

IST. EL. INF.
BIBLIOTECA
Posiz. *RECUPERO*

PROGETTO ETIS: PROCEDURA PER LA SEGMENTAZIONE DI UNA IMMAGINE E LA DETERMINAZIONE DI PARAMETRI CARATTERISTICI

Renzo Bozzi

B4-05 (1988)

PROGETTO ETIS: PROCEDURA PER LA SEGMENTAZIONE DI UNA IMMAGINE E LA DETERMINAZIONE DI PARAMETRI CARATTERISTICI

Renzo Bozzi

Istituto di Elaborazione della Informazione - CNR Pisa

1. INTRODUZIONE

In questo rapporto viene descritta e documentata la procedura elaborativa SEGMENT orientata alla rappresentazione interattiva di regioni da immagini ed alla estrazione di parametri caratteristici necessari per il processo di riconoscimento di forme. La procedura, che è stata sviluppata nell'ambito di un contratto di collaborazione tecnico-scientifica stipulata fra l'Istituto di Elaborazione della Informazione del CNR e la società Aeritalia G.V.T., fa parte di un nucleo software applicativo in fase di avanzato sviluppo da inserirsi nella stazione di lavoro ETIS-2 specializzata per l'acquisizione e l'analisi di immagini da impiegarsi nel controllo non invasivo di materiali e componentistica aeronautica.

SEGMENT è stata implementata in Fortran sotto sistema operativo MS/DOS su un elaboratore della categoria Personal IBM/AT configurato con una memoria da 640 Kb, floppy da 1.2 MB, hard disk di 20 Mb e dotato della scheda MATROX PIP-1024 per la memorizzazione e la rappresentazione delle immagini ed eventualmente di un PC/Mouse. SEGMENT può essere utilizzata per qualsiasi tipo di segmentazione basata sulla determinazione interattiva dei punti di contorno di una definita regione dell'immagine purché ovviamente il bordo della regione sia visivamente individuabile; in particolare la procedura può essere impiegata per l'estrazione di regioni relative a difetti rilevati da strutture in materiale composito con tecniche di acquisizione ad ultrasuoni nel caso in cui le malformazioni non siano classificabili con le procedure automatiche già previste nel sistema di sviluppo software GEPETIS.

La procedura è stata realizzata come concatenazione di moduli distinti, ciascuno dei quali svolge funzioni definite e complete, per

cui questi moduli sono stati separatamente inseriti anche nelle librerie di moduli elaborativi di impiego generale.

La procedura di segmentazione ed estrazione di aree da immagini digitali si sviluppa attraverso l'esecuzione distinta ed ordinata di due fasi di elaborazione. Nella prima fase si procede alla vettorizzazione delle coordinate dei punti che costituiscono il contorno del particolare da estrarre; nella seconda fase, partendo dalla conoscenza del contorno di questo particolare, si provvede alla estrazione dall'immagine ed alla determinazione di alcuni parametri caratteristici della regione.

Esaminiamo in particolare i moduli preposti a queste fasi, descrivendo in dettaglio i metodi che essi implementano e le modalita' del loro utilizzo.

2. SEGMENTAZIONE DELLE REGIONI

La segmentazione si esegue mediante la vettorizzazione del contorno utilizzando due programmi, VETT o VETL, che implementano due metodi diversi.

Con VETT si determinano e si acquisiscono le coordinate di una serie di punti di contorno della regione da estrarre che verranno poi interpolati con archi di circonferenza. All'inizio dell'esecuzione e' attivato un colloquio mediante il quale vengono richieste informazioni circa la memoria di quadro sulla quale si trova l'immagine da cui si vuole estrarre un certo particolare, il nome del file di disco nel quale saranno memorizzate le coordinate dei punti del suo contorno, e l'apparecchiatura di acquisizione (joy-stick o mouse) che si intende utilizzare.

Dopo il rilevamento dei primi tre punti P_1, P_2, P_3 viene effettuata una interpolazione con un arco di circonferenza in modo da generare, per una maggiore precisione, anche punti del contorno intermedi a quelli acquisiti. L'arco cosi' definito viene disegnato in sovraimpressione sulla memoria di quadro e i punti relativi vengono archiviati.

Per ogni altro punto P_i ricavato, si procede di nuovo ad una interpolazione con un arco di circonferenza per i punti P_{i-2}, P_{i-1}, P_i in modo da ottenere il nuovo tratto secondo criteri di continuita' con

il tratto precedente e, come nel primo caso, si procede alla attivazione grafica ed alla memorizzazione di tutti i punti trovati appartenenti al tratto P_{i-1}, P_i . In questa fase e' possibile verificare sulla memoria di quadro anche l'esattezza dei dati rilevati; in caso di errore si puo', prima di archiviare i dati relativi all'ultimo arco determinato, ripetere l'operazione modificando l'acquisizione dell'ultimo punto.

Per ogni tratto, la determinazione delle coordinate dei punti del contorno comporta la ricerca dell'equazione della circonferenza passante per i punti P_{i-2}, P_{i-1}, P_i e quindi la risoluzione del sistema lineare di equazioni:

$$\begin{cases} Ax_{i-2} + By_{i-2} + C = -(x_{i-2}^2 + y_{i-2}^2) \\ Ax_{i-1} + By_{i-1} + C = -(x_{i-1}^2 + y_{i-1}^2) \\ Ax_i + By_i + C = -(x_i^2 + y_i^2) \end{cases}$$

Questo determina le incognite A,B,C, coefficienti della equazione della circonferenza $x^2+y^2+Ax+By+C=0$, in modo appunto che essa passi per i tre punti suddetti. La risoluzione del presente sistema viene poi affidata ad una routine che implementa il metodo di Gauss.

