LARS INFORMATION NOTE 051368

ON THE APPLICATION OF MAN-MACHINE COMPUTING SYSTEMS TO PROBLEMS IN REMOTE SENSING*

Philip H. Swain and Don A. Germann
Laboratory for Agricultural Remote Sensing
Purdue University, West Lafayette, Indiana

Summary

This paper describes a data processing system which coordinates a broad range of pattern recognition related techniques to aid in the design of pattern classifiers. It is shown how development of the user-system interface enhances the value of the system as a tool for research in agricultural remote sensing.

1. Introduction

Research at the Laboratory for Agricultural Remote Sensing (LARS) concerns the analysis by pattern recognition techniques of multispectral remote sensing data collected by ground-based, airborne, and (ultimately) satellite-borne instruments. A rather broad class of applications of this research is possible (not all of which have even been clearly defined as yet), and it is generally the case that each application requires a pattern recognition system with significantly distinctive characteristics. In such a situation it is necessary to have at hand an efficient and flexible method for designing pattern classifiers; in particular, for performing statistical analysis, feature selection, and calculation of other parameters needed for the realization of the classifiers. This paper describes such a method and its implementation in the form of a system of computer programs. An important feature of this system is the considerable degree of user - system interaction through which is achieved the flexibility required by the research environment.

2. Data Handling and Data Analysis in Remote Sensing: General

Figure 1 shows a block diagram of the overall data flow for the Laboratory for Agricultural Remote Sensing Data Processing System (LARSYS). The principal data input is multispectral data collected on analog tape by a multichannel optical-mechanical scanner and tape recorder mounted aboard an aircraft.** The LARS Aircraft Data Handling Processor (LARSYSAH)

---

* This work was supported by the U. S. Department of Agriculture under Contract No. 12-14-100-8926(20).

** The airborne optical/infrared scanning equipments used in this research were made available by the U. S. Army Electronics Command (USAECOM), Fort Monmouth, N.J. on a no-cost basis to the University of Michigan (who collected the data) for use on contracts administered by USAECOM.

(Presented at the Eleventh Midwest Symposium on Circuit Theory, Notre Dame University, Notre Dame, Indiana, May 13-14, 1968.)

prepares the data for use by the researcher: The data are edited, digitized, and calibrated and recorded on digital tape in a packed format (to reduce the physical volume). To make the data readily accessible to the user, line-sample coordinates (much like x-y coordinates) are added during the digitization process. A special computer subroutine is available which will read any desired area of data (specified by a set of line-sample coordinates) into core memory and pass it to the user's program in unpacked form.

Also available as part of LARSYSAH is a program which prints grey-level displays of selected data on a computer line printer. These displays, which are similar to black and white photographs of the ground areas over which the data were collected, are useful in coordinating the ground truth (see below) with the multispectral scanner data.[1,2]

The other form of data utilized is "ground truth," which is collected on film and in the form of detailed written field reports. Ground truth, including such information as crop species, crop varieties, soil types, percent ground cover, etc., is cataloged and made available to the researcher in convenient form by the LARS Ground Truth Processor (LARSYSGT).

The principal concern of this paper is the LARS Aircraft Data Analysis and Classifier Design System (LARSYSAA) which performs the function of pattern classifier design based on the data from the aircraft mounted multispectral scanner.

3. The LARS Aircraft Data Analysis and Classifier Design System (LARSYSAA)

3.1 The User - System Interface: Introduction

There are at least three important reasons why the user-system interface has received considerable attention in the development of the analysis and classifier design system.

1. Optimal classifier design requires a substantial amount of interaction between the various phases of the design system.* At the present state-of-the-art, this interaction is best coordinated by the researcher.

---

*In this paper three phases of classifier design are distinguished: statistical analysis, in which general statistical properties of the data are measured; feature selection, in which an optimal set of features are selected for use in the recognition process; and classifier synthesis, in which the classifier is designed (or "trained") and tested using the results of the preceding phases. The precise nature of the computations carried out in each phase depends on the particular feature selection and pattern recognition algorithms used by the researcher.

