1-1-1977

Estimating Costs and Performance of Systems for Machine Processing of Remotely Sensed Data

Richard J. Ballard

Lester F. Eastwood

Reprinted from

Symposium on

Machine Processing of

Remotely Sensed Data

June 21 - 23, 1977

The Laboratory for Applications of
Remote Sensing

Purdue University
West Lafayette
Indiana

IEEE Catalog No.
77CH1218-7 MPRSD

ESTIMATING COSTS AND PERFORMANCE OF SYSTEMS FOR MACHINE PROCESSING OF REMOTELY SENSED DATA

RICHARD J. BALLARD AND LESTER F. EASTWOOD, JR.
Washington University

This paper outlines a method for estimating computer processing times and costs incurred in producing information products from digital remotely sensed data. The method accounts for both computation and overhead, and it may be applied to any serial computer. We apply the method to estimate the cost and computer time involved in producing Level II Land Use and Vegetative Cover Maps for a five-state, mid-western region. Our results show that the amount of data to be processed overloads some example computer systems, but that the processing is feasible on others.

I. INTRODUCTION

Because individual state agencies typically lack the resources for machine processing of satellite remote sensing data, satellite data might be of widest benefit to states if it were processed at a shared processing facility. Our research team has studied the feasibility of such a facility[1] by identifying twenty-seven remote sensing-based information products of wide utility to state agencies (in Illinois, Iowa, Minnesota, Missouri, and Wisconsin); determining from a user survey a useful coverage area, update frequency, number of satellite-derivable classes contained, and scale for each product; and estimating the costs and performance of one, regional processing center producing these products.

This paper concentrates on one element of the cost and performance analysis. We present a theoretical method for analyzing computer processing times and costs for information products based on digital, remotely sensed data. The method is based on determining the amount of computation required by typical remote sensing data processing algorithms. It determines processing times and costs as functions of image data parameters (number of pixels, and bands per pixel) and processing variables (number of classification classes, and iterations required to achieve acceptable accuracy).

The method combines two, independent estimation techniques. The first technique estimates processing times on an IBM 360/67 by employing simple interpolation of results observed by a past user of LARSYS.[2] This method is accurate, in that it takes account of all computation tasks, including system overhead. However, it is inflexible, because it applies only to four-band data and to the IBM 360/67. The second technique determines computation times and costs theoretically by calculating computational loads put on any serial computer by a full range of image processing algorithms. By contrast with the first scheme, it can be applied to any serial computer. However, because it fails to account for "overhead" (e.g. running the computer's operating system), it is inaccurate when used alone. Combining the two techniques allows us to take overhead into account, as the first scheme does, while retaining the second technique's flexibility.

The next section presents the first of these two techniques. Section III outlines the basis for our "theoretical" method of estimating single algorithms, computation times, and costs. Total computer times and costs for producing an information product from digital remote sensing data are determined from the product's algorithmic processing sequence in Section IV. We estimate processing times and costs on several computer systems for two products, Vegetative Cover Maps and Level II Land Use Maps. We also estimate the processing required to produce these products annually over our five-state region in the quantities desired by state agencies.

II. ESTIMATION USING OBSERVED LARSYS TIMES AND COSTS

A past LARSYS user has supplied us with tables of costs he incurred in producing Level II Land Use Maps.[2] Table 1 lists these costs. Factors affecting these costs include the number of pixels processed, the number of classes into which data are classified, and the per CPU minute processing cost of the LARSYS computer.

In 1973, when the costs listed in Table 1 were incurred, the LARSYS per CPU minute charge was $6.00, while as of May, 1976, it was $4.83.[2,3]

As it is, Table 1 can be used to estimate both LARSYS processing costs (based on the old $6.00 per CPU minute charge) and processing times in CPU minutes. We estimate processing times from the costs of Table 1 by assuming that the "per run" costs of Table 1 represent input/output and other special overhead, while the "per million pixel" costs represent CPU processing costs. We estimate CPU times by dividing the "per million pixel" costs by $6.00 per CPU minute, the charge upon which Table 1's cost equations are based.

