SR-PO-00491

NAS9-15466

FINAL REPORT

CONTEXTUAL CLASSIFICATION OF MULTISPECTRAL IMAGE DATA: APPROXIMATE ALGORITHM

BY

J. C. Tilton

This report describes activity carried out in the Supporting Research Project.

PURDUE UNIVERSITY
Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907

August 1980

Star Information Form

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
SR-PO-00491		

4. Title and Subtitle	5. Report Date
CONTEXTUAL CLASSIFICATION OF MULTISPECTRAL IMAGE DATA: APPROXIMATE ALGORITHM	August 15, 1980
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No. 081580
9. Performing Organization Name and Address LABORATORY FOR APPLICATIONS OF REMOTE SENSING PURDUE UNIVERSITY 1220 Potter Drive WEST LAFAYETTE, IN 47907	10. Work Unit No.
	11. Contract or Grant No. NAS9-15466
12. Sponsoring Agency Name and Address NASA/JOHNSON SPACE CENTER HOUSTON, TX 77058	13. Type of Report and Period Covered
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Earlier reports have introduced a classification algorithm incorporating spatial context information in a general, statistical manner. Here an approximation to that algorithm is presented which is computationally less intensive, yet produces classifications that are nearly as accurate.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Classification algorithm, remote sensing, multispectral scanners	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
U	U		

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA - JSC

LARS Technical Report 081580

CONTEXTUAL CLASSIFICATION OF MULTISPECTRAL IMAGE DATA: APPROXIMATE ALGORITHM

James C. Tilton

School of Electrical Engineering
and
Laboratory for Applications of Remote Sensing

Purdue University
West Lafayette 47907, U.S.A.

ABSTRACT

Earlier reports[2,3] have introduced a classification algorithm incorporating spatial context information in a general, statistical manner. Here an approximation to that algorithm is presented which is computationally less intensive, yet produces classifications that are nearly as accurate.

I. INTRODUCTION

The most widely used method for classifying remotely sensed data from such sources as multispectral scanners on aircraft or satellite platforms is a point-by-point classification technique in which data from each pixel in the scene are classified individually and and independently[1]. The information normally used by this classifier is only spectral or, in some cases, spectral and temporal. There is generally no provision for using spatial information.

In contrast, when scanner data are displayed in image form, a human analyst routinely uses spatial context to help decide what is in the imagery. Using context, he or she may be able to easily pick out roads, delineate boundaries of agricultural fields, and differentiate between grass in an urban setting

(lawns) and grass in an agricultural setting (pasture or forage crops) where a point-by-point classifier would have much difficulty in doing so.

Earlier reports[2,3] describe the development of a statistical classification algorithm which incorporates spatial context information in a general manner. This algorithm exploits the tendency alluded to above of certain ground-cover classes to be more likely to occur in some contexts than in others.

This contextual classification algorithm is very computationally intensive, typically requiring a large amount of computer time. To reduce execution time, one could exploit the latest improvements in the raw speed of computer components and/or one could take advantage of special computer architectures involving multiple processing elements[2,4]. An alternative tactic discussed here is to look for a less computationally intensive algorithm which approximates the original contextual classification algorithm. If such an algorithm produces classifications that do not differ significantly in accuracy from the original algorithm, the approximate algorithm would be the preferred algorithm in practical applications using conventional (serial) computers.

II. ORIGINAL ALGORITHM

In order to fully discuss the approximate algorithm, a brief description of the original algorithm must be given. For a detailed derivation of the decision function used in the original algorithm, see [2].

Consistent with the general characteristics of imaging systems for remote sensing, we assume a two-dimensional array of $N=N_1\times N_2$ random observations X_{ij} having fixed but unknown classifications ϑ_{ij}, as shown in Figure 1. The observation X_{ij} consists of n measurements (usually containing spectral and/or temporal information), while the classification ϑ_{ij} can be any one of m spectral or

information classes from the set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$.

$$\begin{array}{|cccc|} \hline \vartheta_{11} & \vartheta_{12} & \cdots & \vartheta_{1N_2} \\ \vartheta_{21} & \vartheta_{22} & \cdots & \vartheta_{2N_2} \\ \vdots & & & \\ \vartheta_{N_1 1} & & \cdots & \vartheta_{N_1 N_2} \\ \hline \end{array}$$

Figure 1. A two-dimensional array of $N = N_1 \times N_2$ pixels.