2. Remote sensing applications invariably involve huge masses of data. As a result, the quantity of data input and results output required for a classifier design task consumes a considerable amount of computer time. It is essential, therefore, that the analysis and classifier design system be largely immune from user errors (e.g., control card errors), so that errors in the later stages of processing will not result in loss of all the work which has gone before.

3. In the face of the two requirements already noted, the experimental status of the remote sensing problem makes it desirable that most or all of the processing system be written in a high level compiler language so that modifications to the system may be made quickly and easily by the researcher.

The third of these requirements has been satisfied through use of FORTRAN IV (except for a few minor utility functions which can be accomplished most efficiently through use of assembly language). The way in which the other two requirements (user control, user-error recovery) have been met is discussed in the following sections.

3.2 LARSYSAA System Monitor; Free-form Card Format

Figure 2 shows the control structure of the LARSYSAA system. The figure indicates that the system is composed of a Monitor and four distinct processing phases, each processing phase directed by its own supervisor. The multiphase structure results largely from the need to minimize the amount of core memory occupied at any one time by program instructions, in order to maximize the amount of memory available for data. In fact, for the same reason the individual processors are also decomposed into multiple phases which are only called into core memory by the respective supervisors as needed.

Processing under LARSYSAA commences when the computer operating system recognizes a job control card calling for the LARSYSAA system and loads the LARSYSAA Monitor into core memory from the program library.

The principal responsibility of the LARSYSAA Monitor is to recognize and interpret Monitor Control Cards which request loading of the processor phases from the program library. These control cards and all other control cards and data cards read by LARSYSAA may have an almost arbitrary format. On control cards, a special key word must be punched first, followed by any other key words associated with the control card, separated by commas. On data cards, parameters are punched in a specific order, separated by at least one blank. Should the user inadvertently punch an unrecognizable keyword or inconsistent parameter, the card in error is printed on the console typewriter along with instructions as to the action necessary to resume processing. Diagnosis and correction of errors in this manner before they can produce abnormal termination of processing prevents the loss of intermediate results stored in core memory by preceding stages of processing. The researcher, who will generally operate the computer and punch control cards as they are needed (he may elect to type them in at the console if desired), is thus freed from the burden of remembering and adhering to rigid input format requirements and may concentrate more fully on the analysis and design problem at hand. The experienced user finds himself operating in a conversational mode with the computer program; the novice finds that the key-word, free-form card input and attendant error checking speeds the process of learning to use the system effectively.

3.3 The Processor Supervisors

The responsibilities of the Processor Supervisors are threefold: interpretation of processor control cards, dynamic memory allocation, and processor control. Once all of the processor control cards have been read (and any necessary error recovery performed), the processor scans the list of operations (processing options, see Table 1) that have been requested and calculates the amount of core memory required to perform those operations. If the core memory needed exceeds that available, the supervisor reports this fact to the user who may then reenter the processor control cards with appropriate changes. In pattern recognition terms, these changes generally involve trading off the number of processing options requested against the number of pattern features and/or pattern classes.

After a suitable set of processor control cards has been read, the supervisor calculates base addresses and sizes of all variable arrays so that core memory will be efficiently utilized. The size and address information is then available to the processor subroutines as needed.

The flexibility gained through the use of dynamic memory allocation broadens significantly the range of problems that can be handled, even in the face of fairly severe core memory limitations.

Once the memory allocation procedure is complete, the analysis and design operations requested are carried out under the direction of the supervisor.

3.4 The Processors

To discuss the processors in general terms, a possible analysis and design procedure will be described.

Given a new set of digitized scanner data and the associated ground truth, the researcher's first task is to select a set of training patterns. To accomplish this, he obtains a set of grey-level printouts which aids in locating the boundaries of the agricultural fields, roads, bodies of water, etc. Data from fields of known classification are then used by the Statistical Processor to calculate various statistical quantities for the pattern classes and produce the graphical data displays listed in the first part of Table 1.