Tale modo di procedere ci permette anche di ovviare a certe situazioni critiche come nel caso di un sistema con matrice a determinante nullo; in questo caso (oppure quando il raggio di curvatura della circonferenza supera un valore predefinito) si ricorre ad una semplice interpolazione lineare per i punti P_{i-1}, P_i .

Alla fine della generazione dell'ultimo tratto, il programma memorizza, come coordinate dell'ultimo punto, quelle del primo punto acquisito in maniera da chiudere automaticamente il contorno del particolare e trasferisce tutti i dati sul file di disco individuato all'inizio.

In figura 1 e' rappresentato lo schema logico seguito dal programma.

Con il programma VETL invece si determinano e si acquisiscono le coordinate dei punti del contorno ricorrendo al rilevamento di un certo numero di punti che vengono interpolati linearmente. Mediante l'attivazione di un colloquio iniziale, vengono richiesti i parametri di

acquisizione: la memoria di quadro sulla quale si trova l'immagine da cui si vuole estrarre un certo particolare, il nome del file di disco nel quale saranno memorizzate le coordinate dei punti del contorno, e l'apparecchiatura di acquisizione (joy-stick o mouse) che si intende utilizzare.

La fase successiva e' costituita dal rilevamento interattivo delle coordinate di una serie di punti di contorno della regione di interesse. Dopo il rilevamento dei primi due punti P_1, P_2 viene effettuata una interpolazione lineare in modo da generare, per una maggiore precisione, anche punti del contorno intermedi a quelli presi e quindi si procede ad una attivazione grafica di controllo sulla memoria di quadro e ad una memorizzazione di tutti i punti trovati. Allo stesso modo, per ogni altro punto P_i ricavato, si interpola linearmente, secondo l'incremento maggiore delle coordinate, il tratto P_{i-1}, P_i ricorrendo alle formule:

$Y = (Y_2 - Y_1) * (X_i - X_1) / (X_2 - X_1) + Y_1$ per l'interpolazione su X

$X = (X_2 - X_1) * (Y_i - Y_1) / (Y_2 - Y_1) + X_1$ per l'interpolazione su Y

dove X_1, Y_1 e X_2, Y_2 sono le coordinate rispettivamente del primo e dell'ultimo punto del tratto e X_i e Y_i sono le generiche x e y dell'i-mo punto interno al segmento a partire dal primo.

In questa fase e' possibile verificare sulla memoria di quadro l'esattezza dei dati rilevati ed eventualmente correggere l'ultima acquisizione ripetendo il rilevamento.

Alla fine della generazione dell'ultimo tratto, il programma memorizza, come coordinate dell'ultimo punto, quelle del primo punto acquisito in maniera da chiudere automaticamente il contorno del particolare e trasferisce tutti i dati sul file di disco individuato all'inizio.

Sulle coordinate determinate dei punti del contorno che sono state trasferite su file di disco, viene eseguito uno smoothing con finestra definibile. E' possibile anche eseguire uno smoothing con una diversa dimensione della finestra mediante il modulo SMOOT che, a partire dai dati archiviati, costruisce un file di disco di dati filtrati e visualizza il risultato sulla memoria di quadro.

Il diagramma dinamico concettuale secondo cui opera il programma VETL e' raffigurato in figura 2, quello del programma SMOOT in figura 3.

3. ESTRAZIONE DI REGIONI E DEFINIZIONE DEI PARAMETRI CARATTERISTICI

Questa funzione viene effettuata dal modulo elaborativo ESTRAZ che trova fondamentale supporto in un algoritmo di filling implementato come subroutine al suo interno. Tale algoritmo e' implementato anche come modulo elaborativo a se' stante di nome FILRUN il quale provvede al riempimento del poligono definito da un contorno rilevato con i moduli VETT o VETL.

ESTRAZ e' un programma che, operando congiuntamente su una immagine contornata e sul contorno riempito, esegue l'estrazione della regione di interesse ricopiandola in una apposita memoria di uscita.

Descrizione del programma FILRUN

Il programma chiede se il contorno da riempire e' residente su una memoria di quadro o in un file di disco; in quest'ultimo caso viene eseguito il trasferimento su una memoria di quadro prescelta.

Individuato un pixel interno al poligono, denominato "seme", che servira' come base di partenza per lo sviluppo del filling, facendo uso del joy-stick o del mouse, questo sara' "acceso" come lo saranno tutti quelli alla sua destra, finche' non si incontra il bordo del contorno. Il pixel che precede il bordo e' memorizzato in uno stack in quanto estremo di un "run", dove per "run" si intende una serie di pixel adiacenti, orizzontali e interni al poligono.

Il pixel di partenza, salvato prima dell'inizio del riempimento a destra, serve ancora come elemento di inizio per il riempimento a sinistra che termina una volta raggiunto il bordo. Il pixel che segue il pixel di bordo sinistro raggiunto e' anch'esso salvato. Una volta completato il riempimento del "run" cui fa parte il seme, si passa alla linea superiore per scoprire se in questa ci sono "run" da riempire; una volta individuati tutti gli eventuali "run" superiori, l'algoritmo esplora e riempie quelli relativi alle righe sottostanti quello di partenza.