Let $\underline{X}$ denote a vector whose components are the ordered observations:

$$\underline{X} = [X_{ij} \mid i = 1,2,\ldots,N_1; j = 1,2,\ldots,N_2]^T.$$

Similarly, let $\underline{\vartheta}$ be the vector of states (true classifications) associated with the observations in $\underline{X}$:

$$\underline{\vartheta} = [\vartheta_{ij} \mid i = 1,2,\ldots,N_1; j = 1,2,\ldots,N_2]^T.$$

Let the action (classification) taken with respect to pixel (i,j) be denoted by $a_{ij} \varepsilon \Omega$. We restrict the decision function $a_{ij}(\cdot)$ to depend only on an arbitrary but fixed subset of p observations in $\underline{X}$. This subset includes, along with X_{ij}, p-1 observations spatially near to, but not necessarily adjacent to, X_{ij}. These p-1 observations serve as the spatial context for X_{ij} and are taken from the same spatial positions relative to pixel position (i,j) for all i and j. Call this arrangement of pixels together with X_{ij} the p-context array, several examples of which are shown in Figure 2. Group the p observations in the p-context array into a vector of observations $\underline{X}_{ij}$ and let $\underline{\vartheta}_{ij}$ be the vector of true but unknown

classifications associated with the observations in $\underline{X}_{ij}$. Let $\underline{\vartheta}^p \varepsilon \Omega^p$ stand for p-vectors of classes. Each component of $\underline{\vartheta}^p$ is a variable which can take on any classification value. Note that $\underline{\vartheta}_{ij}$ is the particular instance of $\underline{\vartheta}^p$ associated with pixel position (i,j). Correspondence of the components of $\underline{\vartheta}_{ij}$, $\underline{X}_{ij}$ and $\underline{\vartheta}^p$ to the positions in the p-context array is fixed but arbitrary except that the pixel to be classified will always correspond to the p^{th} component.

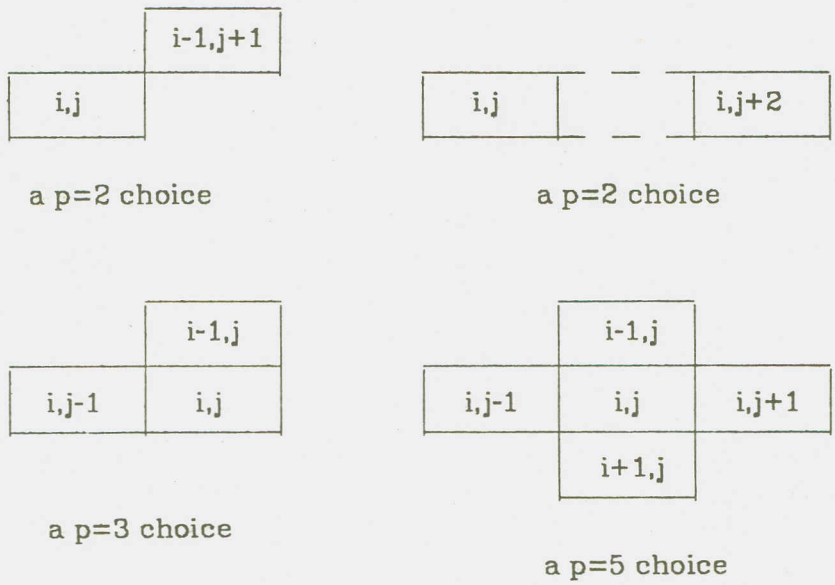

Figure 2. Examples of p-context arrays.