Because of uncertainties as to ground truth details (such as the precise locations of field boundaries), the researcher may at this point have limited confidence in the training set he has selected. However, by using the graphical output of the Statistical Processor, he may select a set of pattern features which appear to be useful for differentiating between the classes and, temporarily bypassing the Feature Selection Processor, use the Classification Processor and Display Processor to produce a classification based on the tentative training set. It has been found that even such "crude" classification can yield a printout or map of the data which is considerably more detailed than a grey-level printout (probably because the information contained in several spectral channels is condensed into a single display). The researcher can use this result to refine the training set, perhaps then performing one or more reiterations of the same procedure to achieve additional refinement.

Once a reasonable amount of confidence in the training set has been attained, the Feature Selection Processor is brought into the processing loop. By means of a suitable algorithm (see, for example, Min, Landgrebe, and Fu[3]), the Feature Selection Processor provides information as to the best one, two, three, etc., features to be used to obtain optimal classification for the specific problem at hand. Using this information and additional passes through the Classification and Display Processors, the researcher's principal remaining task is to decide, on the basis of reasonable computation time and the desired level of classification reliability, which optimal feature set (i.e. how many features) should be used.

Figure 3 shows a grey-level printout and a classification result with an aerial photograph of the agricultural terrain over which the data were taken. The classification task was to discriminate wheat (printed with W's) from various other classes (all printed as blanks).

4. Program Modification Facilities

As noted in Section 3.1, the research environment requires that the analysis and classifier design program be easily modified - for example, to test new classification algorithms. This flexibility is achieved by a) the dynamic storage allocation approach discussed above; b) inter- and intra-program communication via common storage areas; c) residence of the source language program on a tape which is easily modified by an editing program; and d) a self-directed System Construction Program which, once initiated, performs all of the steps necessary to go from source language to operational progam. Item b increases considerably the importance of the System Construction Program which has the responsibility of inserting all COMMON cards into the source alnguage "deck" during system construction. This relieves the user of the chore of modifying the COMMON cards in every program and subroutine each time a change is made involving the common variables.

5. Concluding Remarks

This paper has presented some aspects of a system of computer programs which, in spite of a fairly complex processing situation and demanding core memory requirements, allows the user to solve a wide range of problems without actual modification of the program. When modifications are unavoidable, the system structure is such as to allow the changes to be implemented easily with the aid of a System Construction Program. A "conversational mode" of operation which is of particular value in the research environment has been achieved through the development of techniques which optimize man-machine communication and minimize the inefficiencies which usually result from a high level of on-line user-system interaction.

References

1. Laboratory for Agricultural Remote Sensing, Remote Sensing in Agriculture, Volume 2, Research Bulletin No. 832 (Annual Report), Agricultural Experiment Station, Purdue University, Lafayette, Indiana, 1967.

2. Laboratory for Agricultural Remote Sensing, Remote Sensing in Agriculture, Volume 3, to appear.

3. Min, P. J., Landgrebe, D. A., and Fu, K. S., "On Feature Selection in Multiclass Pattern Recognition," Proc. Second Annual Princeton Conference on Information Sciences and Systems, March 25 - 26, 1968, Princeton University, Princeton, N. J.

