หอสมุดกลาง จุฬาลงกรณมหาวิทยาลัย

บรรณานุกรม

บุญชนะ กลั่นคำสอน และ ธงชัย จารุพพัฒน์. "การสำรวจและศึกษาสภาพความเปลี่ยนแปลงความเปลี่ยนแปลงของพื้นที่ป่าไม้ในท้องที่ภาคตะวันออก ด้วยภาพถ่ายดาวเทียม." กรุงเทพมหานคร: กรมป่าไม้, ธันวาคม 2520.

Duda, R. O., and Hart, P. E. Pattern Recognition and Scene Analysis. New York: John Willey and Sons, 1973.

Morrison, D. F. Multivariate Statistical Method. New York: McGraw Hill, 1967.

Prasit Prapinmongkolkarn, and Chavalit Thisyakorn. "Feasibility Study of System Impremention for the Digital Processing of LANDSAT Data." Final Report Submitted to Thailand National Remote Sensing Program as Partial Fulfillment of a Research Contract Granted by TNRSP, August 1977.

Prasit Prapinmongkolkarn, Nimitra Kattiyakulwanich, and Chavalit Thisyakorn. "Data Processing Capability in Thailand." Admitted to be Published in Engineering Journal of Thailand.

Swain, Philip H., and Lindenlaub, Yohn C. "Pattern Recognition: A Basis for Remote Sensing Data Analysis." In Remote Sensing Technology and Application, Course Note on July 31 - August 11. Indiana: Purdue University, 1972.

Thomas, Valerie L. "Generation and Physical Characteristics of the ERTS MSS System Corrected Computer Compatible Tapes." Maryland: Goddard Space Flight Center, July 1973.

Tongchat Hongladaromp, and Nimitra Kattiyakulwanich. "Computer Processing of Remote Sensing Data." AIT Final Report Submitted to the Commettee for Coordination of Investigative of the Lower Mekong Basin, March 1977.

Wagner, Harvey L. "Feasibility Study of an Interactive System for Processing Digital Multispectral Scanner Data." Master's Tehsis in Remote Sensing, University of Michigan, 1976.

ภาคผนวก ก.

การคำนวณการกระจายข้อมูล

สูตรโดยทั่วไปของ Multivariate Normal Density มีดังนี้คือ

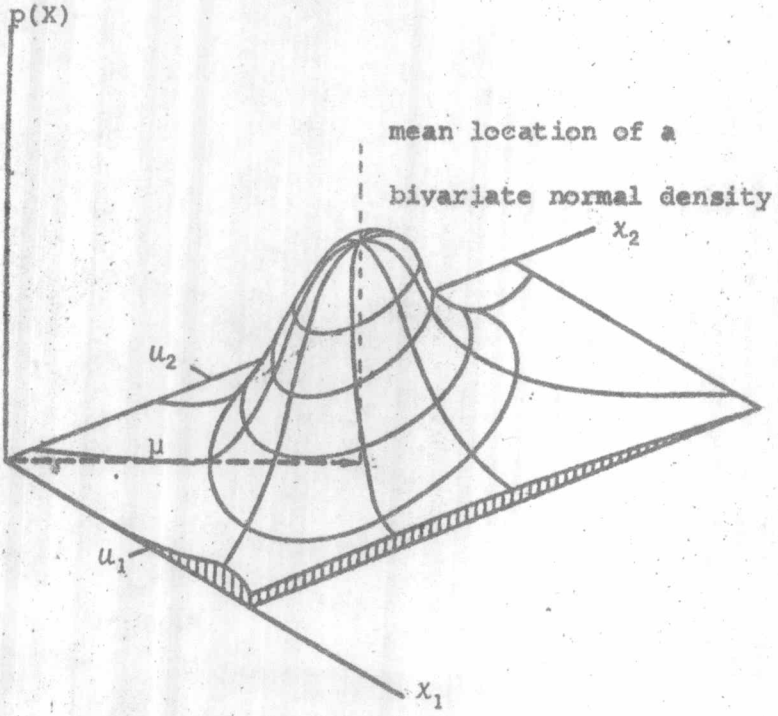
$$p(X) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{\frac{1}{2}}} \exp\ -\frac{1}{2}(X-\mu)^{t}\Sigma^{-1}(X-\mu) \qquad (1)$$

จะมี μ และ Σ เป็น parameters

โดยที่ $\mu = \begin{bmatrix} u_1 \\ u_2 \\ . \\ . \\ . \\ u_d \end{bmatrix}$ เป็น d-component mean vector

และ $\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12}\cdots\sigma_{1d} \\ \sigma_{21} & \sigma_{22}\cdots\sigma_{2d} \\ . & . \quad . \\ . & . \quad . \\ . & . \quad . \\ \sigma_{d1} & \sigma_{d2}\cdots\sigma_{dd} \end{bmatrix}$ เป็น $d \times d$ covariance matrix

การกระจายข้อมูลจะรวมกันเป็นกลุ่มโดยมีจุดศูนย์กลางอยู่ที่ mean vector และมีรูปร่างไปตามค่า covariance matrix (ถ้าเป็น Bivariate Normal Density จะมีลักษณะเป็นระฆังคว่ำ ดังรูป)



BIVARIATE NORMAL DENSITY

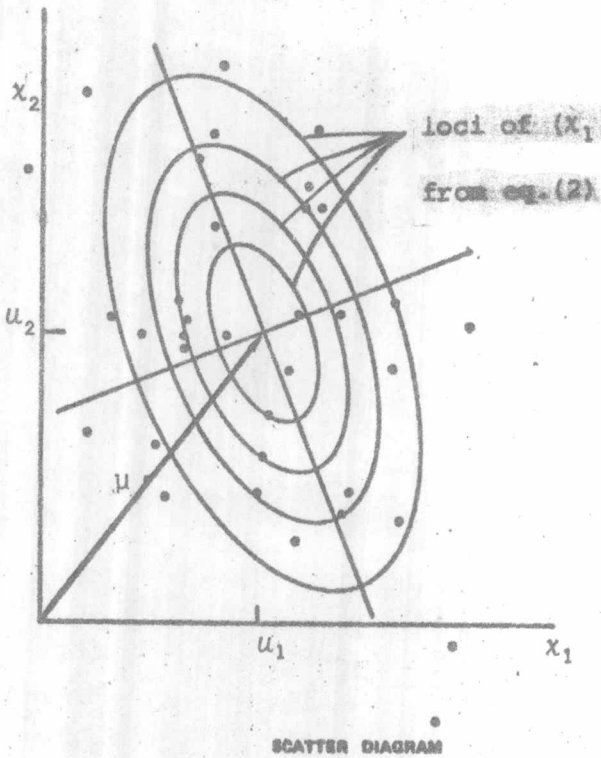
ในการดูการกระจายข้อมูลสามารถพิจารณาได้จาก

$$(X-\mu)^t \Sigma^{-1} (X-\mu) = c \qquad (2)$$

ซึ่งเป็นสมการกำลังสองที่มีเส้นรูปร่าง (contour) เป็น hyperellipsoids โดยที่ c คือค่าคงที่ที่สามารถกำหนดได้ตามต้องการ เช่น กำหนดจากหนึ่ง standard deviation ดังนั้นถ้า X เป็นตัวแปรใด ๆ จะกำหนดค่า c ได้

โดยที่ $\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_d \end{bmatrix} - \begin{bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ u_d \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \cdot \\ \cdot \\ \sigma_d \end{bmatrix}$ เมื่อ $\sigma_i^2 = \sigma_{ii}$; i=1,...,d

สำหรับการคำนวณการกระจายข้อมูลของ training class ในวิทยานิพนธ์นี้ ตัวแปรต่าง ๆ จะมาจาก 2-dimensional หรือเป็น Bivariate Normal Density ซึ่งจะมีเส้นรูปร่าง (contour) เป็น ellipsoids ดังรูป (จะเป็น plane ตัดขวาง normal surface ในรูปที่แล้ว เมื่อกำหนดให้ $c = c_1$)



SCATTER DIAGRAM

<u>ตัวอย่าง</u> เพื่อให้เป็นที่เข้าใจยิ่งขึ้นจะขอยกตัวอย่างการคำนวณโดยใช้ค่าสถิติต่าง ๆ ของ training class หนึ่ง เช่น Dense Forest (D-FRS) ของ ปี 2516 ซึ่งมีค่าสถิติดังนี้

$$\mu = \begin{bmatrix} 11.00000 \\ 17.33333 \end{bmatrix} \quad , \quad \Sigma = \begin{bmatrix} 0.74699 & 0.54217 \\ 0.54217 & 1.91165 \end{bmatrix}$$

$$\Sigma^{-1} = \begin{bmatrix} 1.68571 & -0.47809 \\ -0.47809 & 0.65870 \end{bmatrix}$$

<u>ในขั้นแรก</u> เราจะคำนวณค่า c ในสมการ (2) เสียก่อนโดย

กำหนดให้ $X - \mu = \begin{bmatrix} 0.865 \\ 1.383 \end{bmatrix}$ ซึ่งเป็นค่าหนึ่ง standard deviation ของข้อมูลข้างต้นนี้

แทนค่าในสมการ (2)

ดังนั้น $c = \begin{bmatrix} 0.865 & 1.383 \end{bmatrix} \begin{bmatrix} 1.68571 & -0.47809 \\ -0.47809 & 0.65870 \end{bmatrix} \begin{bmatrix} 0.865 \\ 1.383 \end{bmatrix} \triangleq c_1$

ได้ $c_1 = 1.383$

<u>ในขั้นต่อไป</u>

สมมุติให้ $X - \mu = \begin{bmatrix} a \\ b \end{bmatrix}$

แล้วแทนค่า $X - \mu = \begin{bmatrix} a \\ b \end{bmatrix}$ และ $c = c_1$ กลับไปในสมการ (2)

ดังนั้น $\begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} 1.68571 & -0.47809 \\ -0.47809 & 0.65870 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = c = 1.38$

ซึ่งเราจะแก้สมการนี้หาความสัมพันธ์ของ a และ b ได้ดังนี้

$$b = \frac{\pm\sqrt{1.38 - 1.3387a^2 + .5891a}}{.8116} \qquad (3)$$

<u>ในขั้นสุดท้าย</u>

เราจะสามารถหาค่า X จากสมการ (3) นี้ได้โดย

- สมมุติค่าต่าง ๆ ของ a
- คำนวณค่า b
- คำนวณ $X = \mu + \begin{bmatrix} a \\ b \end{bmatrix}$

ภาคผนวก ข.

ประวัติ LIGMALS Package

ความเป็นมาของโปรแกรมลิกมัลส์ในประเทศไทย

ลิกมัลส์เป็นชื่อของโปรแกรมชุดหนึ่ง (LIGMALS - LANDSAT Interactive Gray Map And Level Slicing Systems) ซึ่งเริ่มต้นเขียนโดย E. L. Wagner ในผลงานวิจัยระดับปริญญาโท ที่ The University of Michigan เมื่อปี 1976 โดยมีจุดประสงค์ขั้นต้นเพื่อออกแบบให้ใช้ได้กับเครื่องคอมพิวเตอร์ขนาดเล็ก ในลักษณะ interactive กับข่ายคอมพิวเตอร์ของมหาวิทยาลัย (MTS - Michigan Terminal System) ซึ่งประกอบด้วยเครื่องคอมพิวเตอร์รุ่น AMDAHL 470/V6

โปรแกรมชุดนี้ ผู้ใช้เพียงแต่ป้อนข้อมูลที่ต้องการและทำการเรียก subroutines ต่าง ๆ ให้ทำงาน ซึ่งประกอบด้วย Subroutines ที่สำคัญดังต่อไปนี้ REFORM, MAIN, RSET/DTAP, MLSET, ALSET, PMAP, CMAP, SVID, LVID, HSTGRM, DMSTRS และ RATIO นอกจากนี้ โปรแกรมลิกมัลส์ยังเรียกใช้ utility routines ที่มีใช้อยู่ในระบบ MTS อีกด้วยเช่น FREAD, FLKLTR, FINCMD, GUINFO, IGGTRL, IGBGNS, IGDA, IGMR และอื่น ๆ อีกมาก ซึ่ง routines เหล่านี้ ช่วยประกอบการทำงานของโปรแกรมลิกมัลส์ และช่วยให้งานพัฒนาของโปรแกรมดังกล่าวง่ายขึ้นมาก

ต่อมา เมื่อโปรแกรมชุดนี้ได้รับความนิยมนำไปใช้กันอย่างแพร่หลายมากขึ้น สำหรับหน่วยงานที่ไม่อยู่ในข่ายของระบบ MTS ก็ย่อมไม่สามารถนำโปรแกรมเหล่านี้ไปใช้ได้โดยตรง E. L. Wagner จึงได้ประยุกต์โปรแกรมชุดนี้จาก Interactive Mode มาเป็น Batch Mode เรียกว่า LIGMALS/B ซึ่งเขียนในรูปของภาษา FORTRAN IV มากยิ่งขึ้น และใช้งานได้ดีกับเครื่องคอมพิวเตอร์ขนาดเล็กหรือขนาดกลางทั่ว ๆ ไป LIGMALS/B ประกอบด้วยโปรแกรมต่าง ๆ เหมือนกับลิกมัลส์รุ่นแรก ยกเว้นไม่มี Subroutine MAIN, CMAP, LVID

และ SVID นอกจากนี้ยังได้เพิ่มโปรแกรม CNTR ขึ้นมาอีกด้วยเพื่อนับจำนวนจุดของข้อมูลที่ผ่านระบวนการทำงานโดยการกำหนดของโปรแกรม MLSET หรือ ALSET

ปัจจุบัน H. L. Wagner ทำงานประจำอยู่ที่ ERIM (Environmertal Research Institute of Michigan) และโดยการติดต่อผ่านคุณสุวิทย์ วิบูลย์เศรษฐ์ ผู้ประสานงานโครงการสำรวจทรัพยากรธรรมชาติด้วยดาวเทียม สภาวิจัยแห่งชาติ ผศ. ดร.ประสิทธ์ ประพิณมงคลการ แห่งจุฬาลงกรณ์มหาวิทยาลัย ร่วมกับ ดร.ชวลิต ทิสยากร จากบริษัท ศูนย์คอมพิวเตอร์แห่งประเทศไทยจำกัด ได้ทำการศึกษาโปรแกรมลิกมัลส์ชุดนี้ขึ้นเป็นครั้งแรกในประเทศไทย โดยทดลองใช้กับเครื่องคอมพิวเตอร์ขนาดเล็กรุ่น B1710 ที่บริษัท ศูนย์คอมพิวเตอร์แห่งประเทศไทย และเสนอเป็นรายงานการวิจัย ต่อสภาวิจัยแห่งชาติ เมื่อปี 1977