Table 1. LARSYS Processing Costs for LANDSAT Data.

Algorithm	Processing Cost*
LANDSAT/LARSYS Reformat	$ 65 + 8 (MP)**
Geometric Correction (Linear Nearest Neighbor)	$125 + 525 (MP)
Overlay	$600 + 1500 (MP)
Clustering (approximate)	$500+
Max. Likelihood Classification	
30 classes	$868 (MP)
40 classes	$1157 (MP)
50 classes	$1445 (MP)

*The LARSYS costs presented in this table were charged for processing done in December 1973.[2] The costs are not official figures issued by LARS.

**(MP) = per million pixels of four-band data.

+For clustering training sets of 11,000 pixels.

Extrapolating to any other per CPU minute charge is simple if we assume that fixed costs listed in Table 1 remain unchanged. The total cost of an algorithm is then its fixed cost plus the product of the number of CPU minutes it consumes and the new per CPU minute charge. For example, geometric correction of an entire LANDSAT image at the old $6.00/CPU minute rate cost $125 + $525 (7.56) or $4100. Under our assumptions, the processing time required is $525 (7.56/$6.00) or 660 CPU minutes. Thus, if the new processing charge is $4.83/CPU minute, processing the same data would cost $125 + 660 ($4.83), or $3300.

III. ANALYTIC ESTIMATION OF PROCESSING TIMES AND COSTS

An alternative to this extrapolative method of estimating algorithm processing times and costs is to determine the amount of computation (that is, the number of adds, multiplies, etc.) required to perform each algorithm. These computational estimates can then be used to estimate algorithms' required processing times and costs on any serial computer system.

A. ESTIMATING ALGORITHM COMPUTATION REQUIREMENTS

The first analysis step is to develop a functional description (e.g., a flowchart) illustrating each algorithm's processing sequence. From these descriptions, we estimate the computational requirements of each algorithm as functions of the number of bands per pixel, the number of pixels to be processed, and other image data parameters.

To illustrate how computational requirements are derived, consider the algorithm performing maximum likelihood classification. The maximum likelihood (ML) algorithm computes a measure of the likelihood that an observed pixel value comes from a particular object class. The pixel is assigned to the class for which this measure is greatest.

For Gaussian-distributed data (a common remote-sensing assumption), the likelihood measure that a pixel $\underline{X}$ represents class k is given by

$$L_k(\underline{X}) = \ln(p(k)) - 1/2 \ln|\underline{C}_k| - 1/2(\underline{X} - \underline{M}_k)^T \underline{C}_k^{-1}(\underline{X} - \underline{M}_k) \quad (1)$$

where p(k) is the probability of object class k, $\underline{M}_k$ is the mean vector associated with object class k, $\underline{C}_k$ is the kth class' covariance matrix, and $|\underline{C}_k|$ denotes the determinant of this matrix.

Equation (1) reduces to

$$L_k(\underline{X}) = f(k) - 1/2(\underline{X} - \underline{M}_k)^T \underline{C}_k^{-1}(\underline{X} - \underline{M}_k) \quad (2)$$

where $f(k) = \ln p(k) - 1/2 \ln |\underline{C}_k|$, is a known quantity, for class k, k = 1, 2, ..., C. Given the observed brightness values $\underline{X}$, the algorithm computes $L_k(\underline{X})$ for each of C object classes and assigns the pixel to the class having the largest value of $L_k(\underline{X})$.

After generating a functional description of the algorithm, each algorithm step is analyzed to determine both its computational requirements and the number of times the step is executed per algorithm run. For example, one step in the ML algorithm might be written $\underline{X}_2(k) = \underline{C}_k^{-1} \underline{X}_1(k)$. $\underline{C}_k^{-1}$ is a B x B element matrix, where B is the number of data bands being processed; $\underline{X}_1(k)$ is the B element vector representing the difference

$(X - \underline{M}_k)$. Computing the product $\underline{X}_2(k)$ requires B^2 multiplies and B(B-1) additions. The step $\underline{X}_2(k) = \underline{C}_k^{-1} \underline{X}_1(k)$ is executed C times for each pixel. Therefore, when classifying N_p pixels into one of C classes, the step $\underline{X}_2(k) = \underline{C}_k^{-1} \underline{X}_1(k)$ contributes CN_pB^2 multiplies and $CN_pB(B-1)$ additions to the algorithm's computational requirements. After analyzing each algorithm step, the algorithm's total computational requirements are found by summing each step's computational requirements.