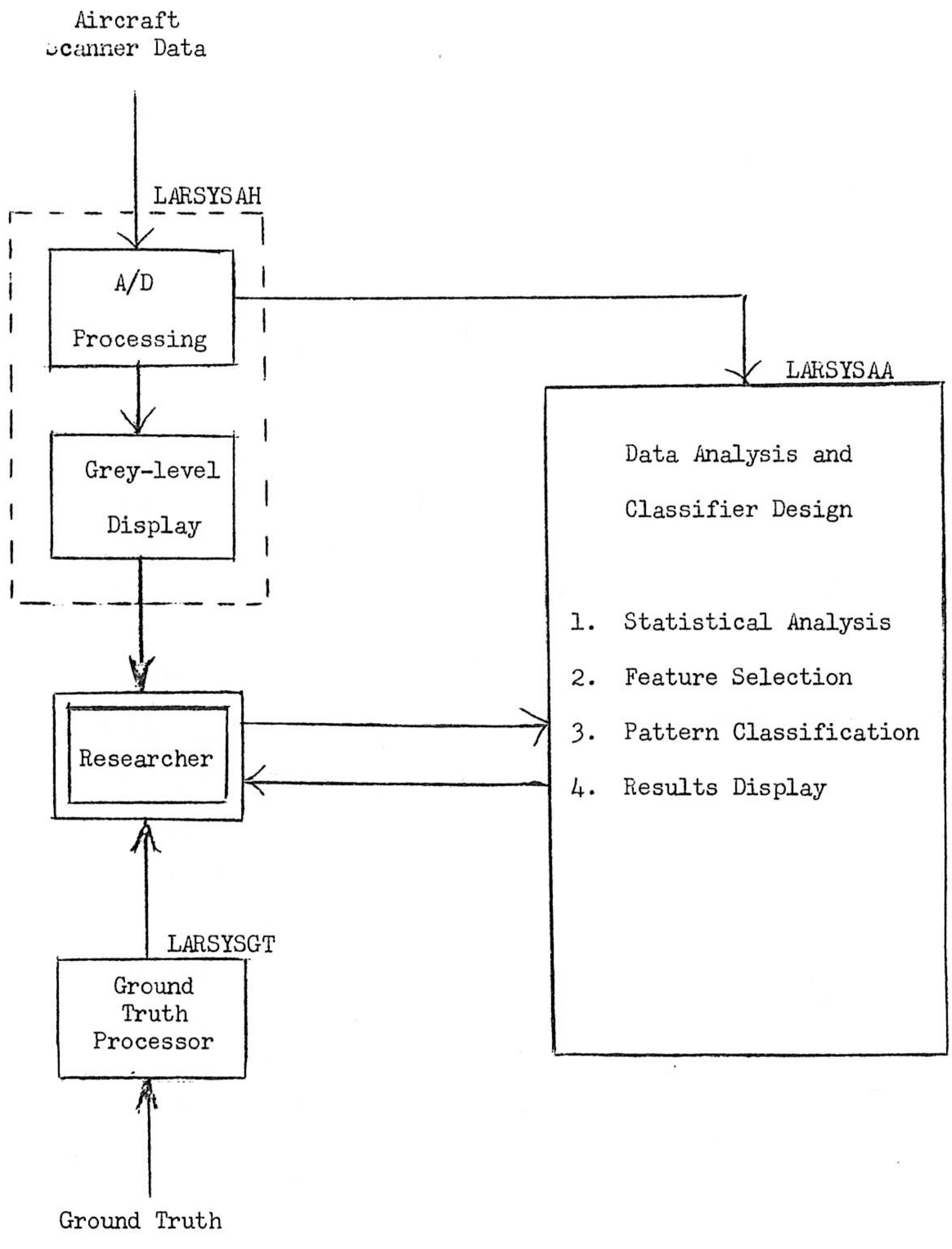


Figure 1. Laboratory for Agricultural Remote Sensing Data Processing System (LARSYS)