Riassumendo, l'algoritmo segue il seguente processo operativo:

- 1) LEGGI IL SEME
- 2) PROPAGA IL COLORE A DESTRA E POI A SINISTRA
- 3) VAI ALLA LINEA SUPERIORE, METTI NELLO STACK TUTTI I PIXEL PIU' A DESTRA DEI RUN CHE HANNO CONTATTO CON IL RUN INFERIORE
- 4) VAI ALLA LINEA INFERIORE, METTI NELLO STACK TUTTI I PIXEL PIU' A DESTRA DEI RUN CHE HANNO CONTATTO CON IL RUN SUPERIORE
- 5) SE LO STACK E' VUOTO VAI A 8), ALTRIMENTI ESTRAI DALLO STACK L'ELEMENTO DI TESTA
- 6) PROPAGA IL COLORE DA QUESTO PUNTO VERSO SINISTRA FINO AD INCONTRARE IL BORDO
- 7) TORNA AL PASSO 3)
- 8) FINE

Essendo i "run" in numero molto inferiore alla totalita' dei pixel interni del poligono, lo stack avra' un'occupazione di memoria contenuta. Per quanto riguarda il numero di visite di ciascun pixel, questo dipende dalla geometria della figura, cioe' da quanti sono i "run" a contatto con il run di cui il pixel fa parte.

Il metodo di riempimento che abbiamo descritto non funziona per poligoni intrecciati, in quei casi cioe' nei quali seguendo il contorno, si passa piu' di una volta per il solito punto.

La figura 4 illustra ulteriormente l'algoritmo, mentre le figure 5a e 5b mostrano come un poligono venga successivamente riempito; in queste figure i numeri indicano la posizione dello stack in cui gli indirizzi degli estremi destri dei "run" sono immagazzinati.

Il programma FILRUN restituisce come risultato il riempimento con un valore scelto, del contorno in esame sulla memoria di quadro e quindi (solo per il caso di lettura da file di disco) le misure in pixel del suo perimetro e della superficie racchiusa. Mentre l'area viene calcolata come numero di pixel interni "accesi" al valore prescelto, il valore del perimetro e' calcolato come somma dei segmenti che costituiscono la spezzata chiusa di contorno.

Descrizione del programma ESTRAZ

All'inizio dell'esecuzione di ESTRAZ vengono richieste una serie di informazioni:

- 1) la prima memoria di quadro, ossia quella su cui si trova memorizzata l'immagine d'ingresso in esame;
- 2) la memoria di quadro di lavoro, che e' quella dove verra' trasferito dall'archivio il contorno del particolare da estrarre, gia' rilevato precedentemente con un programma di vettorizzazione;
- 3) il nome del file di ingresso dei punti di contorno, cioe' il nome dell'archivio;
- 4) il numero di poligonal di contorno, dal momento che e' possibile avere buchi nel particolare da estrarre dovuti a poligonal annidate;
- 5) il ricorso all'uso del joy-stick o del mouse, per selezionare un punto interno al contorno, che servira' per dare inizio all'algoritmo di riempimento;
- 6) la memoria di quadro di uscita sulla quale verra' eseguita la costruzione del particolare da estrarre.

Il programma esegue sulla memoria di lavoro il filling dell'area delimitata dal contorno e ricopia nella memoria di uscita i punti omologhi contenuti nella memoria di ingresso.

Durante l'esecuzione, il programma provvede: al calcolo del numero di pixel che sono stati trasferiti durante questa estrazione, al calcolo della distribuzione del valore di tali pixel e, su richiesta, alla visualizzazione dell'istogramma della distribuzione dei valori dei pixel che verra' disegnato sul monitor grafico-pittorico. Sul monitor alfanumerico verranno date notizie relative al particolare estratto, e piu' precisamente:

- valore massimo
- valore minimo
- misura dell'area (in pixel)
- misura del perimetro (in pixel).

L'area e il perimetro sono calcolati, la prima come somma dei pixel trasferiti, il secondo come la somma della lunghezza dei segmenti che costituiscono le poligonal chiuse. Come ultima cosa il programma lascia

facolta' all'utente di trasferire queste informazioni su una o piu' memorie selezionabili su video grafico pittorico.

Alla fine viene chiesto se si vuol procedere alla estrazione di un altro particolare, ed in caso affermativo, con quali modalita': se inizializzando di nuovo tutto quanto ha concorso al precedente risultato, od in maniera da ottenere l'immagine di una estrazione di piu' particolari.

In figura 6 sono brevemente descritte le procedure secondo cui si muove il programma ESTRAZ.