B. ESTIMATING ALGORITHM PROCESSING TIMES

Using published figures[4,5] listing computer instruction execution times, we determine the time needed to accomplish the required numbers of each instruction. For example, if an algorithm requires 10^6 add operations to process a given amount of data, a computer which takes 5.4 μsec to fetch data and execute an add instruction will require 5.4 seconds to perform the adds. The total estimated CPU time to perform each algorithm is then the sum of the times needed to perform each algorithm's required operations.

As a check on our algorithm time estimates, we estimate algorithm processing times on LARSYS. For example, we estimate that to process an entire frame of LANDSAT imagery (7.56 million pixels) into thirty classes using ML classification would require 1030 CPU minutes on the IBM 360/67. Similarly we estimate that to geometrically correct an entire LANDSAT frame using linear transformation and nearest neighbor resampling would require eleven CPU minutes on the IBM 360/67.

We then compare these estimates with the LARSYS processing times implied by the cost figures of Table 1. For example, the cost of processing LANDSAT data into thirty classes using ML classification is listed as $868 per million pixels. At the old $6.00 per CPU minute cost, this corresponds to a per-LANDSAT-image processing time of 1100 CPU minutes. The cost of geometrically correcting LANDSAT data is listed in Table 1 as $125 per run plus $525 per million pixels. Assuming that the $125 per run charge represents special overhead and does not represent CPU processing charges, this corresponds to a per-LANDSAT-image correction time of 660 CPU minutes.

Our analytic estimates of LARSYS processing times are always lower than estimates derived from observed LARSYS costs. This is understandable; the algorithm functional descriptions on which our estimates are based do not account for the computer's operating system overhead.

Both methods of estimating algorithm run times have faults. Algorithm time estimates based on the costs of Table 1 apply only to four-band data and to the IBM 360/67, while our analytic estimates neglect overhead. We seek to combine the strengths of each method by scaling our analytic estimates to include overhead. The combined estimation method is:

1) Develop an analytic estimate of the number of each type of computer operation to perform a given algorithm.

2) Determine the time required to perform these operations on the IBM 360/67.

3) Determine the total time required to perform the algorithm on the IBM 360/67 based upon the observed costs of Table 1.

4) Compute the algorithm's overhead multiplier by dividing the algorithm processing time found in 3) by the algorithm processing time estimated in 2).

Our scaled estimates of algorithm computational requirements are listed in Table 2. These computational requirements account for overhead and may be used to estimate algorithm processing times on any serial computer.

Table 2. Algorithm Computational Requirements

Task	# Moves (Memory to Memory)	# Adds
Reformat CCTS	$6.7[2BN_p]$	-
Geometric Correction	$59[N_p(4+2B)]$	$59[4N_p]$
Cluster Analysis	$46[ICN_p]$	$46[BI(C+3)N_p]$
ML Classification	$1.1[CN_p]$	$1.1[C(B^2+B+1)CN_p]$

Task	# Multiplies	# Compares
Reformat CCTS	—	—
Geometric Correction	$59[4N_p]$	$59[2N_p]$
Cluster Analysis	$46[BI(C+1)N_p]$	$46[I(C-1)N_p]$
ML Classification	$1.1[(B^2+B+1)CN_p]$	$1.1[(C-1)N_p]$

Notes: a) B = # of bands (4 for current LANDSAT)
b) N_p = # of pixels (7.56×10^6 for current LANDSAT imagery).
c) C = # of object classes or clusters.
d) I = # of iterations (see text).
e) — = implies negligible operation count.