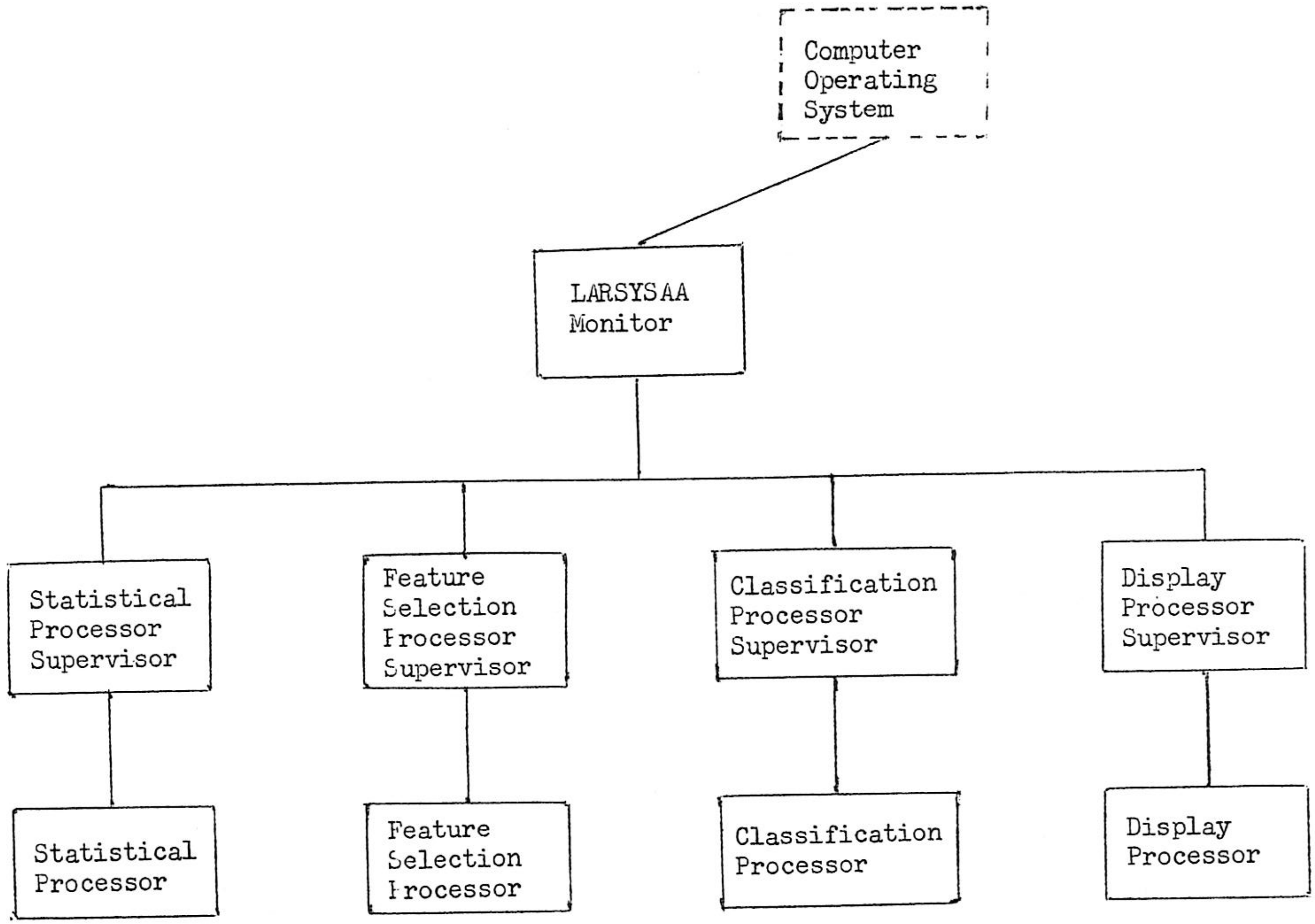
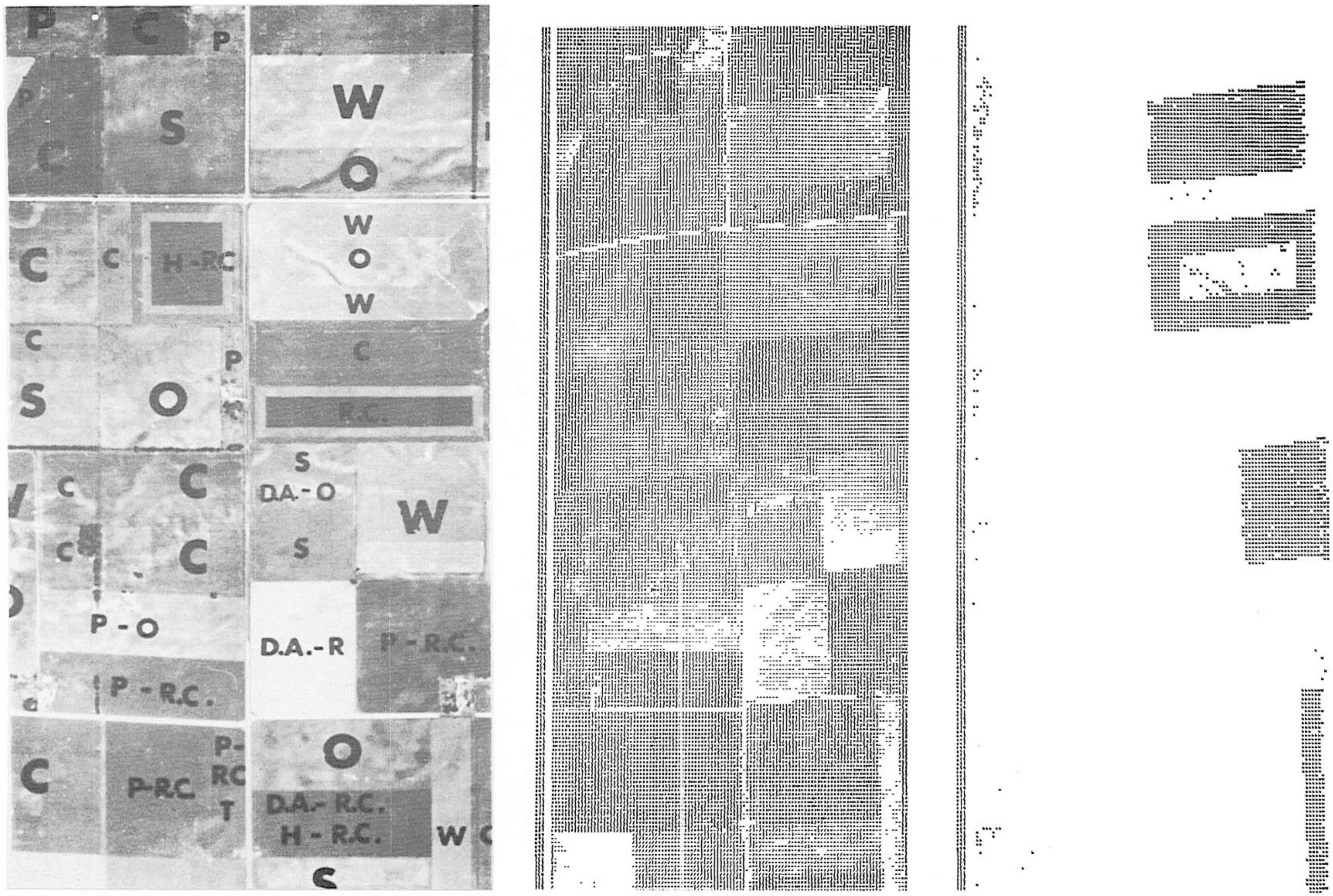