รายละเอียดพอสังเขปของแต่ละโปรแกรม มีดังต่อไปนี้

Phase I

REFORM

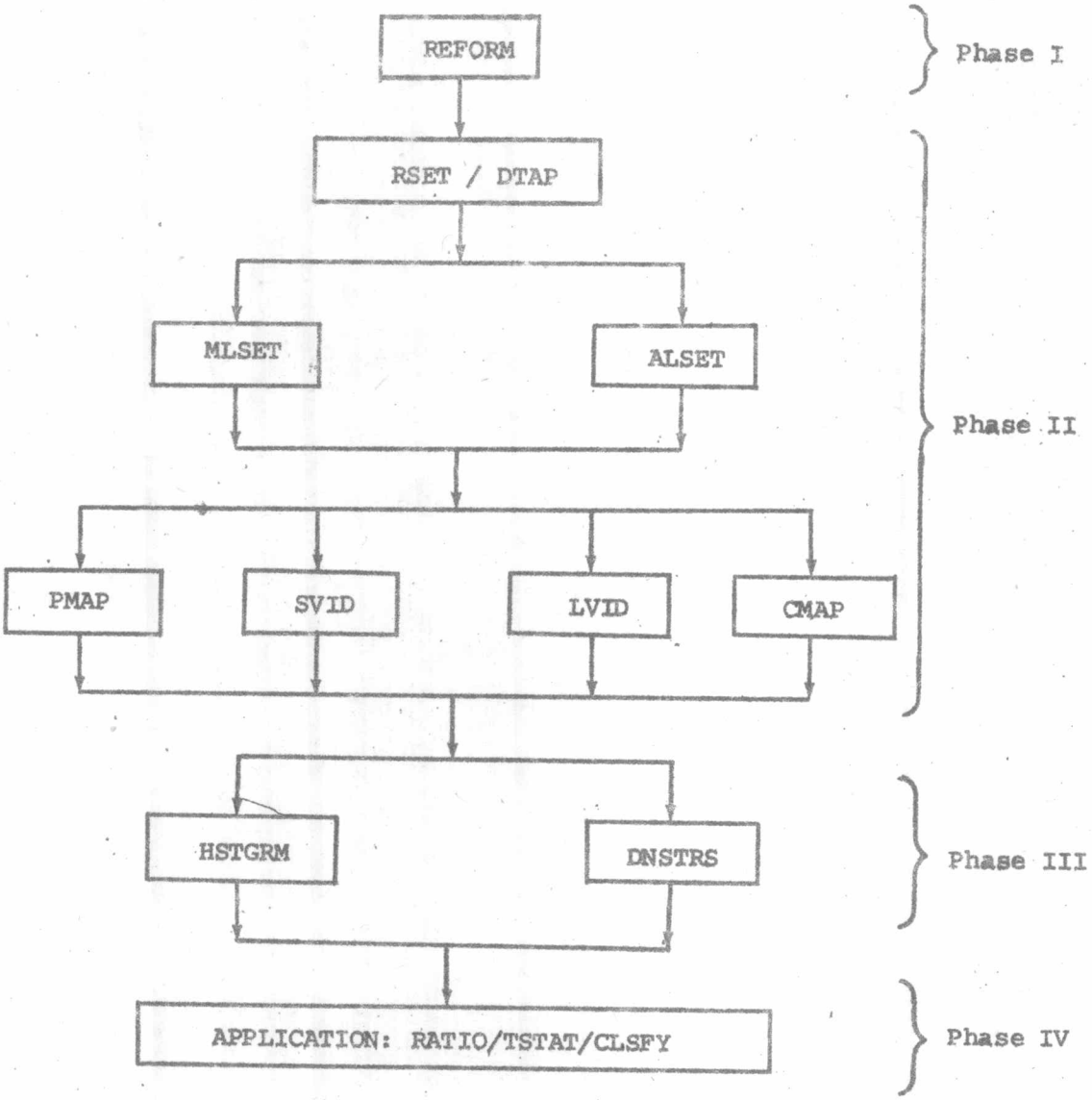
จากข้อมูลในเทปกระทัดรัด (CCT) ซึ่งจัดทำโดยองค์การการบินและอวกาศแห่งชาติ (NASA) ของสหรัฐอเมริกา โปรแกรมนี้จะอ่านและจัดลำดับข้อมูลใหม่ ให้อยู่ในฟอร์แมทของลิกมัลส์ โดยที่ฟอร์แมทเดิมของ CCT เป็นแบบ $A_1A_2B_1B_2C_1C_2D_1D_2A_3A_4B_3\ldots$ ตัวละ 1 Byte และฟอร์แมทของลิกมัลส์เป็นแบบ $A_1B_1C_1D_1A_2B_2C_2D_2A_3B_3C_3\ldots$ ตัวละ 1 Word ซึ่งทำให้ง่ายต่อการเรียกใช้ข้อมูลในโอกาสต่อไป

Phase II

RSET และ DTAP

เป็นโปรแกรมสำหรับเรียกเอาข้อมูลในส่วนที่ต้องการใช้งาน โดยที่ผู้ใช้จะต้องบอกรายละเอียดเกี่ยวกับจำนวน เส้น (Scan Line) ที่ต้องการจากเส้นไหนถึงเส้นไหน และจากจุด (Pixel) ไหนถึงจุดไหน เป็นข้อมูลสำหรับ RSET จากนั้น DTAP จะทำการอ่านและบันทึกข้อมูลที่ต้องการ ลงในหน่วยความจำหน่วยอื่น เช่น จานแม่เหล็กหรือเทปม้วนใหม่ก็ได้

โครงสร้างของโปรแกรมลิกนัลส์



Phase I : จัดเรียงข้อมูลจากฟอร์แมทของ CCT ให้เป็นฟอร์แมทของลิกนัลส์

Phase II : แยกข้อมูลเฉพาะพื้นที่ที่สนใจในช่วงคลื่นใด ๆ ตามต้องการเพื่อแสดงออกมาเป็นภาพพิมพ์ขาวดำ หรือ แสดงออกทางจอโทรภาพ

Phase III : แสดงคุณสมบัติทางสถิติของข้อมูลและความแตกต่างของค่าของข้อมูล (ความเข้ม) ออกมาเป็นกราฟ เฉพาะช่วงคลื่นที่ต้องการ

Phase IV : เป็นส่วนของการนำข้อมูลไปวิเคราะห์ ประยุกต์กับงานวิจัยต่าง ๆ

MLSET และ ALSET

ใช้ข้อมูลจาก DTAP จุดประสงค์เพื่อจัดระดับความเข้มของข้อมูลในบริเวณที่ต้องการให้คอมพิวเตอร์พิมพ์เป็นภาพขาวดำ (Gray Map) ออกเป็น 8 ระดับ (Level Slice) โดยอัตโนมัติ (ALSET) ซึ่งจะนำข้อมูลทั้งหมดมาคำนวณค่าเฉลี่ย ค่าเบี่ยงเบนมาตราฐานเพื่อแบ่งจำนวนข้อมูลในแต่ละพิสัยให้มีจำนวนเท่า ๆ กัน หรือมากน้อยแล้วแต่ความต้องการของผู้ใช้โดยสามารถกำหนดน้ำหนักของค่าคงที่ซึ่งใช้คูณกับค่าเบี่ยงเบนมาตราฐาน หรืออาจจะกำหนดค่าโดยตรงจากผู้ใช้เอง (MLSET) ซึ่งอาจทำ Histogram Plot (HSTGRM) และ Densitometry Plot (DNSTRS) มาช่วยในการกำหนดด้วยก็ได้

PMAP/CMAP

ใช้ระดับความเข้มทั้ง 8 ค่าที่ได้จากโปรแกรม MLSET หรือ ALSET โดยจัดพิสัย (range) ความเข้มออกเป็น 7 พิสัย และกำหนดตัวอักษรแทนแต่ละพิสัยให้จากนั้นจะทำการ process เอาข้อมูลในส่วนนั้นให้เครื่องพิมพ์ (Line Printer) ทำการพิมพ์ตัวอักษรที่แทนค่าระดับความเข้มของจุดนั้น ๆ ออกมาโดยที่ PMAP พิมพ์ตัวอักษรด้วยหมึกดำ ส่วน CMAP ใช้กับเครื่องพิมพ์ที่สามารถเปลี่ยนสีผ้าหมึกได้

SVID/LVID

ลักษณะการทำงานคล้ายคลึงกับ PMAP ทุกประการ เพียงแต่ให้ภาพออกมาทางจอโทรภาพ โดยที่ SVID ให้ภาพมาตราส่วน 1 : 24,000 ส่วน LVID ให้ภาพมาตราส่วน 1 : 50,000

Phase III

HSTGRM และ DNSTRS

เป็นโปรแกรมที่ให้รายละเอียดด้านสถิติและอื่น ๆ ของข้อมูล กล่าวคือ HSTGRM แจกแจงความถี่ของค่าความเข้มทุกค่าออกมาเป็นร้อยละของจำนวนข้อมูลทั้งหมด ส่วนโปรแกรม DNSTRS จะแสดงค่าระดับความเข้มของจุดต่าง ๆ ตามแนวนอนหรือแนวขวางของภาพก็ได้ ทั้งสองโปรแกรมจะให้คอมพิวเตอร์พิมพ์ออกมาเป็นกราฟ

Phase IV

RATIO

เป็นโปรแกรมสำหรับจัดทำข้อมูลชุดใหม่ ให้เป็นอัตราส่วนระหว่างข้อมูลในพื้นที่เดียวกันแต่ต่างช่วงคลื่น (Band) กัน ข้อมูลใหม่ที่ได้จะมีคุณสมบัติของทั้งสองช่วงคลื่นรวมกันอยู่ ทำให้รายละเอียดบางอย่างที่ไม่สามารถเห็นได้หรือเห็นชัดเจนด้วยข้อมูลจากช่วงคลื่นใดช่วงคลื่นหนึ่งปรากฏเด่นชัดขึ้น

TSTAT และ CLSFY

โปรแกรม TSTAT ทำหน้าที่คำนวณ parameters ต่าง ๆ เช่นค่าเฉลี่ยค่าเบี่ยงเบนมาตราฐานในทุก ๆ ช่วงคลื่นของข้อมูลในพื้นที่ตัวอย่าง (Training Area) สำหรับตัวอย่างแต่ละชนิดซึ่งผู้ใช้เลือกขึ้นมาโดยที่ข้อมูลในแต่ละช่วงคลื่นจะถูกนำมาทำเป็นอัตราส่วนต่อผลบวกของข้อมูลจุดนั้นในทุกช่วงคลื่นเสียก่อน parameters เหล่านี้จะถูกนำไปใช้ในโปรแกรม CLSFY ซึ่งจะตรวจสอบข้อมูลที่ต้องการวิเคราะห์แต่ละจุดในทุกช่วงคลื่น โดยวิธีที่เรียกว่า Spherical Classification ก่อนที่จะถูกระบุให้เป็น sample class ใด ๆ

สรุป

ปัจจุบันคณะวิจัยชุดที่ได้กล่าวถึงในตอนต้น ได้ทำการพัฒนาและปรับปรุงแก้ไขโปรแกรมต่าง ๆ ในชุดของลิกมัลส์ จนสามารถใช้งานได้ดีกับเครื่องคอมพิวเตอร์ของบริษัทศูนย์คอมพิวเตอร์แห่งประเทศไทย อย่างไรก็ดี โปรแกรมชุดลิกมัลส์ในส่วนของการประยุกต์เพื่อแยกแยะ (Classify) และวิเคราะห์ประเภทของข้อมูล อันมีโปรแกรม TSTAT และ CLSFY ซึ่งเป็นโปรแกรมชุดล่าสุดที่ได้รับจาก H. L. Wagner เมื่อเดือนสิงหาคม ปี 1978 ยังไม่มีโอกาสที่จะนำมาศึกษาและทดสอบสมรรถภาพของโปรแกรมดังกล่าว

เมื่อปลายปี 1977 คุณกันยา จิระพยุงไชย ได้ทำการวิจัยในระดับปริญญาโท ด้วยทุนสนับสนุนจากสภาวิจัยแห่งชาติ ได้เขียนโปรแกรมเพิ่มเติมเพื่อวิเคราะห์ประเภทของป่าไม้ในภาคตะวันออกของประเทศไทย โดยใช้เทคนิค Maximum Likelihood Ratio เรียกชื่อโปรแกรมนี้ว่า CLSSIFY และยังได้ปรับปรุงโปรแกรมในการจัดทำฟอร์แมทแบบลิกมัลส์ด้วยภาษา

โคบอล เพื่อลดเวลาคอมพิวเตอร์ในการทำงานในส่วนนี้ซึ่งแต่เดิมเขียนด้วยภาษาฟอร์แทรน 4 เปลืองเวลาคอมพิวเตอร์มาก นอกจากนี้ผู้เขียนเองก็ได้ดัดแปลงโปรแกรม DNSTRS เพื่อให้ใช้ได้กับข้อมูลจากเทปและแสดงลักษณะการเปลี่ยนแปลงของความเข้มทั้ง 4 ช่วงคลื่นพร้อมกันบนกราฟแผ่นเดียวกันได้ โดยที่โปรแกราเดิมใช้ได้เฉพาะกับข้อมูลที่ผ่านจากการ DTAP ของช่วงคลื่นใดช่วงคลื่นหนึ่งเท่านั้น

การจัดเรียงลำดับข้อมูลในฟอร์แมทของลิกมัลส์ ทำให้สามารถนำไปใช้ได้สะดวกและง่ายต่อการติดตาม มีความคล่องตัวในการนำไปใช้แม้กับเครื่องคอมพิวเตอร์ขนาดเล็ก การประยุกต์ข้อมูลจากระบบ MSS ของดาวเทียมสำรวจทรัพยากร เพื่อประโยชน์ในด้านต่าง ๆ กระทำได้ง่าย เพราะภาษาฟอร์แทรน 4 เหมาะสำหรับงานด้านวิทยาศาสตร์ประยุกต์ซึ่งจำเป็นจะต้องมีการคำนวณทางด้านคณิตศาสตร์.

โดย หาญ กิตติชานันท์
แผนกวิศวกรรมไฟฟ้า
จุฬาลงกรณมหาวิทยาลัย
24 มีนาคม 2522

ภาคผนวก ค.