Our optimal decision rule now has the form

$$a_{ij}(\underline{X}) = d(\underline{X}_{ij}) \tag{1}$$

for a fixed function $d(\cdot)$ mapping p-vectors of observations to actions. In deriving the explicit decision function we assume that the distribution of $\underline{X}$ is such that every $\underline{X}_{ij}$ for which $\underline{\vartheta}_{ij}=\underline{\vartheta}^p$ has the same marginal density, i.e. the marginal densities depend only on the measurement values in $\underline{X}_{ij}$ and the set of

classifications in $\underline{\vartheta}_{ij}$ and not the location (i,j). Under this assumption the marginal density becomes

$$f_{ij}(\cdot \mid \underline{\vartheta}_{ij} = \underline{\vartheta}^p) = f(\cdot \mid \underline{\vartheta}^p). \tag{2}$$

Utilizing a "0-1 loss function", the decision rule becomes:

$d(\underline{X}_{ij})$ = the action a which maximizes

$$\sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} C(\underline{\vartheta}^p) f(\underline{X}_{ij} \mid \underline{\vartheta}^p) \tag{3}$$

where $C(\underline{\vartheta}^p)$, the *context distribution*, is the relative frequency with which $\underline{\vartheta}^p$ occurs in the array $\underline{\vartheta}$.

One way to satisfy the assumption resulting in the relation expressed by equation (2) is to assume class-conditional independence for $\underline{X}$. In this case, the marginal density becomes

$$f(\underline{X}_{ij} \mid \underline{\vartheta}^p) = \prod_{k=1}^{p} f(X_k \mid \vartheta_k) \tag{4}$$

where X_k and ϑ_k are the k^{th} elements of $\underline{X}_{ij}$ and $\underline{\vartheta}^p$, respectively. There may be other densities for $\underline{X}$ with the necessary property, but it is not apparent how one could construct a useful density without making possibly inconsistent assumptions. Invoking the class-conditional independence assumption, the decision rule in equation (3) becomes:

$d(\underline{X}_{ij})$ = the action a which maximizes

$$\sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} C(\underline{\vartheta}^p) \prod_{k=1}^{p} f(X_k \mid \vartheta_k). \tag{5}$$

The optimal choice of $d(\cdot)$ cannot be implemented in practice since it depends on $G(\underline{\vartheta}^p)$ and the $f(X_k \mid \vartheta_k)$ which are unknown. Methods for estimating the $f(X_k \mid \vartheta_k)$ are well established from considerable experience in using the conventional no-context maximum likelihood decision rule[1]. Methods for estimating $G(\underline{\vartheta}^p)$ from the $\underline{X}_{ij}$ and the effectiveness of these estimations are discussed in the earlier reports[2,3], and are the subject of ongoing research.

III. APPROXIMATE ALGORITHM

To come up with a reasonable approximate algorithm, one must examine the computer implementation of the original decision function*. Consider the case where the set Ω is defined over spectral classes and the class-conditional independence assumption is taken. For this case it is reasonable to assume the densities $f(X_k \mid \vartheta_k)$ in equation (5) to be multivariate normal with mean vector M_{ϑ_k} and covariance matrix Σ_{ϑ_k} giving

$$f(X_k \mid \vartheta_k) = \left[\frac{1}{2\pi}\right]^{\frac{n}{2}} |\Sigma_{\vartheta_k}|^{-\frac{1}{2}} \exp\left[-\tfrac{1}{2}(X_k - M_{\vartheta_k})^T \Sigma_{\vartheta_k}^{-1}(X_k - M_{\vartheta_k})\right]$$

where n is the dimensionality of the observation X_k (see [1] for the rationale behind this assumption in the no-context case). Using the multivariate normal assumption, equation (5) becomes

$$d(\underline{X}_{ij}) = \text{the action } \alpha \text{ which maximizes } d_\alpha(\underline{X}_{ij})$$

where

---

* For this study, the algorithm was implemented on a PDP-11/45 computer in the programming language "C". Test runs were also made on a PDP-11/70 computer.

$$d_a(\underline{X}_{ij}) = \sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} G(\underline{\vartheta}^p) \prod_{k=1}^{p} \left[\frac{1}{2\pi}\right]^{\frac{np}{2}} |\Sigma_{\vartheta_k}|^{-\frac{1}{2}} \exp\left[-\tfrac{1}{2}(X_k - M_{\vartheta_k})^T \Sigma_{\vartheta_k}^{-1}(X_k - M_{\vartheta_k})\right]$$