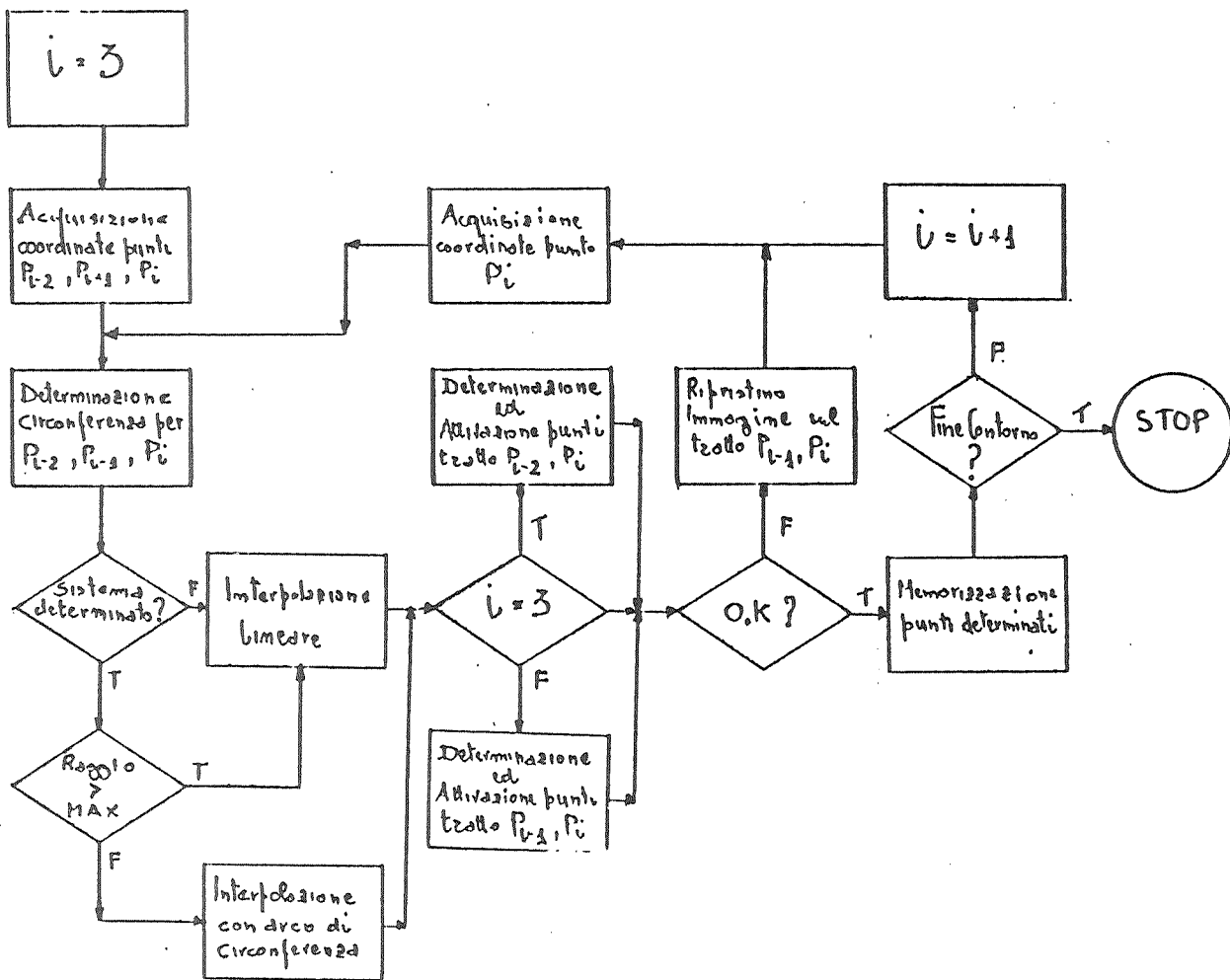


Fig.1 - Schema logico seguito dal programma VETT di interpolazione con archi.

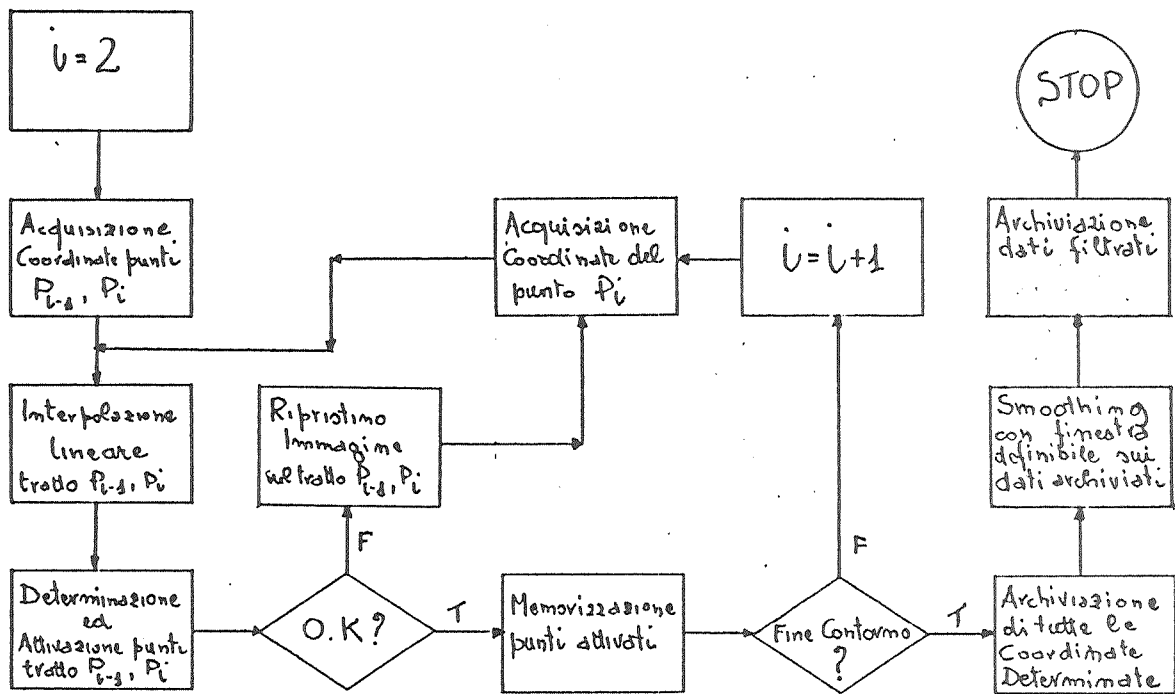


Fig.2 - Schema logico seguito dal programma VETL di interpolazione lineare.