รายละเอียดโปรแกรม

```
      IDENT IMAGE
C
      FILE  9=REFORM ,UNIT=TAPE,RECORD=9964,BLOCKING=1,FIXED,BUFFERS=1
      FILE  11=DTAP      ,UNIT=DISK,RECORD=670,BLOCKING=10,AREA=1000,LOCK
C     SUBROUTINE DTAP
C
C    ********************************************************************
C    * PROGRAM TO EXTRACT SUBSET OF OU-LIGMALS OUT IN ONE CHANNEL       *
C    * AND WRITE IT INTO A SCRATCH DESC FILE                            *
C    * THIS PROGRAM REQUIRES THE LINE-POINT MATRIX OF SUBSET AREA       *
C    ********************************************************************
C
      INTEGER LSTRT,LEND,LINCR,PSTRT,PEND,PINCR,ICHNUM,BUF(40),
     1NUMCHN,NPTS,LSG(5),LCNTR,INBUF(3320),LLAST/0/
      IUNIT=9
C
C   * READ LINE-POINT MATRIX OF SUB-AREA OF DESIRED CHANNEL
C
      READ(5,420)PSTRT,PEND,PINCR,LSTRT,LEND,LINCR,ICHNUM
  420 FORMAT(7I5)
      NPTS=((PEND-PSTRT)/PINCR)+1
      WRITE(6,333)
    6 REWIND IUNIT
C
C   * READ TAPE HEADER FROM OU-LIGMALS OUT ON INPUT UNIT
C
      READ(IUNIT,430,END=240,ERR=200)BUF,(INBUF(I),I=2,3)
  430 FORMAT(40A2,2I4)
      NUMCHN=4
      INBUF(4)=ICHNUM
C
C   * CHECK POINT AVAILABILITY TO SATISFY THE LIMITS OF SCAN LINES
C   * IF NOT, BRANCH TO GENERATE MESSAGE AND PROCEED
C
```

```
CARD NO   FILE NAME          (05/0) 2BCDIC                    5/07/79

    1 ...  ********************************************************************
    2 ...      IDENTIFICATION DIVISION.
    3 ...      PROGRAM-ID. RE-DATA.
    4 ...      AUTHOR. KANYA.
    5 ...      DATE-WRITTEN. 21 JULY 1978.
    6 ...      REMARKS. PROGRAM TO REARRANGE ERDS-DATA-TAPE.
    7 ...      ENVIRONMENT DIVISION.
    8 ...      CONFIGURATION SECTION.
    9 ...      SOURCE-COMPUTER. B-3700.
   10 ...      OBJECT-COMPUTER. B-3700.
   11 ...  ********************************************************************
   12 ...  ****THIS PROGRAM TO REFORMAT OCT DATA INTO CU-SIGNALS FORM
   13 ...  ****READ INPUT DATA  AS CHARACTER (1BYTE/CHAR)
   14 ...  ****THEN CHANGE THESE DATA  FROM HEXA TO DECIMAL VALUE
   15 ...  ****AND REFORMAT THEM TO CU-SIGNALS FORM
   16 ...  ****THE RESULTS FOR THIS PROGRAM ARE INTEGER (4BYTES/CHAR)
   17 ...  ********************************************************************
   18 ...      INPUT-OUTPUT SECTION.
   19 ...      FILE-CONTROL.
   20 ...          SELECT F00000 ASSIGN TO TAPE
   21 ...              RESERVE 1 ALTERNATE AREA.
   22 ...          SELECT CU-DATA ASSIGN TO TAPE
   23 ...              RESERVE 1 ALTERNATE AREA.
   24 ...          SELECT PRINT-DATA ASSIGN TO PRINTER
   25 ...              RESERVE 1 ALTERNATE AREA.
   26 ...      I-O-CONTROL.
   27 ...          MULTIPLE FILE TAPE "B17LIB" CONTAINS F00000.
   28 ...      DATA DIVISION.
   29 ...      FILE SECTION.
   30 ...      FD  F00000
   31 ...          RECORDING MODE IS STANDARD
   32 ...          BLOCK CONTAINS 1  RECORDS
   33 ...          RECORD CONTAINS 3320 CHARACTERS
   34 ...          LABEL RECORD IS STANDARD
```

```
35 ...          DATA RECORDS ARE EROS-HEADER-RECORD, EROS-ANNO-RECORD,
36 ...              EROS-DATA-RECORD.
37 ...      01  EROS-HEADER-RECORD.
38 ...          02  HEADER                  PC X      OC 40 TIMES.
39 ...      01  EROS-ANNO-RECORD            PC X(624).
40 ...      01  EROS-DATA-RECORD            PC X(3320).
41 ...      FD  CU-DATA
42 ...          RECORDING MODE IS STANDARD
43 ...          BLOCK CONTAINS 1  RECORDS
44 ...          LABEL RECORD IS STANDARD
45 ...          VALUE OF ID IS "REFORM"
46 ...          DATA RECORDS ARE CU-HEADER-RECORD, CU-ANNO-RECORD,
47 ...              CU-DATA-RECORD.
48 ...      01  CU-HEADER-RECORD.
49 ...          02  HEADER-RECORD1          PC X(40).
50 ...          02  HEADER-RECORD2          PC 9999.
51 ...          02  HEADER-RECORD3          PC 9999.
52 ...      01  CU-ANNO-RECORD              PC X(624).
53 ...      01  CU-DATA-RECORD.
54 ...          02  ENUM                    PC 9999.
55 ...          02  DATA-RECORD             PC 9(3)   OC 3320 TIMES.
56 ...      FD  PRINT-DATA
57 ...          RECORDING MODE IS STANDARD
58 ...          RECORD CONTAINS 132 CHARACTERS.
59 ...      01  PRINT-OUT                   PC X(132).
60 ...  ************************************************************
61 ...  ****SET PARAMETERS AND INITIALIZE
62 ...  ************************************************************
63 ...      WORKING-STORAGE SECTION.
64 ...      77  N                           PC 9999          VA 0.
65 ...      77  LCNTR                       PC 9999          VA 0.
66 ...      77  COUNTER                     PC 9999          VA 0.
67 ...      77  LENR                        PC 9999.
68 ...      77  LENC                        PC 9999.
69 ...      77  SW                          PC 9 COMP.
70 ...      77  A-HEXA-VALUE                PC 9 COMP        VA 8A0.
```

```
 71 ...      77  B-HEXA-VALUE                PC 9 CMP          VA @B@.
 72 ...      77  C-HEXA-VALUE                PC 9 CMP          VA @C@.
 73 ...      77  D-HEXA-VALUE                PC 9 CMP          VA @D@.
 74 ...      77  E-HEXA-VALUE                PC 9 CMP          VA @E@.
 75 ...      77  F-HEXA-VALUE                PC 9 CMP          VA @F@.
 76 ...      77  X1                          PC 99 CMP.
 77 ...      77  X2                          PC 99 CMP.
 78 ...      77  X3                          PC 99 CMP.
 79 ...      77  X4                          PC 99 CMP.
 80 ...      01  YY                          PC 999 CMP.
 81 ...      01  ZZ                          PC 9999 CMP.
 82 ...      03  J.
 83 ...          02  J1                      PC 9999           VA 1.
 84 ...          02  J2                      PC 9999           VA 2.
 85 ...          02  J3                      PC 9999           VA 3.
 86 ...          02  J4                      PC 9999           VA 4.
 87 ...          02  J5                      PC 9999           VA 5.
 88 ...          02  J6                      PC 9999           VA 6.
 89 ...          02  J7                      PC 9999           VA 7.
 90 ...          02  J8                      PC 9999           VA 8.
 91 ...      01  HEAD-1.
 92 ...          02  RRR-H1                  PC X CMP.
 93 ...          02  RRR-H2                  PC X CMP.
 94 ...      01  HEAD-2.
 95 ...          02  RRR-H3                  PC X CMP.
 96 ...          02  RRR-H4                  PC X CMP.
 97 ...      01  REC-ERROR-DATA-RECORD.
 98 ...          02  RR-CHAR                 OC 3320 TIMES.
 99 ...              03  RRR-D1              PC X CMP.
100 ...              03  RRR-D2              PC X CMP.
101 ...      01  REC-OU-DATA-RECORD.
102 ...          02  RR-DATA-RECORD          PC X(80)   OC 3320 TIMES.
103 ...      01  DATA-OUT1.
104 ...          02  FILLER          PC X(25)   VA "DATA RECORD FORMAT SIZE= ".
105 ...          02  REC-SIZE                PC 9999.
106 ...          02  FILLER                  PC X(7)          VA SPACES.
```

```
107 ...          02  FILLER                  PC X(17)        VA "POINT PER LINE=  ".
108 ...          02  DATA-SIZE               PC 9999.
109 ...          02  FILLER                  PC X(75)        VA SPACES.
110 ...      01  DATA-OUT2.
111 ...          02  FILLER                  PC X(12)        VA " SCAN LINE# ".
112 ...          02  LINE-NO                 PC 9999.
113 ...          02  FILLER                  PC X(15)        VA "HAS BEEN COPIED".
114 ...          02  FILLER                  PC X(101)       VA SPACES.
115 ...      PROCEDURE DIVISION.
116 ...      DECLARATIVES.
117 ...      START SECTION.
118 ...          USE ERROR PROCEDURE ON F00000.
119 ...      AAA.
120 ...          MOVE 1 TO SW.
121 ...      END DECLARATIVES.
122 ...      OPEN-FILE.
123 ...          OPEN INPUT F00000.
124 ...  ************************************************************
125 ...  ****READ HEADER INFO. THEN FIND DATA BLOCK SIZE, ADJUSTED*
126 ...  ****LINE LENGTH AND WRITE THEM ON OUTPUT TAPE
127 ...  ************************************************************
128 ...          READ F00000.
129 ...          MOVE ERR-HEADER-RECORD TO HEADER-RECORD1.
130 ...          MOVE      HEADER(17) TO HEAD-1.
131 ...          MOVE      HEADER(18) TO HEAD-2.
132 ...          PERFORM SP-LOOP THROUGH SP-L4.
133 ...          MOVE ZZ TO HEADER-RECORD2,LENR,REC-SIZE.
134 ...          DIVIDE ZZ BY 8 GIVING LENL.
135 ...          MOVE      HEADER(39) TO HEAD-1.
136 ...          MOVE      HEADER(40) TO HEAD-2.
137 ...          PERFORM SP-LOOP THROUGH SP-L4.
138 ...          MOVE ZZ TO HEADER-RECORD3,DATA-SIZE.
139 ...          OPEN OUTPUT CU-DATA.
140 ...          WRITE CU-HEADER-RECORD.
141 ...          OPEN OUTPUT PRINT-DATA.
142 ...          WRITE PRINT-OUT FROM DATA-OUT1.
```

```
143 ...  ********************************************************************
144 ...  ****READ ANNOTATION BLOCK DATA                                      .
145 ...  ********************************************************************
146 ...           READ F00000.
147 ...           WRITE CU-ANNO-RECORD FROM EROS-ANNO-RECORD.
148 ...           PERFORM RSKIP N TIMES.
149 ...           ADD N TO LCNTR.
150 ...           PERFORM REED 1576 TIMES.
151 ...       CLOSE-FILE.
152 ...           CLOSE F00000 WITH RELEASE.
153 ...           CLOSE CU-DATA WITH RELEASE.
154 ...           CLOSE PRINT-DATA WITH RELEASE.
155 ...           STOP RUN.
156 ...       RSKIP.
157 ...           READ F00000.
158 ...       REED.
159 ...           ADD 1 TO LCNTR.
160 ...           MOVE 0 TO SW.
161 ...  ********************************************************************
162 ...  ****READ A LINE OF DATA CHANGE ITS VALUE AND REFORMAT
163 ...  ********************************************************************
164 ...           READ F00000 AT END GO TO CLOSE-FILE.
165 ...           IF SW = 1 GO TO REED.
166 ...           MOVE EROS-DATA-RECORD TO RE-EROS-DATA-RECORD.
167 ...           PERFORM LOOP-1 THROUGH LOOP-5 LENR TIMES.
168 ...           PERFORM REFORM-LOOP LENL TIMES.
169 ...  ********************************************************************
170 ...  ****WRITE REFORMATED DATA ON OUTPUT TAPE
171 ...  ********************************************************************
172 ...           MOVE LCNTR TO LNUM.
173 ...           MOVE LCNTR TO LINE-NO.
174 ...           WRITE CU-DATA-RECORD.
175 ...           WRITE PRINT-OUT FROM DATA-OUT2.
176 ...           SUBTRACT LENR FROM J1, J2, J3, J4, J5, J6, J7, J8.
177 ...           MOVE 0 TO COUNTER.
```

```
178 ...  ************************************************************
179 ...  ****COMPUTE THE VALUE OF DATA BLOCK SIZE  AND DATA ADJUSTED
180 ...  ****LINE LENGTH
181 ...  ************************************************************
182 ...      SP-LOOP.
183 ...          IF RRR-H1=A-HEXA-VALUE MOVE 10 TO X1 GO TO SP-L1.
184 ...          IF RRR-H1=B-HEXA-VALUE MOVE 11 TO X1 GO TO SP-L1.
185 ...          IF RRR-H1=C-HEXA-VALUE MOVE 12 TO X1 GO TO SP-L1.
186 ...          IF RRR-H1=D-HEXA-VALUE MOVE 13 TO X1 GO TO SP-L1.
187 ...          IF RRR-H1=E-HEXA-VALUE MOVE 14 TO X1 GO TO SP-L1.
188 ...          IF RRR-H1=F-HEXA-VALUE MOVE 15 TO X1 GO TO SP-L1.
189 ...          MOVE RRR-H1 TO X1.
190 ...      SP-L1.
191 ...          IF RRR-H2=A-HEXA-VALUE MOVE 10 TO X2 GO TO SP-L2.
192 ...          IF RRR-H2=B-HEXA-VALUE MOVE 11 TO X2 GO TO SP-L2.
193 ...          IF RRR-H2=C-HEXA-VALUE MOVE 12 TO X2 GO TO SP-L2.
194 ...          IF RRR-H2=D-HEXA-VALUE MOVE 13 TO X2 GO TO SP-L2.
195 ...          IF RRR-H2=E-HEXA-VALUE MOVE 14 TO X2 GO TO SP-L2.
196 ...          IF RRR-H2=F-HEXA-VALUE MOVE 15 TO X2 GO TO SP-L2.
197 ...          MOVE RRR-H2 TO X2.
198 ...      SP-L2.
199 ...          IF RRR-H3=A-HEXA-VALUE MOVE 10 TO X3 GO TO SP-L3.
200 ...          IF RRR-H3=B-HEXA-VALUE MOVE 11 TO X3 GO TO SP-L3.
201 ...          IF RRR-H3=C-HEXA-VALUE MOVE 12 TO X3 GO TO SP-L3.
202 ...          IF RRR-H3=D-HEXA-VALUE MOVE 13 TO X3 GO TO SP-L3.
203 ...          IF RRR-H3=E-HEXA-VALUE MOVE 14 TO X3 GO TO SP-L3.
204 ...          IF RRR-H3=F-HEXA-VALUE MOVE 15 TO X3 GO TO SP-L3.
205 ...          MOVE RRR-H3 TO X3.
206 ...      SP-L3.
207 ...          IF RRR-H4=A-HEXA-VALUE MOVE 10 TO X4 GO TO SP-L4.
208 ...          IF RRR-H4=B-HEXA-VALUE MOVE 11 TO X4 GO TO SP-L4.
209 ...          IF RRR-H4=C-HEXA-VALUE MOVE 12 TO X4 GO TO SP-L4.
210 ...          IF RRR-H4=D-HEXA-VALUE MOVE 13 TO X4 GO TO SP-L4.
211 ...          IF RRR-H4=E-HEXA-VALUE MOVE 14 TO X4 GO TO SP-L4.