Let $d_a'(\underline{X}_{ij}) = \ln[d_a(\underline{X}_{ij}) \times (2\pi)^{\frac{pn}{2}}]$. Maximizing $d_a'(\underline{X}_{ij})$ is equivalent to maximizing $d_a(\underline{X}_{ij})$. Letting $Q_{\vartheta_k}(X_k) = (X_k - M_{\vartheta_k})^T \Sigma_{\vartheta_k}^{-1}(X_k - M_{\vartheta_k})$, we have

$$d_a'(\underline{X}_{ij}) = \ln\left[\sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} G(\underline{\vartheta}^p) \prod_{k=1}^{p} |\Sigma_{\vartheta_k}|^{-\frac{1}{2}} \exp[-\tfrac{1}{2} Q_{\vartheta_k}(X_k)]\right]$$

$$= \ln\left[\sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} \exp\left[\ln G(\underline{\vartheta}^p) - \tfrac{1}{2}\sum_{k=1}^{p}\left[\ln|\Sigma_{\vartheta_k}| + Q_{\vartheta_k}(X_k)\right]\right]\right]$$

$$= \ln\left[\sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} \exp[F(\underline{X}_{ij}, \underline{\vartheta}^p)]\right]$$

where

$$F(\underline{X}_{ij}, \underline{\vartheta}^p) \triangleq \ln G(\underline{\vartheta}^p) - \tfrac{1}{2}\sum_{k=1}^{p}\left[\ln|\Sigma_{\vartheta_k}| + Q_{\vartheta_k}(X_k)\right]$$

In the simulated and real data sets studied (see the earlier reports[2,3]), the term $\exp[F(\underline{X}_{ij}, \underline{\vartheta}^p)]$ ranges over a larger negative exponential range than available on the PDP-11/45 (an exponential range of $10^{\pm 37}$ is available). To circumvent this problem it was necessary to use the following procedure.

Let

$$M_a(\underline{X}_{ij}) \triangleq \max_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} F(\underline{X}_{ij}, \underline{\vartheta}^p)$$

and rewrite $d_a'(\underline{X}_{ij})$ as follows:

$$d_a'(\underline{X}_{ij}) = \ln\left[\exp[M_a(\underline{X}_{ij})] \sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} \exp[F(\underline{X}_{ij}, \underline{\vartheta}^p) - M_a(\underline{X}_{ij})]\right]$$

$$= M_a(\underline{X}_{ij}) + \ln\left[\sum_{\substack{\underline{\vartheta}^p \in \Omega^p, \\ \vartheta_p = a}} \exp[F(\underline{X}_{ij}, \underline{\vartheta}^p) - M_a(\underline{X}_{ij})]\right]. \qquad (6)$$

Calculating $d_a'(\underline{X}_{ij})$ in this way insures that at least one term of the sum does not cause underflow because the exponent of the maximum term, $M_a(\underline{X}_{ij})$, is never taken. This procedure also makes it less likely that other terms in the sum will cause underflow (the $F(\underline{X}_{ij}, \underline{\vartheta}^p)$ tend to be large *negative* numbers).

In checking out this particular implementation of the decision function, it was noted that $M_a(\underline{X}_{ij})$ was in most cases significantly larger than the logarithmic term in equation (6). This observation suggested the following approximation of the decision function first given in equation (5):

$$d(\underline{X}_{ij}) = \text{the action } a \text{ which maximizes } M_a(\underline{X}_{ij}), \qquad (7a)$$

or in the notation of equation (5):

$$d(\underline{X}_{ij}) = \text{the action } a \text{ which maximizes for all } \underline{\vartheta}^p \in \Omega^p \text{ with } \vartheta_p = a$$

$$C(\underline{\vartheta}^p) \prod_{k=1}^{p} f(X_k \mid \vartheta_k). \qquad (7b)$$

Comparing equations (6) and (7a) one can see that the implementation of equation (7a) requires less computation and storage than equation (6). In equation (7a), the logarithmic term in equation (6) need not be calculated and the individual values of $F(\underline{X}_{ij}, \underline{\vartheta}^p)$ for a particular action a need not be stored; only the maximum value is needed. We would expect, then, that this approximate

algorithm will take less computation time than the original algorithm for any data set. The effect of the approximation on classification accuracy, however, may be data dependent.