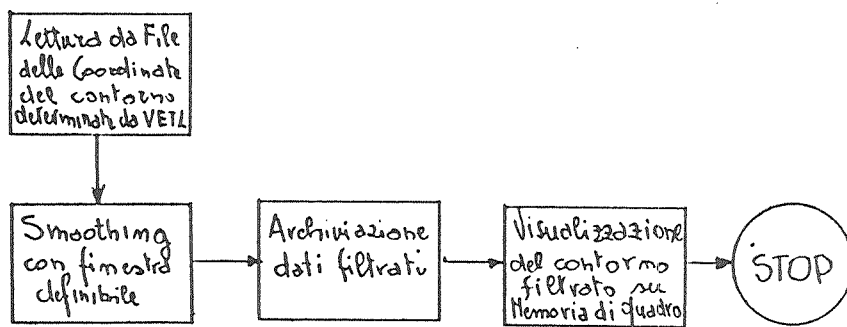


Fig.3 - Schema logico seguito dal programma SMOOT di smoothing.

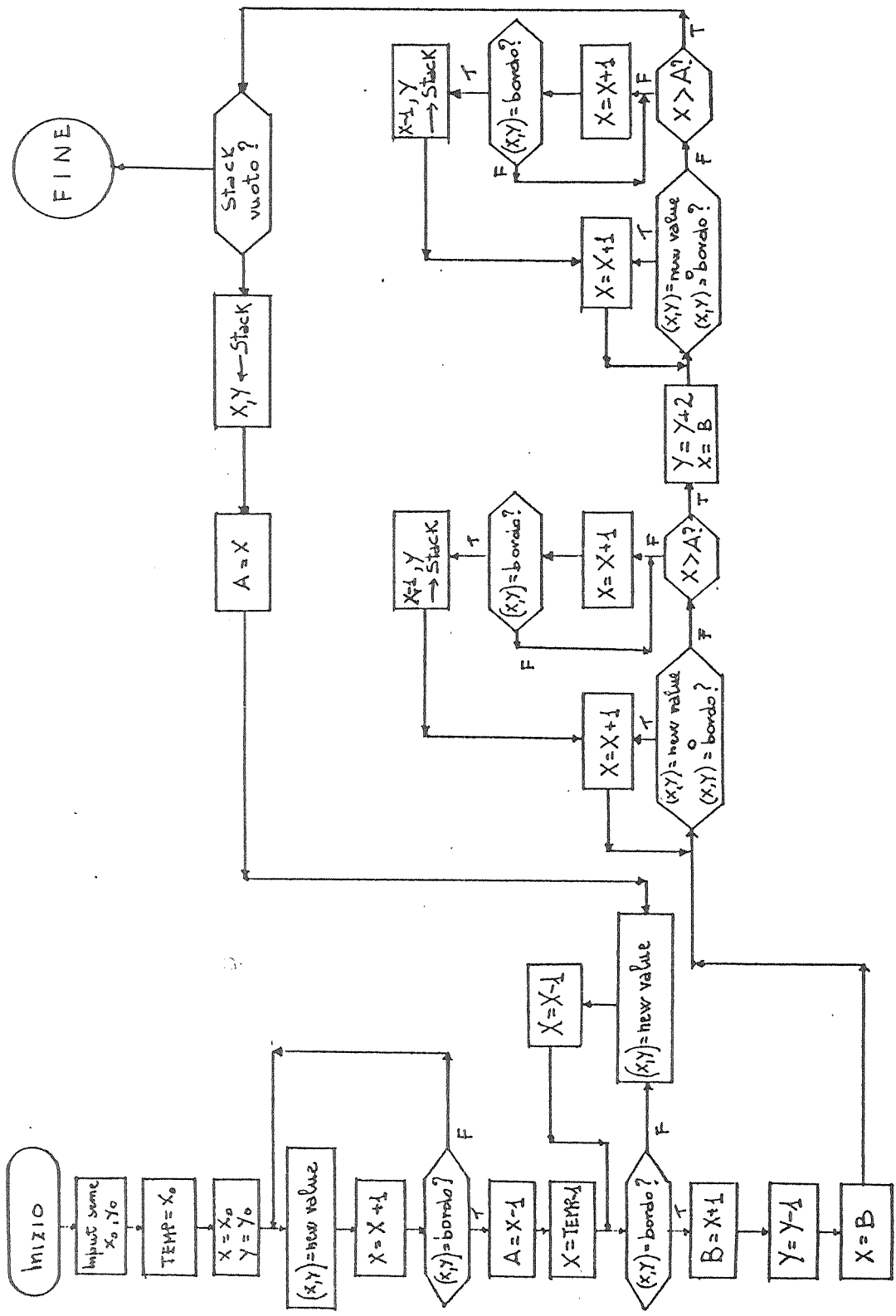


Fig.4 - Schema logico seguito dal programma FILRUN che esegue il filling di contorno.
 N.B. Tenere presente che l'origine degli assi cartesiani è in alto a sinistra.