212 ...          IF RRR-H4=F-HEXA-VALUE MOVE 15 TO X4 GO TO SP-L4.
213 ...          MOVE RRR-H4 TO X4.
214 ...      SP-L4.
215 ...          COMPUTE ZZ = X1*16**3 + X2*16**2 + X3*16 + X4.
```

```
 216 ...  ************************************************************
 217 ...  ****COMPUTE THE DECIMAL VALUE FOR EACH DATA
 218 ...  ************************************************************
 219 ...      LOOP-1.
 220 ...          ADD 1 TO COUNTER.
 221 ...          IF RRR-D1(COUNTER) = A-HEXA-VALUE MOVE 10 TO X1 GO TO LOOP-2.
 222 ...          IF RRR-D1(COUNTER) = B-HEXA-VALUE MOVE 11 TO X1 GO TO LOOP-2.
 223 ...          IF RRR-D1(COUNTER) = C-HEXA-VALUE MOVE 12 TO X1 GO TO LOOP-2.
 224 ...          IF RRR-D1(COUNTER) = D-HEXA-VALUE MOVE 13 TO X1 GO TO LOOP-2.
 225 ...          IF RRR-D1(COUNTER) = E-HEXA-VALUE MOVE 14 TO X1 GO TO LOOP-2.
 226 ...          IF RRR-D1(COUNTER) = F-HEXA-VALUE MOVE 15 TO X1 GO TO LOOP-2.
 227 ...          MOVE RRR-D1(COUNTER) TO X1.
 228 ...      LOOP-2.
 229 ...          IF RRR-D2(COUNTER) = A-HEXA-VALUE MOVE 10 TO X2 GO TO LOOP-3.
 230 ...          IF RRR-D2(COUNTER) = B-HEXA-VALUE MOVE 11 TO X2 GO TO LOOP-3.
 231 ...          IF RRR-D2(COUNTER) = C-HEXA-VALUE MOVE 12 TO X2 GO TO LOOP-3.
 232 ...          IF RRR-D2(COUNTER) = D-HEXA-VALUE MOVE 13 TO X2 GO TO LOOP-3.
 233 ...          IF RRR-D2(COUNTER) = E-HEXA-VALUE MOVE 14 TO X2 GO TO LOOP-3.
 234 ...          IF RRR-D2(COUNTER) = F-HEXA-VALUE MOVE 15 TO X2 GO TO LOOP-3.
 235 ...          MOVE RRR-D2(COUNTER) TO X2.
 236 ...      LOOP-3.
 237 ...          MULTIPLY X1 BY 16 GIVING YY.
 238 ...          ADD X2 TO YY.
 239 ...          MOVE YY TO RR-DATA-RECORD(COUNTER).
 240 ...  ************************************************************
 241 ...  ****REFORMATTING SECTION.
 242 ...  ************************************************************
 243 ...      REFORM-LOOP.
 244 ...          MOVE RR-DATA-RECORD(J1) TO DATA-RECORD(J1).
 245 ...          MOVE RR-DATA-RECORD(J3) TO DATA-RECORD(J2).
 246 ...          MOVE RR-DATA-RECORD(J5) TO DATA-RECORD(J3).
 247 ...          MOVE RR-DATA-RECORD(J7) TO DATA-RECORD(J4).
 248 ...          MOVE RR-DATA-RECORD(J2) TO DATA-RECORD(J5).
 249 ...          MOVE RR-DATA-RECORD(J4) TO DATA-RECORD(J6).
 250 ...          MOVE RR-DATA-RECORD(J6) TO DATA-RECORD(J7).
 251 ...          MOVE RR-DATA-RECORD(J8) TO DATA-RECORD(J8).
 252 ...          ADD 8 TO J1,J2,J3,J4,J5,J6,J7,J8.
 253 ...      END-REFORM.
 254 ...      END-OF-JOB.
END OF FILE       254 CARDS
```

```
      INBUF(3)=INBUF(3)/NUMCHN
      IF(PEND.GT.INBUF(3))GO TO 225
C
C  * SET UP HEADER INFO FOR QUICK ACCESS SUBSET. FILE HEADER
C
   12 INBUF(3)=NPTS
      INBUF(5)=PSTRT
      INBUF(6)=PEND
      INBUF(7)=PINCR
      INBUF(8)=LSTRT
      INBUF(9)=LEND
      INBUF(10)=LINCR
      REWIND 11
C
C  * WRITE HEADER BLOCK ON OUTPUT UNIT
C
      WRITE(11)(INBUF(I),I=2,10)
C
C  * READ AND WRITE LANDSAT ANNOTATION BLOCK INTO OUTPUT FILE
C
      READ(IUNIT,450,END=240,ERR=200)(INBUF(I),I=1,144)
  450 FORMAT(144A1)
      WRITE(11)(INBUF(I),I=1,144)
      WRITE(6,309)
      LTST=LSTRT
      LCNTR=0
C
C  * SET UP INTERNAL POINTERS  TO SELECT PROPER PIXELS FOR SUBSET
C
      ISTRT=ICHNUM+((PSTRT-1)*NUMCHN)
      IEND=NUMCHN*PEND
      IINCR=(NUMCHN*PINCR)
C
C  * READ A LINE OF DATA AND BRANCH TO TEST FOR DESIRED LINE NO. RANGE
C
      READ(IUNIT,470,END=250,ERR=215)LNUM,INBUF
```

```
 71 ...       LSKIP=LTST-LNUM-1
 72 ...       IF(LSKIP.LE.-1)GO TO 14
 73 ...       IF(LSKIP.LE.1)GO TO 15
 74 ...       DO 100 I=1,LSKIP
 75 ...       READ(IUNIT,470)
 76 ...  100  CONTINUE
 77 ...       GO TO 13
 78 ...   14  LBACK=LNUM-LTST+1
 79 ...       DO 115 I=1,LBACK
 80 ...       BACKSPACE IUNIT
 81 ...  115  CONTINUE
 82 ...   13  READ(IUNIT,470,END=250,ERR=215)LNUM,INBUF
 83 ...  470  FORMAT(I4,10(332I3))
 84 ...C      WRITE(6,310)LTST,LNUM
 85 ...C 310  FORMAT('  LTST=',I5,'  LNUM=',I5)
 86 ...C
 87 ...C    * LOOP UNTIL FIND FIRST LINE NUMBER TO BE USED IN SUBSET
 88 ...C
 89 ...   15  IF(LNUM.LT.LTST)GO TO 13
 90 ...       IF(LNUM.GT.LTST)WRITE(6,311)LTST,LNUM
 91 ...       WRITE(11)LNUM,(INBUF(I),I=ISTRT,IEND,IINCR)
 92 ...       LCNTR=LCNTR+1
 93 ...       LLAST=LNUM
 94 ...   16  LTST=LTST+LINCR
 95 ...C
 96 ...C    * CHECK LINE NUMBERS FOR MISSING LINES ETC.
 97 ...   17  IF(LTST.GT.LEND)GO TO 150
 98 ...       IF(LLAST.GE.LTST)GO TO 18
 99 ...       GO TO 13
100 ...   18  LTST=LTST+1
101 ...       GO TO 17
102 ...C
103 ...C    * SEMI DIAGNOSTIC ERROR MESSAGE. CHECK FORMAT STATEMENTS
104 ...C    * TO DETERMINE CONDITIONS WHEN USED.
105 ...C
106 ...  150 WRITE(6,319)LCNTR,LLAST
```

```
      ENDFILE 11
      REWIND 11
      WRITE(6,321)
      STOP
  200 WRITE(6,323)IUNIT
      WRITE(6,329)
      WRITE(6,327)
      STOP
  215 WRITE(6,323)
      WRITE(6,331)LTST
      GO TO 13
  225 WRITE(6,337)PEND,INBUF(3)
      WRITE(6,339)INBUF(3)
      PEND=INBUF(3)
      NPTS=((PEND-PSTRT)/PINCR)+1
      GO TO 12
  240 WRITE(6,343)
      WRITE(6,329)
      WRITE(6,327)
      STOP
  250 WRITE(6,343)
      WRITE(6,331)LTST
      WRITE(6,347)LCNTR
      ENDFILE 11
      WRITE(6,327)
      STOP
C
C     * WRITE OUTPUT FORMAT STATEMENTS
C
  303 FORMAT(' * LIGMALS SUBSET EXTRACTION ROUTINE',//,' * READY TO PROCE
     1ED',//,' * REWINDING INPUT TAPE - STAND BY')
  309 FORMAT(' * HEADER INFO COMPLETED, SUBSET EXTRACTION BEGINS',//,' *
     1STAND BY')
  311 FORMAT(' ***WARNING***',//,' * LINE NUMBER ',I5,' WAS NOT FOUND',//,
     1' * LINE NUMBER ',I5,' HAS BEEN SUBSTITUTED FOR IT',//,' * EXECUTIO
     2N CONTINUING. STAND BY')
```

```
  143 ...   317 FORMAT(' ***WARNING***',/,' * LINE NUMBER ',I5,' WAS NOT FOUND',/,
  144 ...      1' * LINE DELETED FROM OUTPUT FILE, EXECUTION CONTINUING')
  145 ...   319 FORMAT(' * DATA SUBSET COMPLETED',/,' * ',I5,' LINES WERE COPIED,
  146 ...      1ENDING WITH LINE ',I5)
  147 ...   321 FORMAT(' * OUTPUT FILE TERMINATED.')
  148 ...   323 FORMAT(' ***ERROR***',/,' * A FATAL READ ERROR HAS OCCURED ON UNIT
  149 ...      1 ', I5)
  150 ...   327 FORMAT(' * SUBSET EXTRACTION ABORTED.')
  151 ...   329 FORMAT(' * ERROR OCCURED DURING AN ATTEMPT TO READ HEADER INFORMAT
  152 ...      1ION')
  153 ...   331 FORMAT(' * ERROR OCURED DURING AN ATTEMPT TO READ DATA LINE#',I5)
  154 ...   337 FORMAT(' ***ERROR***',/,' * NUMBER OF DATA POINTS REQUESTED EXCEED
  155 ...      1S DATA BASE AVAILABLE',/,' * POINT # ',I5,' WAS LAST POINT REQUEST
  156 ...      2ED.',/,' * BUT DATA SET CONTAINS ONLY ',I5,' POINTS')
  157 ...   339 FORMAT(' * TRUNCATED AT POINT # ',I5)
  158 ...   343 FORMAT(' ***ERROR***',/,' AN UNEXPECTED END OF FILE WAS ENCOUNTERE
  159 ...      1D')
  160 ...   347 FORMAT(' * ',I5,' LINES HAVE BEEN SUCCESSFULLY COPIED.')
  161 ...      END
END OF FILE        161 CARDS
```

```
      1 ...IDENT IMAGE
      2 ...FILE  9=REFORM,UNIT=TAPE,RECORD=9964,BLOCKING=1,FIXED,BUFFERS=1
      3 ...FILE  1=MSG,UNIT PRINTER
      4 ...C     SUBROUTINE DNSTRS FOR 4-CHANNEL
      5 ...      INTEGER LSTRT,LEND,LINCR,PSTRT,PEND,PINCR,ICHNUM,NUMCHN,NPTS,LS(8)
      6 ...     1,LCNTR,INBUF(3320)
      7 ...      INTEGER LINE/'LINE'/,ITES,IHLD
      8 ...      INTEGER PIXEL(105),BL/'    '/,AXIS/'+   '/,BA/'4   '/,BB/'5   '/,B
      9 ...     1C/'6   '/,BD/'7   '/
     10 ...      INTEGER  BUF(40)
     11 ...      WRITE(6,301)
     12 ...      REWIND 9
     13 ...   10 READ(9,430,END=205,ERR=215) BUF,(INBUF(I),I=2,3)
     14 ...      ISZ=INBUF(2)
     15 ...      NPTS=INBUF(3)/4
     16 ...      READ(5,401)ITES,IHLD
     17 ...      READ(9,450,END=205,ERR=215) (INBUF(I),I=1,144)
     18 ...      IF(ITES.EQ.LINE) GO TO 20
     19 ...      IF((IHLD+1).LT.1.OR.(IHLD+1).GT.NPTS)GO TO 245
     20 ...      WRITE(1,402)ITES,IHLD
     21 ...      IHLD=((IHLD-1)*4)+1
     22 ...      DO 12 I=1,105
     23 ...      PIXEL(I)=BL
     24 ...   12 CONTINUE
     25 ...      PIXEL(3)=AXIS
     26 ...   19 READ(9,470,END=225,ERR=235)LNUM,INBUF
     27 ...      IAA=(INBUF(IHLD)/2)+3
     28 ...      IAB=(INBUF(IHLD+1)/2)+3
     29 ...      IAC=(INBUF(IHLD+2)/2)+3
     30 ...      IAD=(INBUF(IHLD+3)/2)+3
     31 ...      IF(IAA.LT.1.OR.IAA.GT.105) GO TO 13
     32 ...      PIXEL(IAA)=BA
     33 ...      IF(IAA.EQ.IAB) GO TO 14
     34 ...   13 IF(IAB.LT.1.OR.IAB.GT.105) GO TO 15
     35 ...      PIXEL(IAB)=BB
     36 ...   14 IF(IAA.EQ.IAC.OR.IAB.EQ.IAC) GO TO 16
     37 ...   15 IF(IAC.LT.1.OR.IAC.GT.105) GO TO 17
     38 ...      PIXEL(IAC)=BC
```

```
39 ...    16 IF(IAA.EQ.IAD.OR.IAB.EQ.IAD.OR.IAC.EQ.IAD) GO TO 18
40 ...    17 IF(IAD.LT.1.OR.IAD.GT.105) GO TO 18
41 ...       PIXEL(IAD)=BD
42 ...    18 WRITE(1,403)LNUM,INBUF(IHLD),(INBUF(IHLD+IB),IB=1,3),PIXEL
43 ...       PIXEL(IAA)=BL
44 ...       PIXEL(IAB)=BL
45 ...       PIXEL(IAC)=BL
46 ...       PIXEL(IAD)=BL
47 ...       IF(IAA.EQ.3.OR.IAB.EQ.3.OR.IAC.EQ.3.OR.IAD.EQ.3) PIXEL(3)=AXIS
48 ...       GO TO 19
49 ...    20 READ(9,470,END=225,ERR=235)LNUM,INBUF
50 ...       IF(IHLD.LT.LNUM) GO TO 245
51 ...       GO TO 23
52 ...    22 READ(9,470,END=225,ERR=235)LNUM,INBUF
53 ...    23 IF(LNUM.LT.IHLD) GO TO 22
54 ...       WRITE(1,402)ITES,IHLD
55 ...       DO 24 I=1,105
56 ...       PIXEL(I)=BL
57 ...    24 CONTINUE
58 ...       PIXEL(3)=AXIS
59 ...       DO 31 I=1,ISZ,4
60 ...       IAA=(INBUF(I)/2)+3
61 ...       IAB=(INBUF(I+1)/2)+3
62 ...       IAC=(INBUF(I+2)/2)+3
63 ...       IAD=(INBUF(I+3)/2)+3
64 ...       IF(IAA.LT.1.OR.IAA.GT.105) GO TO 25
65 ...       PIXEL(IAA)=BA
66 ...       IF(IAA.EQ.IAB) GO TO 26
67 ...    25 IF(IAB.LT.1.OR.IAB.GT.105) GO TO 27
68 ...       PIXEL(IAB)=BB
69 ...    26 IF(IAA.EQ.IAC.OR.IAB.EQ.IAC) GO TO 28
70 ...    27 IF(IAC.LT.1.OR.IAC.GT.105) GO TO 29
71 ...       PIXEL(IAC)=BC
72 ...    28 IF(IAA.EQ.IAD.OR.IAB.EQ.IAD.OR.IAC.EQ.IAD) GO TO 30
73 ...    29 IF(IAD.LT.1.OR.IAD.GT.105) GO TO 30
74 ...       PIXEL(IAA)=BL
75 ...       PIXEL(IAB)=BL
76 ...       PIXEL(IAC)=BL
77 ...       PIXEL(IAD)=BD
```

```
   78 ...    30 WRITE(1,403) I,INBUF(I),(INBUF(I+IA) IA=1,3),PIXEL
   79 ...       PIXEL(IAD)=BL
   80 ...       IF(I/A.EQ.3.OR.IAB.EQ.3.OR.IAC.EQ.3.(R.IAD.EQ.3) PIXEL(3)=AXIS
   81 ...    31 CONT NUE
   82 ...   205 WRITE(6,404)
   83 ...       WRITE(6,406)
   84 ...       STOP
   85 ...   215 WRITE(6,410)
   86 ...       WRITE(6,406)
   87 ...       STOP
   88 ...   225 WRITE(1,405)
   89 ...       WRITE(6,407)
   90 ...       STOP
   91 ...   235 WRITE(6,410)
   92 ...       STOP
   93 ...   245 WRITE(6,412)
   94 ...       STOP
   95 ...   402 FORMAT('1','** DENSITOMETRY PLOT.',A ,'TRACE',/,3X,'INDEX #',I6,/,
   96 ...      1'2','PIXEL #',5X,'BRIGHTNESS',/,14X, OF',/,11X,'CHANNEL #',7X,'0',
   97 ...      29X,'20',8X,'40',8X,'60',8X,'80',7X,' 00',7X,'120',7X,'140',7X,'160
   98 ...      3',7X,'180',7X,'200',//,9X,'4',3X,'5' 3X,'6',3X,'7',/,2(27X,10('+',
   99 ...      49X),/),27X,100('+'),2(/,27X,10('+',9 )))
  100 ...   403 FORMAT('9',I4,3X,4I4,2X,105A1)
  101 ...   404 FORMAT(' ***ERROR***','/,' * AN UNEXP CTED END OF FILE HAS ENCOUNTE
  102 ...      1RED')
  103 ...   405 FORMAT('1',' * END OF PLOT')
  104 ...   406 FORMAT(' * ERROR OCCURED DURING ATTE T TO READ HEADER RECORD')
  105 ...   407 FORMAT(' * NORMAL TERMINATATION. RET N TO LIGMALS C=MODE')
  106 ...   410 FORMAT(' ***ERROR***','/,' * FATAL RE ) ERROR ON INPUT UNIT','/,
  107 ...      1' * EXECUTION ABORTED')
  108 ...   412 FORMAT(' ***ERROR***','/,' * INVALID DEX NUMBER ON INPUT','/,
  109 ...      1' * EXECUTION ABORTED')
  110 ...   301 FORMAT(' * DENSITOMETRIC PLOTTING R TINE')
  111 ...   400 FORMAT(I1)
  112 ...   401 FORMAT(A4,2X,I4)
  113 ...   430 FORMAT(40A1,2I4)
  114 ...   450 FORMAT(144A1)
  115 ...   470 FORMAT(I4,10(332I3))
  116 ...       END
END OF FILE        116 CARDS
```

```
CARD NO   FILE: CARDS                (06/0) EBCDIC                    5/07/79

     1 ...IDENT IMAGE
     2 ...C
     3 ...FILE  9=REFORM ,UNIT=TAPE,RECORD=9964,BLOCKING=1,FIXED,BUFFERS=1
     4 ...FILE  10=MSG,UNIT=PRINTER
     5 ...C     SUBROUTINE ALSET
     6 ...      INTEGER LSTRT,LEND,LINCR,PSTRT,PEND,PINCR,NPTS,ICHNUM,NUMCHN,
     7 ...     1LS(8),LCNTR,INBUF(3320),IMIN,IMAX,ITOT,IMN
     8 ...      WRITE(6,301)
     9 ...      IUNIT=9
    10 ...      WRITE(10,10)
    11 ...      READ (5,20)  PSTRT,PEND,PINCR,LSTRT,LEND,LINCR,ICHNUM
    12 ...   10 FORMAT(' * ENTER COORDINATE OF INTERESTED AREA',/,