The overlay algorithm was not analyzed and is not listed in Table 2. To estimate this algorithm's run time on computer's other than the IBM 360/67, we define a speed factor for computer X by

$$SF(X) = \frac{\sum \text{Est. algorithm run times on computer X}}{\sum \text{Est. algorithm run times on the 360/67}} \quad (3)$$

Then, for example, the overlay algorithm's processing time on another computer is the product of the computer's speed factor and the overlay algorithm's 360/67 processing time. Based on the computations of Section IV, the speed factors for the Univac 1108 and CDC 7600 are .39 and .033 respectively.

C. ESTIMATING ALGORITHM PROCESSING COSTS

We estimate each computer's cost per CPU minute by assuming that the monthly cost of operating a computing facility is equal to twice the computer's monthly lease cost (a reasonable assumption in costing computing facilities, made to allow for salaries of operating personnel and for maintenance) and that 140 CPU hours of operation are realized monthly. Under these assumptions, the cost per CPU minute is given by

$$\text{Cost per CPU minute} = \frac{2(\text{computer leased cost/mo})}{140 \text{ hrs/mo}} \times \frac{\text{hr}}{60 \text{ min}} \quad (4)$$

As a check on the validity of (4), we estimate per CPU minute processing charges for the IBM 360/67 computer used in LARSYS. The 360/67 has a monthly lease cost of $23,000,[5] giving an estimated per CPU minute processing charge of $5.48. This compares well with LARSYS processing charges of $6.00 per CPU minute and $4.83 per CPU[2,3] minute charged in 1973 and 1976, respectively.

IV. ESTIMATING PRODUCT PROCESSING TIMES AND COSTS

To estimate information product processing times and costs, we first determine the algorithmic sequence required to produce the product. In addition to specifying the algorithms to be used, the sequence specifies the number of required iterations for each algorithm. Some algorithms, such as reformatting, need to run only once. Cluster analysis, on the other hand, is an iterative process and we can estimate the iterations required as a function of the number of clusters sought. Other non-iterative algorithms, such as maximum likelihood classification, must be by run multiple times to correct errors indicated by available ground truth data.

Our studies of the information needs of state agencies in our five state midwestern region indicate that the majority of needed satellite-derivable data is contained in two information products: Level-II Land Use Maps and Vegetative Cover Maps.[1]

In this section, we specify example algorithmic sequences for these products, and use the combined method of Section III to estimate product processing times and costs on three different computers. We also estimate each computer's annual processing time and cost to produce these products in the quantities desired by state agencies.

These satellite-derived products will be more useful to state agencies in their day-to-day activities when the thirty meter spatial resolution imagery of the proposed LANDSAT Follow-On Mission[6] becomes available. Thus we use Follow-On parameters in calculating computation costs. Assuming that a frame of Follow-On imagery covers the same area as a frame of current LANDSAT imagery, a frame of Follow-On imagery would contain 53.6 million pixels.

A. VEGETATIVE COVER MAPS

Based on past experience,[7] we estimate that fifteen classes of vegetation are satellite-derivable; two additional classes, water and other, can account for the map's non-vegetated areas.

Experience has also shown that the current LANDSAT's bands 5 and 7 provide the most useful vegetative data, and that both spring and summer imagery must be included to achieve sufficient classification accuracy. Thus, we assume that after merging two equivalent bands of Follow-On data from spring and summer imagery, the resulting four band imagery will allow vegetative cover to be adequately identified. To reduce processing costs, only the two data bands used per image will actually be geometrically corrected.

Spectral signature estimates for fifteen vegetative cover classes and for water must be derived by cluster analysis of selected "training areas." We assume the training areas comprise 11,000 pixels of the four-band merged data. Our own experience indicates that an average of sixteen clustering iterations will be required to estimate spectral signatures.

The merged data must be classified into seventeen classes using maximum likelihood (ML) classification. In fact, however, only sixteen object classes must be tested; the seventeenth, or "other," class would be chosen only if none of the other sixteen classes are likely. Between runs of the ML algorithm, analysts would compare interpreted imagery with known ground truth to locate classification errors, and would modify spectral signature estimates to correct these errors. We estimate an average of four ML runs would be required per product.

