 0.0516) (30 180 0.0535) (30 225 0.0507) (30 270 0.0461) (30 315 0.0443) (45 0 0.0443) (45 45 0.048) (45 90 0.0507) (45 135 0.0553) (45 180 0.0664) (45 225 0.0599) (45 270 0.0507) (45 315 0.0461) (60 0 0.0535) (60 45 0.0581) (60 90 0.0627) (60 135 0.063599996) (60 180 0.0756) (60 225 0.0655) (60 270 0.059) (60 315 0.0535) (75 0 0.0673) (75 45
0.0738) (75 90 0.0775) (75 135 0.0821) (75 180 0.0894) (75 225 0.0793) (75 270 0.0728) (75 315 0.0784))
Results:
Technique PGC.NORMAN.PLUS Estimate 0.7637 Error 0.1896 Coefficients 0.9323 -3.0073
Technique PGC.NEAR.NADIR Estimate 0.7813 Error 0.1476 Coefficients 1.0488 -6.2942
Restricted Historical Data:
CT6-45-1 CT7-41-1 CT5-42-1 CT8-42-1 CT10-49-1 CT5-26-1 CT5-28-1 CT8-56-1 CT10-28-1 CT11-58-1
Data Characterization:
Nadir data is available
Strings:
COMPLETE FULL-string with 0 degrees azimuth
COMPLETE FULL-string with 45 degrees azimuth
COMPLETE FULL-string with 90 degrees azimuth

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle An Expert System Shell for Inferring Vegetation Characteristics - Implementation of Additional Techniques (Task E)		5. Report Date October 1992
		6. Performing Organization Code
7. Author(s) P. Ann Harrison		8. Performing Organization Report No. B921019-U-2R04
		10. Work Unit No. 462-61-14
9. Performing Organization Name and Address JM Systems, Inc. One Ivybrook Blvd., Suite 190 Ivyland, PA 18974		11. Contract or Grant No. NAS5-30127
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001 NASA/Goddard Space Flight Center Greenbelt, MD 20771		13. Type of Report and Period Covered Task Report for Task E September - October 1992
		14. Sponsoring Agency Code

15. Supplementary Notes

The Lisp and KEE code for this work is available on a Sun Cartridge Tape.

16. Abstract

The NASA VEGetation Workbench (VEG) is a knowledge based system that infers vegetation characteristics from reflectance data. The VEG subgoal PROPORTION.GROUND.COVER has been completed and a number of additional techniques that infer the proportion ground cover of a sample have been implemented. Some techniques operate on sample data at a single wavelength. The techniques previously incorporated in VEG for other subgoals operated on data at a single wavelength so implementing the additional single wavelength techniques required no changes to the structure of VEG. Two techniques which use data at multiple wavelengths to infer proportion ground cover were also implemented. This work involved modifying the structure of VEG so that multiple wavelength techniques could be incorporated. All the new techniques were tested using both the VEG "Research Mode" and the "Automatic Mode."

17. Key Words (Suggested by Author(s)) EXPERT SYSTEM, ARTIFICIAL INTELLIGENCE, REMOTE SENSING		18. Distribution Statement UNCLASSIFIED - UNLIMITED	
19. Security Classif. (of this report) UNCLASSIFIED	20. Security Classif. (of this page) UNCLASSIFIED	21. No. of pages	22. Price

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171