    13 ...     1' * CONSIST OF: PSTRT PEND PINCR LSTRT LEND LINCR :FORMAT(6I5)')
    14 ...   20 FORMAT (7I5)
    15 ...      NPTS=((PEND-PSTRT)/PINCR)+1
    16 ...      WRITE(10,309)
    17 ...      READ(IUNIT,401,END=200,ERR=215)(INBUF(I),I=2,3)
    18 ...      NUMCHN=4
    19 ...      LCNTR=0
    20 ...      ITOT=0
    21 ...      PCNTR=0
    22 ...      SQPCN=0
    23 ...      IMAX=0
    24 ...      IMIN=32767
    25 ...      INBUF(3)=INBUF(3)/NUMCHN
    26 ...      IF(PEND.GT.INBUF(3))GO TO 225
    27 ...   12 READ(IUNIT,403,END=200,ERR=215)(INBUF(I),I=1,144)
    28 ...      WRITE(6,404)(INBUF(I),I=1,125)
    29 ...      LTST=LSTRT
    30 ...      ISTRT=ICHNUM+((PSTRT-1)*NUMCHN)
    31 ...      IEND=NUMCHN*PEND
    32 ...      IINCR=(NUMCHN*PINCR)
    33 ...   13 READ(IUNIT,405,END=30,ERR=235)LNUM,IBUF
    34 ...      IF(LNUM.LT.LTST)GO TO 13
```

```
      DO 24 I=ISTRT,IEND,IINCR
      IF(INBUF(I).LT.-511.OR.INBUF(I).GT.4096)GO TO 24
      PCNTR=PCNTR+INBUF(I)
      SQPCN=SQPCN+INBUF(I)**2
      IF(INBUF(I).GT.IMAX)IMAX=INBUF(I)
      IF(INBUF(I).LT.IMIN)IMIN=INBUF(I)
   24 CONTINUE
      ITOT=ITOT+NPTS
      IF(LNUM.EQ.LEND)GO TO 30
      GO TO 13
   30 IMN=(PCNTR/ITOT)
      SDSQ=SQPCN-(PCNTR**2)/ITOT
      SDSQ=SDSQ/ITOT
      SD=SQRT(SDSQ)
      WRITE(6,300)IMN,SD,IMIN,IMAX
  300 FORMAT(5X,'MEAN      = ',I4,/,5X,'SD          = ',F11.6,/,5X,
     1'MINIMUM   = ',I4,/,5X,'MAXIMUM   = ',I4)
      LS(1)=IMIN-100
      LS(2)=IMN-(1.1*SD)
      LS(3)=IMN-(.56*SD)
      LS(4)=IMN-(.19*SD)
      LS(5)=IMN+(.25*SD)
      LS(6)=IMN+(.78*SD)
      LS(7)=IMN+(1.75*SD)
      LS(8)=IMAX+100
      WRITE(6,311)ITOT,LS
      STOP
  200 WRITE(6,313)
      WRITE(6,315)
      STOP
  215 WRITE(6,317)
      WRITE(6,315)
      STOP
  225 WRITE(6,337)PEND,INBUF(3)
      PEND=INBUF(3)
      NPTS=((PEND-PSTRT)/PINCR)+1
      GO TO 12
```

```
      72 ...   235 WRITE(6,317)
      73 ...       WRITE(6,319)
      74 ...       GO TO 30
      75 ...   301 FORMAT(' * LIGMALS AUTO-LEVEL SET ROUTINE')
      76 ...   309 FORMAT(' * LEVEL SET INFO. NOW BEING PROCESSED',/,' * STAND BY')
      77 ...   311 FORMAT(' * LEVEL SET CONCLUDED. NUMBER OF POINTS SAMPLED=',I9,/,
      78 ...      1' * LEVEL SET CONSTANTS=',8(1X,I4,',' ),/,' * CONSTANTS HAVE BEEN
      79 ...      2SET INTO VIDEO&GRAY MAP ROUTINES',/,' * RETURNING TO LIGMALS COMMA
      80 ...      3ND MODE')
      81 ...   313 FORMAT(' ***ERROR***',/,' * AN UNEXPECTED END OF FILE HAS BEEN ENC
      82 ...      1OUNTERED')
      83 ...   315 FORMAT(' * ERROR OCCURED DURING AN ATTEMPT TO READ HEADER INFO.',
      84 ...      1/,' * LEVEL SET ATTEMPT ABORTED')
      85 ...   317 FORMAT(' ***ERROR***',/,' * A FATAL READ ERROR HAS OCCURED')
      86 ...   319 FORMAT(' * ERROR OCCURED DURING AN ATTEMPT TO READ VIDEO INFO.',
      87 ...      1/,' * WILL ATTEMPT TO PROCESS L-SET WITH INCOMPLETE DATA')
      88 ...   337 FORMAT(' ***ERROR***',/,' * NUMBER OF DATA POINTS REQUESTED EXCEED
      89 ...      1S DATA BASE AVAILABLE',/,' * POINT # ',I5,' HAS LAST POINT REQUEST
      90 ...      2ED',/,' * BUT DATA SET CONTAINS ONLY ',I5,' POINTS')
      91 ...   401 FORMAT(40A1,2I4)
      92 ...   403 FORMAT(144A1)
      93 ...   404 FORMAT(/,3X,125A1,/)
      94 ...   405 FORMAT(I4,10(33213))
      95 ...       END
END OF FILE          95 CARDS
```

```
CARD NO   FILE: CARDS                (06/0) EBCDIC                    5/07/79

      1 ...IDENT IMAGE
      2 ...C
      3 ...FILE  11=DTAP     ,UNIT=DISK,RECORD=670,BLOCKING=10,AREA=1000,LOCK
      4 ...FILE  6=OPRINT,UNIT=PRINTER
      5 ...FILE  1=MSG,UNIT=PRINTER
      6 ...C     SUBROUTINE PMAP
      7 ...C    ************************************************************
      8 ...C    * PROGRAM TO GENERATE LINE PRINTER CRAY MAPS FROM SINGLE CHANNEL
      9 ...C    * SUBSET FILES PRODUCED WITH DATA FROM DISK
     10 ...C    ************************************************************
     11 ...C
     12 ...       INTEGER LSTRT,LEND,LINCR,PSTRT,PEND,PINCR,ICHNUM,NUMCHN,NPTS,
     13 ...      1LS(8),LCNTR,INBUF(3320)
     14 ...       INTEGER CHAR(7,2),PIXEL(125),PIXEL2(125)
     15 ...       DATA CHAR/"M   ","X   ","0   ","/   ","+   ","*   ","    ","$   ",
     16 ...      1"=   ","    ","    ","    ","    ","    ","    "/
     17 ...C
     18 ...C    * READ & WRITE 8-LEVEL SET NUMBERS
     19 ...C
     20 ...       READ(5,501)(LS(I),I=1,8)
     21 ...       WRITE(6,503)(LS(I),I=1,8)
     22 ...  501 FORMAT(8I4)
     23 ...  503 FORMAT(' * 8-LEVEL SET NUMBERS ARE: ',8I6)
     24 ...       WRITE(6,301)
     25 ...       REWIND 11
     26 ...C
     27 ...C    * READ HEADER INFO FROM INPUT UNIT
     28 ...C
     29 ...       READ(11,ERR=215,END=200)(INBUF(I),I=2,10)
     30 ...       LCNTR=0
     31 ...       NPTS=INBUF(3)
     32 ...       PSTRT=INBUF(5)
     33 ...       PEND=INBUF(6)
     34 ...       PINCR=INBUF(7)
```

```
      LSTRT=INBUF(8)
      LEND=INBUF(9)
      LINCR=INBUF(10)
      IEND=NPTS
C
C     * IF FILE IS TOO WIDE TO MAP, GENERATE MESWSAGE AND PROCEED W/INPUT
C     * TRUNCATED
C
      IF(NPTS.GT.125)GO TO 150
   18 WRITE(6,309)
      WRITE(6,415)
  415 FORMAT('1')
C
C     * PRINT MAP HEADER INFORMATION ON MAP OUTPUT CHANNEL
C
      WRITE(6,401)INBUF(4),PSTRT,PEND,PINCR,LSTRT,LEND,LINCR,LS
C
C     * READ LANDSAT ANNOTATION INFO. FOR PRINTING
C
      READ(11,ERR=215,END=200)(INBUF(I),I=1,144)
      WRITE(6,403)(INBUF(I),I=1,125)
      DO 30 I=1,10
      WRITE(6,409)(CHAR(J,1),J=1,7),(CHAR(J,2),J=1,7)
   30 CONTINUE
      WRITE(6,405)
C
C     * INITIALIZE OUTPUT BUFFERS
C
      DO 19 IA=1,125
      PIXEL(IA)=CHAR(7,2)
      PIXEL2(IA)=CHAR(7,2)
   19 CONTINUE
C
C     * READ A SCAN LINE FOR MAPPING
C
   20 READ(11,ERR=235,END=225)LNUM,(INBUF(I),I=1,NPTS)
```

```
 71 ...        DO 24 IB=1,IEND
 72 ...        DO 22 IA=1,7
 73 ...C
 74 ...C     * TEST VALUE OF EACH PIXEL AGAINST THE LS VECTOR. ASSIGN EACH  PIXEL
 75 ...C
 76 ...        IF(INBUF(IB).GT.LS(IA).AND.INBUF(IB).LE.LS(IA+1))GO TO 23
 77 ...    22 CONTINUE
 78 ...        GO TO 24
 79 ...C
 80 ...C     * A CHARACTER FOR PRINTING
 81 ...C
 82 ...    23 PIXEL(IB)=CHAR(IA,1)
 83 ...       PIXEL2(IB)=CHAR(IA,2)
 84 ...    24 CONTINUE
 85 ...C
 86 ...C     * PRINT AN OUT LINE OF MAP INFORMATION
 87 ...C
 88 ...        WRITE(6,407)LNUM,PIXEL,PIXEL2
 89 ...        LCNTR=LCNTR+1
 90 ...C
 91 ...C     * IF MORE DATA GO BACK AND DO ANOTHER LINE, ELSE FINISH UP
 92 ...C
 93 ...        GO TO 20
 94 ...C
 95 ...C     * SEMI-DIAGNOSTIC ERROR MESSAGE STACK
 96 ...C
 97 ...   150 WRITE(6,321)
 98 ...        WRITE(6,323)
 99 ...        IEND=125
100 ...        GO TO 18
101 ...   200 WRITE(6,311)
102 ...        WRITE(6,313)
103 ...        STOP
104 ...   215 WRITE(6,315)
105 ...        STOP
106 ...   225 WRITE(6,411)
107 ...        WRITE(6,307)
108 ...        WRITE(6,317)LNUM,LCNTR
```

```
      STOP
  235 WRITE(1,319)
      WRITE(1,317)LNUM,LCNTR
      GO TO 20
C
C   * WRITE OUTPUT FORMATS
C
  301 FORMAT(' * LIGMALS LINE PRINTER GRAY MAPPING SYSTEM',/,' * READY T
     1O PROCEED')
  307 FORMAT(' **********',/,' * AN END OF FILE')
  309 FORMAT(' * IMAGE NOW BEING COMPILED',/,' * STAND BY')
  311 FORMAT(' ***ERROR***',/,' * AN UNEXPECTED END OF FILE HAS BEEN ENC
     1OUNTERED')
  313 FORMAT(' * ERROR OCCURED DURING ATTEMPT TO READ HEADER INFORMATION
     1',/,' * VIDEO OUTPUT ATTEMPT ABORTED')
  315 FORMAT(' ***ERROR***',/,' * A FATAL READ ERROR HAS OCCURED ON VIDE
     1O INPUT',/,' * CONDITION IS NON-RECOVERABLE.')
  317 FORMAT(' * VIDEO COMPILATION TERMINATED AT LINE ',I5,' AFTER ',I5,
     1' LINES WERE PROCESSED')
  319 FORMAT(' ***ERROR***',/,' * A FATAL READ ERROR HAS OCCURED ON VIDE
     1O INPUT')
  321 FORMAT(' ***WARNING***',/,' * NUMBER OF PIXELS PER LINE EXCEEDS MA
     1XIMUM ALLOWABLE PICTURE WIDTH',/,' * TRUNCATION OF IMAGE AFTER FIR
     2ST 125 POINTS IN EACH LINE WILL OCCUR')
  323 FORMAT(' * VIDEO OUTPUT ABORTED')
  401 FORMAT(/,'  CU-LIGMALS',
     1   //,' LANDSAT INTERACTIVE GRAYMAP AND LEVEL SLICE SYSTEM',///,
     2' PROJECT LIGMALS-   7 LEVEL LINE PRINTER GRAY MAP',/,'  MAP OF CH
     3ANNEL# ',I5,/,' POINTS ',I5,' TO ',I5,' EVERY ',I5,' POINTS',/,
     4' LINES ',I5,' TO ',I5,' EVERY ',I5,' LINES',/,' LEVEL SEL CONSTAN
     5TS= ',8(I5,','))
  403 FORMAT(/,3X,125A1,/)
  405 FORMAT('1',   131('*'),//,7X,25('+',4X),///)
  407 FORMAT(  1X,  I5,1X,125A1,/,'+',6X,125A1)
  409 FORMAT(10X,7A1,/,'+',9X,7A1)
  411 FORMAT(///,7X,25('+',4X),//,1X,131('*')////)
      END
END OF FILE        146 CARDS
```

```
CARD NO   FILE: CARDS                (06/0) EBCDIC                    5/07/79

      1 ...IDENT IMAGE
      2 ...FILE  9=REFORM ,UNIT=TAPE,RECORD=9964,BLOCKING=1,FIXED,BUFFERS=1
      3 ...C
      4 ...C
      5 ...C      SUBROUTINE TSTATS
      6 ...C
      7 ...C    ************************************************************************
      8 ...C    * TRAINING SET ANALYSIS PROGRAM                                        *
      9 ...C    * ROUTINE TO TAKE THE CO-LIGMALS REFORMATTED MSS DATA TAPE             *
     10 ...C    * ASSUME DATA OF EACH TRAINING AREA IS MULTIVARIATE NORMAL DIST AND*
     11 ...C    * USE MAXIMUM LIKELIHOOD ESTIMATION METHOD TO DERIVE MEAN VECTORS  *
     12 ...C    * & COVARIANGE MATRICES                                               *
     13 ...C    * THIS PROGRAM REQUIRES LINE-POINT COORDINATES DECK INPUT FROM THE *
     14 ...C    * USER                                                                *
     15 ...C    * THESE ESTIMATERS WILL BE USED FOR CLASSIFICATION                    *
     16 ...C    *                                                                     *
     17 ...C    ************************************************************************
     18 ...C
     19 ...C
     20 ...C    * ASSIGN VARIABLES AND DECLARE STORAGE AREAS
     21 ...C
     22 ...       INTEGER LLSTRT(20),LLEND(20),LLINCR(20),PPSTRT(20),PPEND(20),PPINC
     23 ...      1R(20),NCHN,ICHNUM(4),NUMCHN,INBUF(3320),NSET,BUF(40),TPTS,ISET(20)
     24 ...       INTEGER L(4),M(4)
     25 ...       REAL PCNTR(4),SQPCN(4,4),MEAN(4),COV(4,4),A(2,2),
     26 ...      1IDENT(4,4),CLNAME(20)
     27 ...       IUNIT=9
     28 ...       WRITE(6,303)
     29 ...       REWIND IUNIT
     30 ...C
     31 ...C    * READ HEADER INFORMATION
     32 ...C
     33 ...       READ(IUNIT,501,END=205,ERR=215)BUF,(INBUF(I),I=2,3)
     34 ...       NUMCHN=4
```

```
      INBUF(3)=INBUF(3)/NUMCHN
C
C  * READ NUMBER OF CHANNEL AND THE CHANNEL NO. TO BE USED
C  * AND THE NUMBER OF TRAINING AREA FROM CARDS
C
      READ(5,401)NCHN,(ICHNUM(I),I=1,NCHN),NSET,NCARD
      WRITE(6,401)NCHN,(ICHNUM(I),I=1,NCHN),NSET,NCARD
C
C  * READ CLASS NAME FROM A CARD
C
      READ(5,403)(CLNAME(I),I=1,NSET)
      WRITE(6,403)(CLNAME(I),I=1,NSET)
C
C  * TRAINING SETS MUST BE RECTRANGULAR, READ ALL LINE-POINT INCREMENTS
C  * FROM CARDS.    THE NO. OF CARDS TO BE READ DEPENDS UPON NO. OF
C  * TRAINING SETS TO BE ANALYSED
C
      DO 100 I=1,NCARD
      READ(5,405)LLSTRT(I),LLEND(I),LLINCR(I),PPSTRT(I),PPEND(I),PPINCR(
     1I),ISET(I)
      WRITE(6,405)LLSTRT(I),LLEND(I),LLINCR(I),PPSTRT(I),PPEND(I),PPINCR
     1(I)
      IF(PPEND(I).LE.INBUF(3))GO TO 100
      WRITE(6,307)PPEND(I),INBUF(3)
      PPEND(I)=INBUF(3)
      WRITE(6,309)INBUF(3)
  100 CONTINUE
C
C  * READ ANNOTATION BLOCK DATA
C
      READ(IUNIT,503,END=205,ERR=215)(INBUF(I),I=1,144)
      WRITE(6,411)
      II=1
C
C  * INITIALIZE THE VARIABLES TO BE USED IN THE COMPUTATION TO ZEROES
C
   10 TPTS=0
      DO 110 J=1,NCHN
      PCNTR(J)=0.0
```

```
      DO 110 K=1,J
      SQPCN(J,K)=0.0
  110 CONTINUE
      WRITE(6,412)CLNAME(ISET(II)
C
C     * COMPUTE THE LOCATION OF THE DESIRED POINTS WITHIN THE INPUT
C     * BUFFER AREA
C
   12 ISTRT=1+((PPSTRT(II)-1)*NUMCHN)
      IEND=NUMCHN*PPEND(II)
      IINCR=(NUMCHN*PPINCR(II))
      LTST=LLSTRT(II)
      LEND=LLEND(II)
      LINCR=LLINCR(II)
C
C     READ A LINE OF  DATA. COUNT TOTAL POINTS USED, ADD POINT VALUE TO
C     ACCUMULATORS PCNTR & SQPCN
C
      READ(IUNIT,505,END=225,ERR=235)LNUM,INBUF
      LSKIP=LTST-LNUM-1
      IF(LSKIP.LT.-1) GO TO 18
      IF(LSKIP.LE.1)GO TO 25
      DO 115 I=1,LSKIP
      READ(IUNIT,505)
  115 CONTINUE
      GO TO 20
   15 LBACK=LNUM-LTST+1
      DO 18 I=1,LBACK
      BACKSPACE IUNIT
   18 CONTINUE
   20 READ(IUNIT,505,END=225,ERR=235)LNUM,INBUF
   25 IF(LNUM.LT.LTST)GO TO 20
      WRITE(6,423)(INBUF(I),INBUF(I+1),INBUF(I+2),INBUF(I+3),I=ISTRT,