The algorithmic sequence for producing Vegetative Cover Maps is: 1) Reformat spring and summer imagery; 2) Geometrically correct each image; 3) Overlay spring and summer imagery into one, four-band composite image; 4) Cluster analyze portions of the composite image, sixteen iterations required; 5) ML classify the composite imagery into sixteen classes plus "other", four runs required.

Using the combined method of Section III, we estimate the CPU time required to process a frame of Follow-On data into a Vegetative Cover Map on three computers: the IBM 360/67, the Univac 1108, and the CDC 7600. Using (4), we estimate processing costs for each computer. To process the data on the IBM 360/67 would require 662 CPU hours and would cost $218,000. 265 CPU hours would be required to process the data on the Univac 1108, and the processing cost would be $170,000. Finally, to process the data on the CDC 7600 would require 25.6 CPU hours; the processing cost would be $30,000.

B. LEVEL II LAND USE MAPS

Level II Land Use Maps display - by definition - thirty-seven classes.[8] Of these twenty-eight are relevant in our five-state region. However, only sixteen of the twenty-eight classes are non-vegetative classes, and the twelve vegetative classes are displayed in at least as much detail on the Vegetative Cover Maps discussed previously. As a result, if Vegetative Cover Maps are being produced concurrently, to produce Level-II Land Use Maps require only that we process the areas classified "other" in Vegetative Cover Maps into the sixteen non-vegetative land use classes.

The algorithmic sequence needed to produce the nonvegetative sections of Level II Land Use Maps is similar to the sequence used to produce Vegetative Cover Maps with two exceptions. First, Level II Land Use Maps require winter imagery to delineate urban and "built-up" land classes. Therefore, only one raw image must be reformatted and no merging of imagery is required. In addition, classification information is not concentrated in two spectral bands; we assume the best four of Follow-On's imagery bands will be used to produce land use maps.

The algorithmic sequence to produce Level-II Land Use Maps is: 1) Reformat the raw imagery; 2) Geometrically correct the raw imagery; 3) Cluster analyze portions of the image into sixteen nonvegetative classes, sixteen iterations required; 4) ML classify appropriate areas of the image into sixteen non-vegetative land use classes, four runs required.

For ease of comparison, we estimate processing times and costs assuming that an entire Follow-On image is to be processed. Savings due to processing only nonvegetative areas are considered in the next section.

To process an entire Follow-On image into sixteen nonvegetative land use classes would require 369 CPU hours on the IBM 360/67; the processing cost would be $121,000. 137 CPU hours would be required to perform the processing on the Univac 1108, and the corresponding processing cost is $88,000. Finally, to process the image using the CDC 7600 would require 14.3 CPU hours and would cost $17,000.

C. REGIONAL FACILITY PROCESSING TIMES AND COSTS

A number of factors affect the annual computational load of a regional processing facility. The first factor is the area covered by each product. In this example, we assume the facility produces two products, Vegetative Cover Maps and Level II Land Use Maps. Vegetative Cover Maps (including the ubiquitous "other" class) must be produced over the entire region. Data for Level II Land Use Maps, on the other hand, only have to be processed over non-vegetative areas; vegetative land use classes are taken from the Vegetative Cover Maps. Based on Missouri land cover statistics, we estimate ten percent of the five state region must be processed by non-vegetative land use classes.

A second factor affecting computational load is each product's update frequency. Our analysis of state agency needs indicates that Vegetative Cover Maps must be produced for the entire region annually. Level II Land Use Maps, on the other hand, must be updated only every five years. Thus, a regional center must produce land use maps for one-fifth of the five state region annually.

Two additional factors affecting computational load, the number and type of classes each product contains and the seasonal imagery each product requires, have already been discussed.