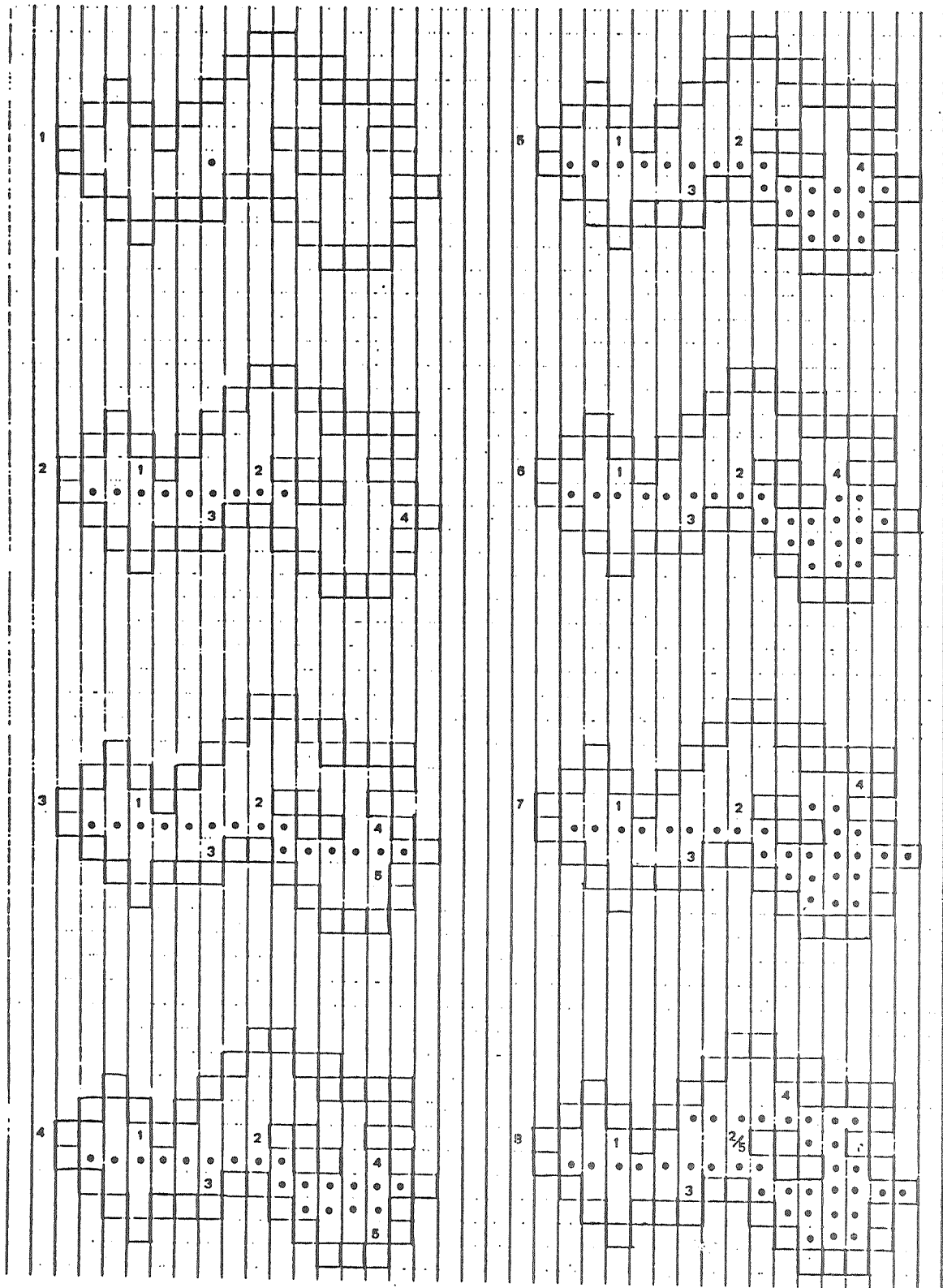


Fig.5a - Esempio di riempimento di un contorno col programma FILRUN (prima parte).