     1IEND,IINCR)
      DO 140 I=ISTRT,IEND,IINCR
      DO 130 JJ=1,NCHN
      J=ICHNUM(JJ)
```

```
      IF(INBUF(I+J-1).LT.-511.OR.INBUF(I+J-1).GT.4096)GO TO 140
      PCNTR(JJ)=PCNTR(JJ)+INBUF(I+J-1)
      WRITE(6,603)PCNTR(JJ),JJ,J,INBUF(I+J-1)
  603 FORMAT(5X,F8.0,5X,3I4)
      DO 120 KK=1,JJ
      K=ICHNUM(KK)
      SQPCN(JJ,KK)=SQPCN(JJ,KK)+(INBUF(I+J-1)*INBUF(I+K-1))
      WRITE(6,601)SQPCN(JJ,KK),I,JJ,J,KK,K,INBUF(I+J-1),INBUF(I+K-1),
     1ISTRT,IEND,IINCR,NCHN,NSET
  601 FORMAT(35X,F8.0,5X,5I4,5X,2I4,5X,5I5)
  120 CONTINUE
  130 CONTINUE
C
C   * INCREMENT POINTS COUNTER
C
      TPTS=TPTS+1
  140 CONTINUE
C
C   * STATISTICS FOR THIS LINE ARE FINISHED. GET NEXT LINE
C
      LLAST=LNUM
      LTST=LTST+LINCR
   30 IF(LTST.GT.LEND)GO TO 50
      IF(LLAST.GE.LTST)GO TO 40
      GO TO 20
   40 LTST=LTST+1
      GO TO 30
   50 II=II+1
      IF(ISET(II).EQ.ISET(II-1))GO TO 12
      WRITE(6,425)
      DO 145 I=1,NCHN
  145 WRITE(6,427)PCNTR(I),TPTS,(SQPCN(I,J),J=1,I)
C
C   * DETERMINE THE MEAN VECTOR & VARIANCE-COVARIANCE MATRIX
C
      DO 150 I=1,NCHN
      MEAN(I)=PCNTR(I)/TPTS
```

```
148 ...       DO 150 J=1,I
149 ...       C=SQPCN(I,J)-((PCNTR(I)*PCNTR(J))/TPTS)
150 ...       COV(I,J)=C/(TPTS-1)
151 ...       COV(J,I)=COV(I,J)
152 ...       A(I,J)=COV(I,J)
153 ...       A(J,I)=COV(I,J)
154 ...   150 CONTINUE
155 ...C
156 ...       CALL MINV(A,NCHN,D,L,M)
157 ...C
158 ...       DO 160 I=1,NCHN
159 ...       DO 160 J=1,NCHN
160 ...       IDENT(I,J)=0.0
161 ...       DO 160 K=1,NCHN
162 ...       IDENT(I,J)=IDENT(I,J)+(A(I,K)*COV(K,J))
163 ...   160 CONTINUE
164 ...       WRITE(6,413)
165 ...       DO 170 I=1,NCHN
166 ...   170 WRITE(6,415)MEAN(I),(COV(I,J),J=1,I)
167 ...       WRITE(6,417)
168 ...       DO 180 I=1,NCHN
169 ...   180 WRITE(6,419)(A(I,J),J=1,I)
170 ...       WRITE(6,418)
171 ...       DO 190 I=1,NCHN
172 ...   190 WRITE(6,419)(IDENT(I,J),J=1,I)
173 ...       WRITE(6,421)D
174 ...       IF(II.LE.NCARD)GO TO 10
175 ...       STOP
176 ...C
177 ...C    * ONLY DIAGNOSTIC ERROR TRAPS AND VARIOUS FORMAT STATEMENTS APPEAR
178 ...C    * BEYOND THIS POINT
179 ...C
180 ...   205 WRITE(6,311)
181 ...       WRITE(6,313)
182 ...       STOP
183 ...   215 WRITE(6,315)
184 ...       WRITE(6,313)
185 ...       STOP
186 ...   225 WRITE(6,311)
```

```
187 ...       WRITE(6,317)LNUM
188 ...       REWIND 9
189 ...       STOP
190 ...   235 WRITE(6,315)
191 ...       WRITE(6,317)LNUM
192 ...       GO TO 20
193 ...   301 FORMAT(43X,'STATISTICS',///,10X,'CLASS NAME',10X,'MEAN VECTOR',20X
194 ...      1,'VARIANCE-COVARIANCE MATRIX'///)
195 ...   303 FORMAT(' * READY TO PROCEED',/,' REWINDING INPUT TAPE - STAND BY',
196 ...      1///)
197 ...   307 FORMAT(' ***ERROR***',/,' * NUMBER OF DATA POINTS REQUESTED EXCEED
198 ...      1S DATA BASE AVAILABLE',/,' * POINT # ',I5,' WAS LAST POINT REQUEST
199 ...      2ED,',/,' * BUT DATA SET CONSTAINS ONLY',I5,' POINTS',/,' * TRAININ
200 ...      3G SET #',I5)
201 ...   309 FORMAT(' * TRUNCATED AT POINT #',I5)
202 ...   311 FORMAT(' ***ERROR***',/,' * AN UNEXPECTED END OF FILE HAS BEEN ENC
203 ...      1OUNTERED')
204 ...   313 FORMAT(' * ERROR OCCURED DURING ATTEMPT TO READ HEADER INFORMATION
205 ...      1')
206 ...   315 FORMAT(' ***ERROR***',/,' * A FATAL READ ERROR HAS OCCURED ON VIDE
207 ...      1O INPUT')
208 ...   317 FORMAT(' * ERROR OCCURED DURING AN ATTEMPT TO READ DATA LINE#',I5)
209 ...   401 FORMAT(7I2)
210 ...   403 FORMAT(15A5)
211 ...   405 FORMAT(7I5)
212 ...   411 FORMAT(40X,'STATISTICS',///)
213 ...   412 FORMAT(43X,A6,//)
214 ...   413 FORMAT(//,10X,'MEAN VECTOR',43X,'COVARIANCE MATRIX',/)
215 ...   415 FORMAT(10X,F10.5,43X,4(F10.5,2X))
216 ...   417 FORMAT(//,10X,'INVERSE MATRIX',/)
217 ...   418 FORMAT(//,10X,'IDENTITY MATRIX',/)
218 ...   419 FORMAT(10X,4(F10.5,2X))
219 ...   421 FORMAT(// ,10X,'DET =',F10.5,///)
220 ...   423 FORMAT(10(4I3,1X))
221 ...   425 FORMAT(//,10X,'SUMX',29X,'TPTS',17X,'SUMXY',/)
222 ...   427 FORMAT(8X,F7.0,26X,I6,14X,4(F8.0))
223 ...   501 FORMAT(40A1,2I4)
224 ...   503 FORMAT(144A1)
225 ...   505 FORMAT(I4,10(332I3))
226 ...       END
```

```
      SUBROUTINE MINV(A,N,D,L,M)
C
C***********************************************************************
C*
C* THE SUBROUTINE MINV
C*
C* PURPOSE
C*    INVERSE A MATRIX
C*
C* USAGE
C*    CALL MINV(A,N,D,L,M)
C*
C* DESCRIPTION OF PARAMETERS
C*    A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C*        RESULTANT INVERSE
C*    N - ORDER OF MATRIX A
C*    D - RESULTANT DETERMINANT
C*    L - WORK VECTOR OF LENGTH N
C*    M - WORK VECTOR OF LENGTH N
C*
C* REMARK
C*    MATRIX A MUST BE A GENERAL MATRIX
C*
C* METHOD
C*    THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT IS
C*    ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT THE
C*    MATRIX IS SINGULAR.
C*
C***********************************************************************
C*
C*
C***********************************************************************
C
C    SEARCH FOR LARGEST ELEMENT
C
      DIMENSION A(1),L(1),M(1)
      D=1.0
      NK=-N
      DO 100 K=1,N
```

```
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(ABS(BIGA)-ABS(A(IJ)))15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C     INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K)35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI)=HOLD
C
C     INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K)50,50,40
   40 JP=N*(I-1)
      DO 45 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   45 A(JI)=HOLD
```

```
C
C     DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT  IS IN BIGA)
C
   50 IF(BIGA)60,55,60
   55 D=0.0
      RETURN
   60 DO 70 I=1,N
      IF(I-K)65,70,65
   65 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   70 CONTINUE
C
C     REDUCE MATRIX
C
      DO 85 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 85 J=1,N
      IJ=IJ+N
      IF(I-K)75,85,75
   75 IF(J-K)80,85,80
   80 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   85 CONTINUE
C
C     DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 95 J=1,N
      KJ=KJ+N
      IF(J-K)90,95,90
   90 A(KJ)=A(KJ)/BIGA
   95 CONTINUE
C
C     PRODUCT OF PIVOTS
C
```

```
    341 ...         D=D*BIGA
    342 ...C
    343 ...C        REPLACE PIVOT BY RECIPROCAL
    344 ...C
    345 ...         A(KK)=1.0/BIGA
    346 ...  100 CONTINUE
    347 ...C
    348 ...C        FINAL ROW AND COLUMN INTERCHANGE
    349 ...C
    350 ...         K=N
    351 ...  105 K=(K-1)
    352 ...         IF(K)140,140,110
    353 ...  110 I=L(K)
    354 ...         IF(I-K)125,125,115
    355 ...  115 JQ=N*(K-1)
    356 ...         JR=N*(I-1)
    357 ...         DO 120 J=1,N
    358 ...         JK=JQ+J
    359 ...         HOLD=A(JK)
    360 ...         JI=JR+J
    361 ...         A(JK)=-A(JI)
    362 ...  120 A(JI)=HOLD
    363 ...  125 J=M(K)
    364 ...         IF(J-K)105,105,130
    365 ...  130 KI=K-N
    366 ...         DO 135 I=1,N
    367 ...         KI=KI+N
    368 ...         HOLD=A(KI)
    369 ...         JI=KI-K+J
    370 ...         A(KI)=-A(JI)
    371 ...  135 A(JI)=HOLD
    372 ...         GO TO 105
    373 ...  140 RETURN
    374 ...         END
END OF FILE          374 CARDS
```

```
CARD NO    FILE: CARDS                    (06/0) EBCDIC                    5/07/79
     1 ...IDENT IMAGE
     2 ...FILE  9=REFORM ,UNIT=TAPE,RECORD=9964,BLOCKING=1,FIXED,BUFFERS=1
     3 ...C
     4 ...C       SUBROUTINE HSTGRM
     5 ...C
     6 ...C     ***************************************************************************
     7 ...C     * ROUTINE TO GENERATE A PRINTER PLOT OF A HISTOGRAM OF THE          *
     8 ...C     * BRIGHTNESS VALUES IN DATA SET                                    *
     9 ...C     ***************************************************************************
    10 ...C
    11 ...C     * ASSIGN VARIABLES AND DECLARE STORAGE AREAS
    12 ...C
    13 ...       INTEGER LLSTRT(20),LLEND(20),LLINCR(20),PPSTRT(20),PPEND(20),
    14 ...      1PPINCR(20),INBUF(3320), BUF(40),ICHNUM(4),ISET(20)
    15 ...       INTEGER PIXEL(101),BL/'    '/,AXIS/'+   '/,AS/'*    '/,OVER/'$   '/
    16 ...       REAL CNT(4,100),NCNT(4),PCNTR(4),ACUM,CLNAME(20)
    17 ...       IUNIT=9
    18 ...       WRITE(6,303)
    19 ...       REWIND IUNIT
    20 ...C
    21 ...C     * READ HEADER INFORMATION
    22 ...C
    23 ...       READ(IUNIT,501,END=205,ERR=215)BUF,(INBUF(I),I=2,3)
    24 ...       NUMCHN=4
    25 ...       INBUF(3)=INBUF(3)/NUMCHN
    26 ...C
    27 ...C     * READ NUMBER OF CHANNEL AND THE CHANNEL NO. TO BE USED
    28 ...C     * AND THE NUMBER OF TRAINING AREA FROM CARDS
    29 ...C
    30 ...       READ(5,401)NCHN,(ICHNUM(I),I=1,NCHN),NSET,NCARD
    31 ...       WRITE(6,401)NCHN,(ICHNUM(I),I=1,NCHN),NSET,NCARD
    32 ...C
    33 ...C     * READ CLASS NAME FROM A CARD
    34 ...C
```

```
      READ(5,403)(CLNAME(I),I=1,NSET)
      WRITE(6,403)(CLNAME(I),I=1,NSET)
C
C     * TRAINING SETS MUST BE RECTRANGULAR, READ ALL LINE-POINT INCREMENTS
C     * FROM CARDS.    THE NO. OF CARDS TO BE READ DEPENDS UPON NO. OF
C     * TRAINING SETS TO BE ANALYSED
C
      DO 100 I=1,NCARD
      READ(5,405)LLSTRT(I),LLEND(I),LLINCR(I),PPSTRT(I),PPEND(I),PPINCR(
     1I),ISET(I)
      WRITE(6,405)LLSTRT(I),LLEND(I),LLINCR(I),PPSTRT(I),PPEND(I),PPINCR
     1(I)
      IF(PPEND(I).LE.INBUF(3))GO TO 100
      WRITE(6,307)PPEND(I),INBUF(3)
      PPEND(I)=INBUF(3)
      WRITE(6,309)INBUF(3)
  100 CONTINUE
C
C     * READ ANNOTATION BLOCK DATA
C
      READ(IUNIT,503,END=205,ERR=215)(INBUF(I),I=1,144)
      WRITE(6,301)
      II=1
C
C     * INITIALIZE THE VARIABLES TO BE USED IN THE COMPUTATION TO ZEROES
C
   10 DO 110 J=1,NCHN
      PCNTR(J)=0.0
      NCNT(J)=0.0
      DO 110 K=1,100
      CNT(J,K)=0.0
  110 CONTINUE
      PIXEL(1)=AXIS
      DO 12 IA=2,101
      PIXEL(IA)=BL
   12 CONTINUE
```

```
C
C    * COMPUTE THE LOCATION OF THE DESIRED POINTS WITHIN THE INPUT
C    * BUFFER AREA
C
   14 ISTRT=1+((PPSTRT(II)-1)*NUMCHN)
      IEND=NUMCHN*PPEND(II)
      IINCR=(NUMCHN*PPINCR(II))
      LTST=LLSTRT(II)
      LEND=LLEND(II)
      LINCR=LLINCR(II)
C
C     READ A LINE OF  DATA, COUNT TOTAL POINTS USED, ADD POINT VALUE TO
C     ACCUMULATORS PCNTR , CNT & NCNT
C
      READ(IUNIT,505,END=225,ERR=235)LNUM,INBUF
      LSKIP=LTST-LNUM-1
      IF(LSKIP.LT.-1)GO TO 16
      IF(LSKIP.LE.1)GO TO 25
      DO 115 I=1,LSKIP
      READ(IUNIT,505)
  115 CONTINUE
      GO TO 20
   16 LBACK=LNUM-LTST+1
      DO 120 I=1,LBACK
      BACKSPACE IUNIT
  120 CONTINUE
   20 READ(IUNIT,505,END=225,ERR=235)LNUM,INBUF
   25 IF(LNUM.LT.LTST)GO TO 20
      DO 140 I=ISTRT,IEND,IINCR
      DO 130 JJ=1,NCHN
      PCNTR(JJ)=PCNTR(JJ)+1
      J=ICHNUM(JJ)
C
C    * CHECK TO SEE WHETHER POINT IS OUT OF RANGE IS SO, BRANCH TO
C    * HANDLE IT ELSEWHERE  IF NOT, COUNT IN THE APPROPRIATE BIN.
C
```

```
107 ...       IF(INBUF(I+J-1).LE.0.OR.INBUF(I+J-1).GT.100)GO TO 26
108 ...       CNT(JJ,INBUF(I+J-1))=CNT(JJ,INBUF(I+J-1))+1
109 ...       GO TO 130
110 ...    26 NCNT(JJ)=NCNT(JJ)+1
111 ...   130 CONTINUE
112 ...   140 CONTINUE
113 ...C
114 ...C    * CALCUTORS  FOR THIS LINE ARE FINISHED. GET NEXT LINE
115 ...C
116 ...       LLAST=LNUM
117 ...       LTST=LTST+LINCR
118 ...    30 IF(LTST.GT.LEND)GO TO 50
119 ...       IF(LLAST.GE.LTST)GO TO 40
120 ...       GO TO 20
121 ...    40 LTST=LTST+1
122 ...       GO TO 30
123 ...C
124 ...C    * POINT COUNTING IS OVER, PRINT AXES ON OUTPUT UNIT AND
125 ...C    * GENERATE HISTOGRAM.
126 ...C
127 ...    50 II=II+1
128 ...       IF(ISET(II).EQ.ISET(II-1))GO TO 14
129 ...       DO 150 J=1,NCHN
130 ...       ACUM=0.0
131 ...       WRITE(6,411)PCNTR(J),NCNT(J)
132 ...       DO 145 K=1,100
133 ...       IF(ACUM.GT.99.0)GO TO 145
134 ...C
135 ...C    * COMPUTE THE LENGTH OF THE LINE NEEDED FOR EACH BIN AND
136 ...C    * GENERATE THE APPROPRIATE OUTPUT
137 ...C
138 ...       CNT(J,K)=(CNT(J,K)/PCNTR(J))*100.
139 ...       IAD=CNT(J,K)+1
140 ...       IF(CNT(J,K).EQ.0.0)GO TO 56
141 ...       IF(IAD.GT.101)IAD=101
```

```
      DO 54 IA=2,IAD
      PIXEL(IA)=AS
   54 CONTINUE
C
C  * ACCUMULATE THE TOTAL PERCENTAGE OF POINTS
C
   56 ACUM=ACUM+CNT(J,K)
      WRITE(6,413)K,CNT(J,K),ACUM,PIXEL
      DO 58 IA=2,IAD
      PIXEL(IA)=BL
   58 CONTINUE
  145 CONTINUE
      WRITE(6,412)CLNAME(ISET(II)),ICHNUM(J)
  150 CONTINUE
      IF(II.LE.NCARD)GO TO 10
      WRITE(6,415)
      STOP
C
C  * ONLY DIAGNOSTIC ERROR TRAPS AND VARIOUS FORMAT STATEMENTS APPEAR