IV. EXPERIMENTAL RESULTS

It now remains to be tested empirically whether the lower computational and storage requirements of the approximate algorithm result in a significant savings in computer costs when compared to the original algorithm, and whether the classifications produced by the approximate algorithm differ significantly from the classifications produced by the original algorithm.

The approximate algorithm was compared for accuracy with the original algorithm in tests using the simulated data set and the real data sets described in [2]. (The real data sets will be subsequentially referred to as the LACIE and Bloomington data sets.) Included in the comparisons were algorithms that take only the three or five maximum terms in the summation in equation (5). These additional algorithms serve to give an indication of how many terms in the summation are needed to produce classifications equivalent to those produced by the original algorithm. The results of this study are summarized in Table 1. The context distribution for the simulated data set test was estimated by tabulation from the template classification from which the simulated data set was generated and the context distribution for the LACIE data set was tabulated from the first 25 lines of a ground-truth-guided no-context classification* as described in [2]. Both data sets were evaluated over the entire 50-pixel square area. The

* A ground-truth guided classification is performed just like the usual no-context classification except that the classifier is restricted to selecting spectral classes from the information class indicated by the ground truth data.

context distribution for the Bloomington data set was tabulated from entire 50-pixel square area of a ground-truth-guided no-context classification. Since the Bloomington data set has only 1317 ground-truth pixels, the ground-truth-guided classification was allowed to degenerate to the usual unguided no-context classification over the remaining 1183 pixels. The Bloomington data set was evaluated over the 1317 ground-truth pixels. Eight-nearest-neighbor context was used in all cases.

Table 1

PERFORMANCE OF APPROXIMATION ALGORITHM IN TERMS OF ACCURACY
Context distribution estimated from ground-truth-guided class'n.

Data Set	Overall Accuracy, %			
	Orig. Alg., Eq. (5)	5 Largest Terms of Sum in Eq. (5)	3 Largest Terms of Sum in Eq. (5)	Approx. Alg., Eq. (7a&b)
Simulated	96.84	96.88	97.04	97.04
LACIE	87.52	87.52	87.52	87.47
Bloomington	95.60	95.60	95.52	95.52

As can be seen in Table 1, the approximate algorithm performed very well in terms of overall accuracy when compared to the original algorithm. The table also shows that in the two real data sets, the five largest terms of the sum in equation (5) are all that are needed to produce identical classifications to those produced by the full sum (the original algorithm).

The accuracy of the approximate algorithm was also tested in two cases where the "Power Method" was used in estimating the context distribution (see [3] for a description of the Power Method). Table 2 displays the classification accuracies resulting from applying the Power Method to the Bloomington data

set as described in [3]. Also displayed are the accuracy results from applying the Power Method in a similar manner to the LACIE data set.

Table 2

PERFORMANCE OF APPROXIMATION ALGORITHM IN TERMS OF ACCURACY
Context distribution estimated using Power Method.

Data Set	Overall Accuracy, %	
	Original Algorithm, Equation (5)	Approximate Algorithm, Equation (7a&b)
Bloomington	88.46	88.38
LACIE	86.70	86.66

Again the approximate algorithm produced overall accuracies that were very close to those produced by the original algorithm. To put these minor accuracy differences in proper perspective, it helps to note that a conventional uniform-priors no-context classifier produced overall accuracies of 83.07% on the Bloomington data set and 78.73% on the LACIE data set.

The approximate algorithm was compared in terms of computer timings with the original algorithm on the simulated data set and the two real Landsat data sets. Highly optimized versions of each algorithm (written in the "C" programming language) were run on PDP-11/45 and PDP-11/70 computers. Also compared to these two algorithms was a highly optimized version of the original algorithm that simply allowed the underflows to occur rather than attempting to circumvent the underflows. This version allows comparison of the approximate algorithm to a simulated implementation of the original algorithm on a computer with adequate exponential range.