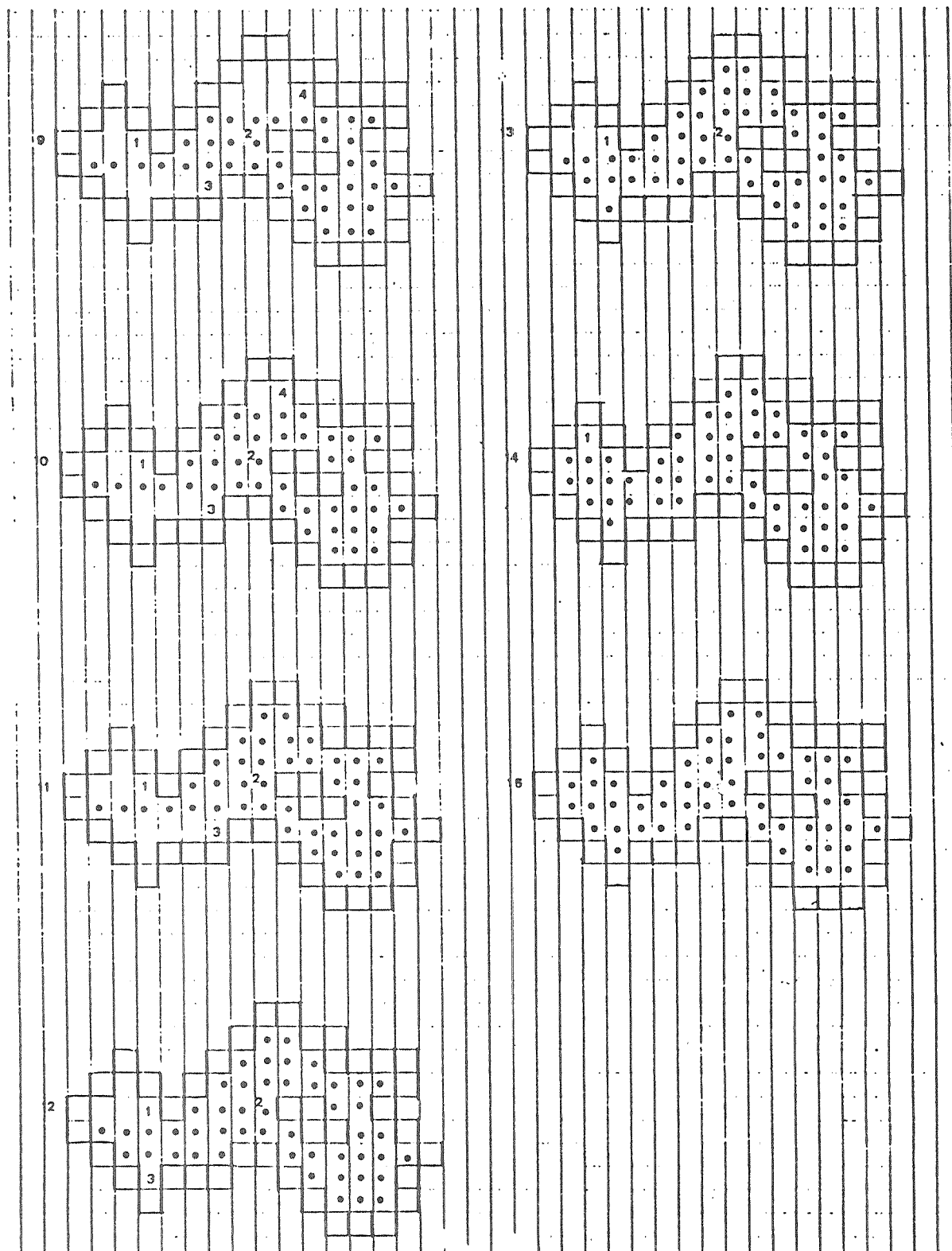


Fig.5b - Esempio di riempimento di un contorno col programma FILRUN
(seconda parte).