C  * BEYOND THIS POINT
C
  205 WRITE(6,311)
      WRITE(6,313)
      STOP
  215 WRITE(6,315)
      WRITE(6,313)
      STOP
  225 WRITE(6,311)
      WRITE(6,317)LNUM
      REWIND 9
      STOP
  235 WRITE(6,315)
      WRITE(6,317)LNUM
      GO TO 20
  301 FORMAT(' * LIGMALS HISTOGRAM PLOTTING ROUTINE')
```

```
177 ...   303 FORMAT(' * READY TO PROCEED',/,' REWINDING INPUT TAPE - STAND BY',
178 ...      1///)
179 ...   307 FORMAT(' ***ERROR***',/,' * NUMBER OF DATA POINTS REQUESTED EXCEED
180 ...      1S DATA BASE AVAILABLE',/,' * POINT # ',I5,' WAS LAST POINT REQUEST
181 ...      2ED,',/,' * BUT DATA SET CONSTAINS ONLY',I5,' POINTS',/,' * TRAININ
182 ...      3G SET #',I5)
183 ...   309 FORMAT(' * TRUNCATED AT POINT #',I5)
184 ...   311 &ORMAT(' ***ERROR***',/,' * AN UNEXPECTED END OF FILE HAS BEEN ENC
185 ...      1OUNTERED')
186 ...   313 FORMAT(' * ERROR OCCURED DURING ATTEMPT TO READ HEADER INFORMATION
187 ...      1')
188 ...   315 FORMAT(' ***ERROR***',/,' * A FATAL READ ERROR HAS OCCURED ON VIDE
189 ...      1O INPUT')
190 ...   317 FORMAT(' * ERROR OCCURED DURING AN ATTEMPT TO READ DATA LINE#',I5)
191 ...   401 FORMAT(6I2)
192 ...   403 FORMAT(15A5)
193 ...   405 FORMAT(6I5)
194 ...   411 FORMAT('1',' * LIGMALS HISTOGRAM PLOT',//,'TOTAL # OF POINTS USED=
195 ...      1',F7.0,15X,'TOTAL # OF POINTS OUT OF RANGE',2X,F7.0,///,3X,'VALUE'
196 ...      2,3X,'PCNT',3X,'CUM. PCNT',///,33X,'10%', 7X,'20%', 7X,'30%', 7X,'4
197 ...      30%', 7X,'50%',2(/,33X,9('+',9X)),/,23X,101('+'),2(/,23X,10('+',9X)
198 ...      4))
199 ...   412 FORMAT(////,33X,A5,5X,'CHANNEL #',I4)
200 ...   413 FORMAT(3X,I4,1X,F6.2,1X,F6.2,2X,101A1)
201 ...   415 FORMAT(//,' * PLOT COMPLETED.')
202 ...   501 FORMAT(40A1,2I4)
203 ...   503 FORMAT(144A1)
204 ...   505 FORMAT(I4,10(332I3))
205 ...       END
END OF FILE        205 CARDS
```

```
CARD NO   FILE: CARDS                (06/0) EBCDIC                    5/07/79

      1 ...IDENT IMAGE
      2 ...C
      3 ...FILE  1=DTAP11 ,UNIT=DISK,RECORD=670,BLOCKING=10,AREA=300,RANDOM,LOCK
      4 ...FILE  2=DTAP21 ,UNIT=DISK,RECORD=670,BLOCKING=10,AREA=300,RANDOM,LOCK
      5 ...FILE  6=OPRINT,UNIT=PRINTER
      6 ...FILE  8=MSG,UNIT=PRINTER
      7 ...C
      8 ...C    * SUBROUTINE CLASSIFY                                          *
      9 ...C
     10 ...C    ********************************************************************
     11 ...C    * PROGRAM TO CLASSIFY CU-LIGMALS REFORMATED DATA OF DESIRED AREA   **
     12 ...C    * THIS PROGRAM REQUIRES STATISTICS INPUT FROM PREVIOUS RUN OF      **
     13 ...C    * PROGRAM TSTATS, THEY ARE MEAN VECTOR, SIGMA INVERSE MATRIX AND   *
     14 ...C    * DETERMINANT OF EACH CLASS                                       **
     15 ...C    *                                                                 **
     16 ...C    * THE PROGRAM WILL GENERAT DISCRIMINANT AUNCTION FOR EACH CLASS    *
     17 ...C    * THEN SELECTS THE LARGEST VALUE TO ASSIGN PATTERN                 *
     18 ...C    *                                                                 **
     19 ...C    * THE MAXIMUM LIKELIHOOD CLASSIFICATION IS BASED ON ASSUMPTION THAT*
     20 ...C    * THE DISTRIBUTION OF ALL DATA OF EVERY CHANNEL IS MULTIVARIATE    *
     21 ...C    * NORMAL DISTRIBUTION                                             **
     22 ...C    *                                                                 **
     23 ...C    ********************************************************************
     24 ...C
     25 ...C    * ASSIGN VARIABLES AND DELCARE STORAGE AREAS
     26 ...C
     27 ...       INTEGER LSTRT,LEND,LINCR,PSTRT,PEND,PINCR,NPTS,LCNTR,INBFA(150),
     28 ...      1INBFB(150),INBFC(150),INBFD(150),NCL,NCHN,ICHNUM(4)
     29 ...      2,ICOUNT(10),HEAD(2,80),CLNAME(10,15)
     30 ...       REAL X(4),MEAN(4,10),SIGINV(4,4,10),DFT(10),PROD(4),DISCF
     31 ...      1,AREA(10)
     32 ...       INTEGER CHAR(15),PIXEL(125),BLANK/"    "/
     33 ...       WRITE(6,301)
     34 ...       READ(5,501)NCHN,(ICHNUM(I),I=1,NCHN),NCL
```

```
      READ(5,500)(CHAR(I),I=1,NCL)
      READ(5,600)AREAP
C
C     * READ AND WRITE ALL INPUT STATISTICS FROM DATA CARDS
C
      WRITE(6,502)
      DO 13 J=1,NCL
      DO 12 I=1,NCHN
      READ(5,503)MEAN(I,J),(SIGINV(I,K,J),K=1,NCHN)
      WRITE(6,504)MEAN(I,J),(SIGINV(I,K,J),K=1,NCHN)
   12 CONTINUE
      WRITE(6,506)
   13 CONTINUE
      READ(5,505)(DET(I),I=1,NCL)
      WRITE(6,508)(DET(I),I=1,NCL)
      DO 14 I=1,2
      READ(5,507)(HEAD(I,J),J=1,80)
   14 CONTINUE
      DO 15 I=1,NCL
      READ(5,509)(CLNAME(I,J),J=1,15)
   15 CONTINUE
C
C     * READ INDIVIDUAL SUBIMAGE
C
      READ(1'2341,END=225,ERR=235)(INBFA(I),I=2,10)
      READ(2'2341,END=226,ERR=236)(INBFB(I),I=2,10)
C     READ(3'2341,END=227,ERR=237)(INBFC(I),I=2,10)
C     READ(4'2341,END=228,ERR=238)(INBFD(I),I=2,10)
      IF(INBFA(3).NE.INBFB(3))GO TO 245
      DO 16 I=5,10
      IF(INBFA(I).NE.INBFB(I))GO TO 245
   16 CONTINUE
C
C     * SET PARAMETERS FOR HEADING OUTPUT
C
      NPTS=INBFA(3)
```

```
      PSTRT=INBFA(5)
      PEND=INBFA(6)
      PINCR=INBFA(7)
      LSTRT=INBFA(8)
      LEND=INBFA(9)
      LINCR=INBFA(10)
      IEND=NPTS
      IF(NPTS.LE.125)GO TO 17
      WRITE(6,321)
      IEND=125
   17 WRITE(6,303)
      WRITE(6,415)
  415 FORMAT('1')
      WRITE(6,401)(ICHNUM(I),I=1,NCHN)
      WRITE(6,402)PSTRT,PEND,PINCR,LSTRT,LEND,LINCR
C
C     * READ ANNOTATION BLOCK DATA AND WRITE FOR HEADING OUTPUT
C
      READ(1'2342,END=225,ERR=235)(INBFA(I),I=1,144)
      READ(2'2342,END=226,ERR=236)(INBFB(I),I=1,144)
C     READ(3'2342,END=227,ERR=237)(INBFC(I),I=1,144)
C     READ(4'2342,END=228,ERR=238)(INBFD(I),I=1,144)
      DO 18 I=1,2
      WRITE(6,404)(HEAD(I,J),J=1,80)
   18 CONTINUE
      WRITE(6,406)CHAR(1),(CLNAME(1,J),J=1,15)
      DO 19 I=2,NCL
      WRITE(6,408)CHAR(I),(CLNAME(I,J),J=1,15)
   19 CONTINUE
C
C     * INITIALIZE PIXEL COUNTERS TO ZERO
C
      DO 21 I=1,NCL
      ICOUNT(I)=0
   21 CONTINUE
      WRITE(6,405)
```

```
107 ...C
108 ...C    * SET INITIALIZE AREA TO BLANK
109 ...C
110 ...       DO 20 IB=1,125
111 ...       PIXEL(IB)=BLANK
112 ...    20 CONTINUE
113 ...C
114 ...C    * INITIALIZE FOR LINE COUNTER
115 ...C
116 ...       LCNTR=0
117 ...C
118 ...C    * READ A LINE OF DATA OF ALL CHANNELS WE USED
119 ...C
120 ...       IREC=LSTRT
121 ...    22 READ(1'IREC,END=255,ERR=265)LNUMA,(INBFA(I),I=1,NPTS)
122 ...       READ(2'IREC,END=255,ERR=266)LNUMB,(INBFB(I),I=1,NPTS)
123 ...C      READ(3'IREC,END=255,ERR=267)LNUMC,(INBFC(I),I=1,NPTS)
124 ...C      READ(4'IREC,END=255,ERR=268)LNUMD,(INBFD(I),I=1,NPTS)
125 ...       IF(IREC.EQ.LNUMA)GO TO 23
126 ...       IREC=IREC+LINCR
127 ...       GO TO 22
128 ...C
129 ...C    * COMPARE LINE NUMBERS OF EACH CHANNEL, GO ON EXECUTION IF THEY ARE
130 ...C    * EQUAL
131 ...C
132 ...    23 IF(LNUMA.NE.LNUMB)GO TO 275
133 ...       DO 30 IB=1,IEND
134 ...       X(1)=INBFA(IB)
135 ...       X(2)=INBFB(IB)
136 ...C      X(3)=INBFC(IB)
137 ...C      X(4)=INBFD(IB)
138 ...C
139 ...C      SET INITIAL VALUE TO BE COMPARED IN SUCCEEDING INSTRUCTIONS
140 ...C
141 ...       GMAX=-32767.
142 ...       DO 28 I=1,NCL
```

```
143 ...        DISCF=(-ALOG(ABS(DET(I))))/2.
144 ...        DO 26 J=1,NCHN
145 ...C
146 ...C     * GENERATE THE DISCRIMINANTE FUNCTION
147 ...C
148 ...        PROD(J)=0.
149 ...        DO 24 K=1,NCHN
150 ...        PROD(J)=PROD(J)+(X(K)-MEAN(K,I))*SIGINV(K,J,I)
151 ...     24 CONTINUE
152 ...        DISCF=DISCF-PROD(J)*(X(J)-MEAN(J,I))*.5
153 ...     26 CONTINUE
154 ...C
155 ...C     * SELECT THE MAXIMUM DISCRIMINANT
156 ...C
157 ...        IF(DISCF.LE.GMAX)GO TO 28
158 ...        GMAX=DISCF
159 ...        IA=I
160 ...     28 CONTINUE
161 ...C
162 ...C     * ASSIGN ITS PATTERN
163 ...C
164 ...        PIXEL(IB)=CHAR(IA)
165 ...        ICOUNT(IA)=ICOUNT(IA)+1
166 ...     30 CONTINUE
167 ...        WRITE(6,407)LNUMA,PIXEL
168 ...C
169 ...C     * INCREMENT LINE COUNTER AND GET NEXT LINE
170 ...C
171 ...        LCNTR=LCNTR+1
172 ...        IREC=IREC+LINCR
173 ...        IF(IREC.GT.LEND)GO TO 255
174 ...        GO TO 22
175 ...C
176 ...C     * SEMI-DIAGNOSTIC ERROR MASSAGE STACK
177 ...C
178 ...   225 WRITE(6,309)
```

```
179 ...       WRITE(6,311)
180 ...       STOP
181 ...   226 WRITE(6,309)
182 ...       WRITE(6,313)
183 ...       STOP
184 ...   227 WRITE(6,309)
185 ...       WRITE(6,315)
186 ...       STOP
187 ...   228 WRITE(6,309)
188 ...       WRITE(6,317)
189 ...       STOP
190 ...   235 WRITE(6,319)
191 ...       WRITE(6,311)
192 ...       STOP
193 ...   236 WRITE(6,319)
194 ...       WRITE(6,313)
195 ...       STOP
196 ...   237 WRITE(6,319)
197 ...       WRITE(6,315)
198 ...       STOP
199 ...   238 WRITE(6,319)
200 ...       WRITE(6,317)
201 ...       STOP
202 ...   245 WRITE(6,323)
203 ...       STOP
204 ...   255 WRITE(6,409)
205 ...       WRITE(6,306)
206 ...       ITOTAL=0
207 ...       NCL=NCL-1
208 ...       TAREA=0
209 ...       DO 32 I=1,NCL
210 ...       AREA(I)=AREAP*ICOUNT(I)
211 ...       WRITE(6,307)CHAR(I),ICOUNT(I),AREA(I)
212 ...       TAREA=TAREA+AREA(I)
213 ...       ITOTAL=ITOTAL+ICOUNT(I)
214 ...    32 CONTINUE
215 ...       AREA(NCL)=AREAP*ICOUNT(NCL)
216 ...       ITOTAL=ITOTAL+ICOUNT(NCL)
```

```
      TAREA=TAREA+AREA(NCL)
      WRITE(6,304)ICOUNT(NCL),AREA(NCL)
      WRITE(6,308)ITOTAL,TAREA
      STOP
  265 WRITE(8,319)
      WRITE(8,311)
      WRITE(8,305)LNUMA,LCNTR
      IREC=IREC+LINCR
      GO TO 22
  266 WRITE(8,319)
      WRITE(8,313)
      WRITE(8,305)LNUMA,LCNTR
      IREC=IREC+LINCR
      GO TO 22
  267 WRITE(8,319)
      WRITE(8,315)
      WRITE(8,305)LNUMA,LCNTR
      IREC=IREC+LINCR
      GO TO 22
  268 WRITE(8,319)
      WRITE(8,317)
      WRITE(8,305)LNUMA,LCNTR
      IREC=IREC+LINCR
      GO TO 22
  275 WRITE(6,323)
      WRITE(6,325)LCNTR
      STOP
C
C   * WFITE OUTPUT FORMATS
C
  301 FORMAT(' * CU-LIGMALS LINE PRINTER CLASSIFICATION MAPPING SYSTEM',
     1/,' * READY TO PROCEED',/)
  303 FORMAT(' * IMAGE NOW BEING COMPILED',/,' * STAND BY')
  304 FORMAT(16X,'BLANK',' = ',I8,3X,'PIXELS',5X,F15.5,3X,'KM.SQ.')
  305 FORMAT(' **********',/,' * EOF ENCOUNTERED TERMINATE AT LINE',I5,
     1' AFTER ',I5,' LINES WERE PROCESSED')
  306 FORMAT(14X,'SUMMARY OF POINT VALUES',11X,'AREA OF EACH CHARACTER')
  307 FORMAT(20X,A1,' = 'I8,3X,'PIXELS',5X,F15.5,3X,'KM.SQ.')
  308 FORMAT(/,16X,'TOTAL = 'I8,3X,'PIXELS',5X,F15.5,3X,'KM.SQ.')
```

```
 256 ...   309 FORMAT(' ***ERROR***',/,' * AN UNEXPECTED END OF FILE HAS BEEN ENC
 257 ...      1OUNTERED',/,' * ERROR OCCURED WHILE ATTEMPING TO READ HEADER INFO.
 258 ...      2')
 259 ...   311 FORMAT(' * ERROR ON INPUT UNIT #1')
 260 ...   313 FORMAT(' * ERROR ON INPUT UNIT #2')
 261 ...   315 FORMAT(' * ERROR ON INPUT UNIT #3')
 262 ...   317 FORMAT(' * ERROR ON INPUT UNIT #4')
 263 ...   319 FORMAT(' ***ERROR***',/,' * A FATAL READ ERROR HAS OCCURED DURING
 264 ...      1ATTEMPED TO READ HEADER INFO.')
 265 ...   321 FORMAT(' ***WARNING***',/,' * NUMBER OF PIXELS PER LINE EXCEEDS MA
 266 ...      1XIMUM ALLOWABLE PICTURE WIDTH',/,' * TRUNCATION OF IMAGE AFTER FIR
 267 ...      2ST 125 POINTS IN EACH LINE WILL OCCUR')
 268 ...   323 FORMAT(' ***ERROR***',/,' * VIDEO INPUT ERROR',/,
 269 ...      1' * INPUT CHANNELS NOT IDENTICAL')
 270 ...   325 FORMAT(' * ',I5,'LINES HAVE BEEN PROCESSED')
 271 ...   401 FORMAT(/,' * CU-LIGMALS',/,
 272 ...      1' PROJECT CU-LIGMALS CLASSIFICATION MAP',/,
 273 ...      2' MAP OF CHANNEL# ',4I4)
 274 ...   402 FORMAT(' MAP FROM POINTS ',I5,' TO ',I5,' EVERY ',I5,' POINTS',/,
 275 ...      1' LINES ',I5,' TO ',I5,' EVERY ',I5' LINES',/)
 276 ...   403 FORMAT(//,3X,125A1,/)
 277 ...   404 FORMAT(10X,80A1,/)
 278 ...   405 FORMAT('1',131('*'),///,7X,25('+',4X),///)
 279 ...   406 FORMAT(10X,'CLASSIFYING SYMBOLS :',10X,A1,' = ',15A1,/)
 280 ...   407 FORMAT(1X,I5,1X,125A1)
 281 ...   408 FORMAT(41X,A1,' = ',15A1,/)
 282 ...   409 FORMAT(///,7X,25('+',4X),//,1X,131('*')////)
 283 ...   500 FORMAT(10A1)
 284 ...   501 FORMAT(6I2)
 285 ...   502 FORMAT(5X,'MEAN VECTOR',17X,'INVERSE MATRIX',/)
 286 ...   503 FORMAT(5F10.5)
 287 ...   504 FORMAT(6X, F10.5,7X,4F10.5)
 288 ...   505 FORMAT(10F10.5)
 289 ...   506 FORMAT(' ')
 290 ...   507 FORMAT(80A1)
 291 ...   508 FORMAT(5X,'DETERMINANTS : ',10F10.5,//)
 292 ...   509 FORMAT(15A1)
 293 ...   600 FORMAT(F10.6)
 294 ...       END
END OF FILE          294 CARDS
```