Table 3

PERFORMANCE OF APPROXIMATION ALGORITHM IN TERMS OF TIMINGS
(50-pixel square LACIE data set, two-nearest-neighbor context,
480 nonzero elements in context distribution, PDP-11/45 computer)

Classifier	Time in Seconds	
	Real+	User+
Original Algorithm with underflow protection	2993	2636
Original Algorithm without underflow protection	2498	2388
Approximation Algorithm	1247	1185

The length of time the classifier took to process the 50-pixel square data sets varied depending primarily on the number of nonzero elements of the context distribution. (The number of terms that need to be evaluated in the sum in equation (5) and the number of terms to be compared in the maximization of equation (7b) are equal to the number of nonzero elements in the context distribution.) The ratio of timings between the three programs remained fairly consistent, however, across all data sets. Tables 3 and 4 display typical quiet system* timings on a PDP-11/45 computer for cases of few nonzero elements of the context distribution (480) and relatively large number of nonzero elements (2193). Table 5 gives the timings for the case displayed in Table 4, but run on a PDP-11/70 computer.

+ Real time is the time the program is running in the computer including the time the program is swapped out for other tasks. User time is essentially time spent doing computations.

* The runs were made during early morning hours when few other tasks were being performed by the computer.

Table 4

PERFORMANCE OF APPROXIMATION ALGORITHM IN TERMS OF TIMINGS
(50-pixel square simulated data set, two-nearest-neighbor context,
2193 nonzero elements in context distribution, PDP-11/45 computer)

Classifier	Time in Seconds	
	Real	User
Original Algorithm with underflow protection	18596	14702
Original Algorithm without underflow protection	15064	14290
Approximation Algorithm	9079	8675

Table 5

PERFORMANCE OF APPROXIMATION ALGORITHM IN TERMS OF TIMINGS
(50-pixel square simulated data set, two-nearest-neighbor context,
2193 nonzero elements in context distribution, PDP-11/70 computer)

Classifier	Time in Seconds	
	Real	User
Original Algorithm with underflow protection	7240	5832
Original Algorithm without underflow protection	6830	6573
Approximation Algorithm	2747	2526

The three tables show that the approximate algorithm averaged less than half the real and user time taken by either of the other two algorithms. This amounts to a significant improvement in computation time.

V. CONCLUDING REMARKS

The contextual classification algorithm developed in [2] is very computationally intensive, typically requiring a large amount of computer time. An approximation to this algorithm has been explored in this report. Experimental results from one simulated and two real data sets show that on these data sets the approximate algorithm takes significantly less computer time while producing classifications that do not differ significantly in accuracy from classifications produced by the original algorithm.

By the nature of the approximate algorithm, it is expected that similar time savings will occur when the approximate algorithm is used on other data sets. Whether or not the accuracy results presented here can be expected with other data sets depends on the extent to which the data sets tested here are representative of remotely sensed data in general. We expect that they are fairly representative. Further tests are planned to confirm that the approximation does not significantly affect classification accuracy.

VII. REFERENCES

[1] P. H. Swain and S. M. Davis, eds., *Remote Sensing: The Quantitative Approach*, McGraw-Hill, New York, 1978.

[2] P. H. Swain, P. E. Anuta, D. A. Landgrebe, and H. J. Siegel, *Vol. III: Processing Techniques Development, Part 2: Data Processing and Information Extraction Techniques*, LARS Contract Report 113079, Laboratory for Applications of Remote Sensing (LARS), Purdue University, West Lafayette, IN., November 1979.

[3] J. C. Tilton, P. H. Swain, and S. B. Vardeman, "Context Distribution Estimation for Contextual Classification of Multispectral Image Data," *Proceedings of the 1980 Machine Processing of Remotely Sensed Data Symposium* (IEEE Catalog No. 80 CH 1533-9 MPRSD), pp. 171-180, June 1980.

[4] H. J. Siegel, P. H. Swain, and B. W. Smith, "Parallel Processing Implementations of a Contextual Classifier for Multispectral Remote Sensing Data," *Proceedings of the 1980 Machine Processing of Remotely Sensed Data Symposium* (IEEE Catalog No. 80 CH 1533-9 MPRSD), pp. 19-29, June 1980.