One remaining consideration is the acquisition of cloud-free imagery. EROS statistics show that twenty-five percent of 901 LANDSAT images taken over sample areas in each of the five states had ten percent cloud cover or less. Probability of cloud cover showed no strong seasonal dependence. A single Follow-On Mission is therefore likely to provide the coverage required in winter, spring, and summer without excessive mosaicing to produce "cloud-free" imagery. If orbital overlap and edge effects are included, forty-five LANDSAT images are required to cover the five-state region.

This information allows us to specify a total satellite input data for the regional center. The amount of processing required is determined by the products' algorithmic sequences, coverage areas, and update frequencies. To produce Vegetative Cover Maps and Level II Land Use Maps over the five-state region, the following algorithms must be performed annually:

Reformat 99 images.
Geometrically correct 2 bands on 90 images.
Geometrically correct 4 bands on 9 images.
Overlay 2 bands on 45 pairs of images.
Cluster Analyze portions of 54 images into 16 classes, 4 runs per image.
ML Classify 45 images into 16 classes, 4 runs per image.
ML Classify 10 percent of 9 images into 16 classes, 4 runs per image.

To calculate the CPU time and cost for this processing, we assume that partitioning data requires negligible processing time, e.g., the time to ML classify ten percent of nine images is equivalent to the time to process ninety percent of one image. Using the method of Section III, we calculate the center's annual processing times and costs. To perform the center's annual processing on the IBM 360/67 would require 31,000 CPU hours and would cost $10,000,000. The Univac 1108 requires 12,000 hours to perform the required processing at a cost of $7,760,000. Finally, 1200 CPU hours are required to perform the required processing on the CDC 7600. The annual processing cost using the CDC 7600 would be $1,440,000.

Each computer's required annual processing time indicates whether the computer is suitable for use at the center. Our estimates of cost per CPU minute are based on 140 CPU hours of processing per month, implying that 1700 processing hours are available annually. Of the three computers considered, only the CDC 7600 can process the center's products in the time allotted.

V. CONCLUSIONS

We have outlined a method to estimate computer processing times and costs for information products based upon digital remotely sensed data. The method accounts for image data and processing parameters. Furthermore, it accounts for operating system overhead and may be applied to any serial computer system.

By analyzing the computational load required to produce a given menu of information products, the products' required processing times and costs on a particular computer system may be estimated. This indicates how heavily a particular computer system will be utilized in product production, and whether the system will be overloaded. Thus, the method of this paper could be of great utility in designing an appropriate processing facility for a given menu of information products.

VI. ACKNOWLEDGEMENTS

This work was supported by the National Aeronautics and Space Administration under Contract No. NAS5-20680. The authors would like to acknowledge helpful discussions with Profs. J. K. Gohagan, C. T. Hill, and R. P. Morgan and with G. G. Crnkovich and T. R. Hays of the EODMS staff. In addition, the cooperation of Mr. Leonard Gaydos of the United States Geological Survey is greatly appreciated.

VII. REFERENCES

1. Eastwood, L. F., et al., Final Report - Program on Earth Observation Data Management Systems (EODMS), Center for Development Technology, Washington University, St. Louis, Missouri, December, 1976.

2. Gaydos, L., Personal interview, United States Geological Survey Geography Program, Ames Research Center, Moffett Field, California, May, 1976.

3. Phillips, T., Telephone interview, LARS, Purdue University, West Lafayette, Indiana, May, 1976.

4. Van Vleck, E. M., et al., Earth Resources Ground Data Handling Systems for the 1980's NASA TMX-62,240, NASA-Ames Research Center, Moffett Field, California, March, 1973, p. 113.

5. "Characteristics of Digital Computers," Computers and People, Vol. 24, No. 10B, October 31, 1975, pp. 95-118.

6. Barker, J., Personal interview, NASA Goddard Space Flight Center, Greenbelt, Maryland, June, 1976.

7. Williams, D. L., Personal interview, Department of Geography, University of Illinois, Champaign-Urbana, Illinois, June, 1976.

8. Anderson, J. R. et al., "A Land Use and Land Cover Classification System for Use with Remote Sensor Data," U.S. Geological Survey Professional Paper #964, U.S. Government Printing Office, Washington, D.C., 1976.