Figure 2. LARSYSAA Control Structure



(a) (b) (c)

Figure 3. (a) Aerial photo. (b) Grey-level display. (c) Classification results.

Statistical Analysis Facilities

Compute mean vector & covariance matrix for each class.

Compute mean vector & covariance matrix for each field.

Punch data deck containing statistics and other pertinent information for future use with Classification Processor.

Histogram selected features for each class.

Histogram selected features for each field.

Print spectral plots for each class.

Print spectral plots for each field.

Print as many spectral plots as desired, each displaying results for up to four different classes.

Feature Selection Facilities

Determine optimal sets of 1, 2, 3, ... features.

Classification Facilities

Perform pattern recognition using any subset of classes and features made available by the Statistical Processor.

Display Facilities

Print information as to source of training data.

Outline training sets if they appear in results display map.

Print results of training operations.

Use a specified symbol set for results display map.

Compute and print classifier performance evaluation for training set
- a) On per class basis.
- b) On per field basis.

List areas used as test samples for performance evaluation.

Outline on results map the areas used as test samples.

Compute and print classifier performance evaluation for test set
- a) On per class basis.
- b) On per field basis.

Apply liklihood thresholding to establish a rejection class.

Recompute and print performance evaluations on the basis of any specified grouping of classes.

Table 1: LARSYSAA Processing Facilities