ประวัติ

นางสาวกันยา จิรพยุงไชย เกิดเมื่อวันที่ 14 กันยายน 2495 ณ จังหวัดสงขลา สำเร็จการศึกษาสถิติศาสตร์บัณฑิต จากคณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณมหาวิทยาลัย เมื่อปี พ.ศ. 2518 เคยร่วมงานวิจัยเรื่อง "Feasibility Study of System Implementation for the Digital Processing of LANDSAT Data" ซึ่งเป็นโครงการวิจัยที่ได้รับเงินอุดหนุนการวิจัยจากสภาวิจัยแห่งชาติ เคยมีบทความลงพิมพ์ในคอมพิวเตอร์สาร ฉบับเดือนธันวาคม 2520 เรื่อง "ขบวนการแปรภาพด้วยเครื่องคอมพิวเตอร์" และในหนังสือจุลสารดาวเทียมฉบับเดือนตุลาคม 2521 เรื่อง "การใช้เครื่องคอมพิวเตอร์ขนาดเล็กวิเคราะห์ข้อมูลจากดาวเทียม LANDSAT" มีความสนใจเป็นพิเศษทางด้านวิจัยข้อมูลจากดาวเทียมสำรวจทรัพยากร และสถิติ ซึ่งมีประสบการณ์มากกว่า 2 ปี ในการวิเคราะห์ข้อมูลจากดาวเทียมนี้ในเชิงสถิติโดยเครื่องคอมพิวเตอร์ เบอร์โรส์ รุ่น บี 1710, บี 1720, บี 3700 และ ไอบีเอ็ม 370/138 เป็นสมาชิกของคอมพิวเตอร์สมาคมตั้งแต่ปี พ.ศ. 2520 เป็นลูกจ้างชั่วคราวในตำแหน่ง เจ้าหน้าที่ระบบงานคอมพิวเตอร์ 3 ที่โครงการสำรวจทรัพยากรธรรมชาติด้วยดาวเทียม สภาวิจัยแห่งชาติ ตั้งแต่เดือนตุลาคม 2520

หอสมุดกลาง จุฬาลงกรณมหาวิทยาลัย