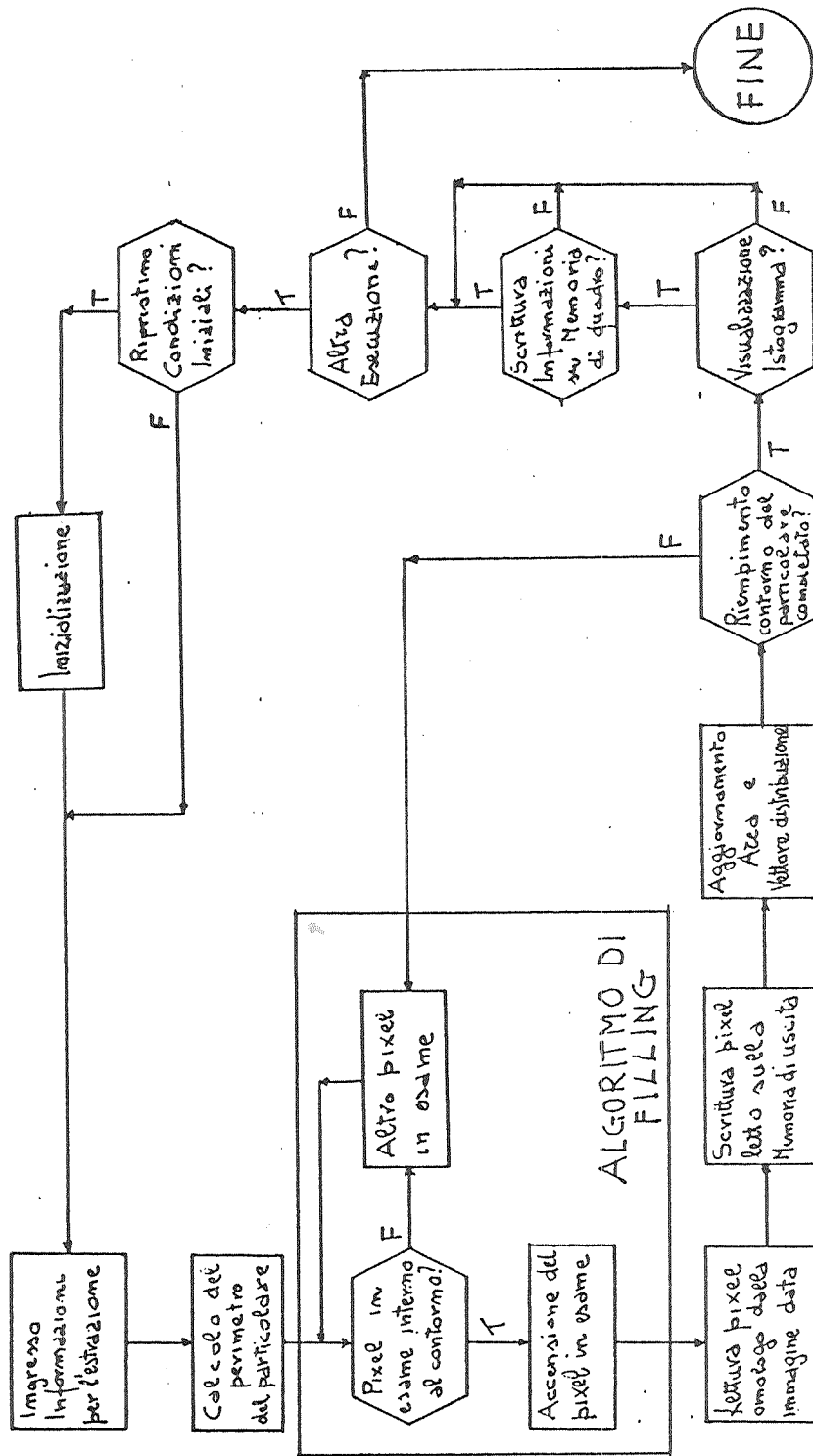


Fig.6 - Schema logico seguito dal programma ESTRAZ per l'estrazione di un particolare da un'immagine.

```

PROGRAM SEGMENT
C
C   PROCEDURA DI VETTORIZZAZIONE DI UN CONTORNO CON INTERPOLAZIONE
C   LINEARE, SMOOTHING CON FINESTRA DEFINIBILE, E SUA ESTRAZIONE
C   DALLA IMMAGINE ORIGINALE SU MEMORIA DI QUADRO.
C
  IMPLICIT INTEGER*2 (I-N)
  INTEGER*2 VX(3000),VY(3000),IX(800),IY(800),XYMIN,XYMAX
  CHARACTER*64 FNAM
  CHARACTER*1 IMOV,SINO,MJ
C
C
  NNC=0
  CALL CLS
  LMAXV=2999
  NUM=0
  WRITE(*,2)
2  FORMAT(///// ,25X,'*** VETTORIZZAZIONE DI UN CONTORNO ***'/30X,
  *1X,'con Interpolazione Lineare'/)
  WRITE(*,3)
3  FORMAT(/5X,'*** INGRESSO DATI ***'/)
7  WRITE(*,5)
5  FORMAT(5X,'Memoria di Quadro (0-3) ? : '\)
  READ(*,*,ERR=7) IMM
  IF(IMM.LT.0.OR.IMM.GT.3) GOTO 7
  CALL DISPL(IMM)
  WRITE(*,6)
6  FORMAT(5X,'Ok ? (y/n) : '\)
  READ(*,10) SINO
10  FORMAT(A1)
  IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 7
41  WRITE(*,40)
40  FORMAT(/5X,'Selezione punti da Joy-stick (J) o Mouse (M) ? : '\)
  READ(*,10) MJ
62  NNC=NNC+1
  CALL CLS
  NUM=0
  K=0
  ICX=255
  ICY=255
  IF(MJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 1
C
C   USO DEL JOESTICK
C
  WRITE(*,56)
56  FORMAT(/5X,'Istruzioni per l'uso del Joy-stick :'/5X,'ESC',
  *4X,'per acquisire Coordinate'/5X,'FRECCE per spostamenti'/5X,
  *'BARRA per spostamenti veloci'/5X,'CR per terminare'/)
  WRITE(*,4)
4  FORMAT(/22X,'*** INIZIO RILEVAMENTO PUNTI ***'/)
C
  CALL GETXY(ICX,ICY)
  A=ICX
  B=ICY
  IC1=ICX
  IC2=ICY
  CALL CURS(IC1,IC2)
125  CALL GETXY(ICX,ICY)
  C=ICX
  D=ICY
  CALL INRET(A,B,C,D,IX,IY,LUN)
  IFINE=NUM+LUN
  IF(IFINE.GT.LMAXV) THEN

```



```

WRITE(*,532)
532  FORMAT(' ATTENZIONE !: Area memorizzazione contorno SATURA'/
*15X,'Ci si ARRESTA ! ')
STOP
ENDIF
DO 24 KL=1,LUN
NUM=NUM+1
VX(NUM)=IX(KL)
VY(NUM)=IY(KL)
24  CONTINUE
A=C
B=D
CALL LETCAR(IMOV)
IU=ICHAR(IMOV)
IF(IU.NE.13) GOTO 125
GOTO 133

C
C  USO DEL MOUSE
C
1  XYMIN=0
XYMAX=511
M1=0
CALL MOUSES(M1,M2,M3,M4)
IF(M1.EQ.0) THEN
WRITE(*,321)
321  FORMAT(5X,'ERRORE!! : Mouse non installato')
STOP
ENDIF
WRITE(*,46)
46  FORMAT(/5X,'Istruzioni per l'uso del Mouse :'/78X,'TASTO 4 per',
*1X,'rilevamento Coordinate'/7X,'TASTO 1 per terminare'/)
WRITE(*,4)

C
MM2=-1
CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
A=ICX
B=ICY
IC1=ICX
IC2=ICY
CALL CURS(IC1,IC2)
126  MM2=-1
CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
C=ICX
D=ICY
CALL INRET(A,B,C,D,IX,IY,LUN)
IFINE=NUM+LUN
IF(IFINE.GT.LMAXV) THEN
WRITE(*,532)
ENDIF
DO 25 KL=1,LUN
NUM=NUM+1
VX(NUM)=IX(KL)
VY(NUM)=IY(KL)
25  CONTINUE
A=C
B=D
MM2=-1
CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
IF(MM2.NE.1) GOTO 126

C
133  CALL CURS(IC1,IC2)
NUM=NUM+1
VX(NUM)=VX(1)
VY(NUM)=VY(1)

C
C  SMOOTHING

```

```

L
395 WRITE(*,666)
666 FORMAT(/5X,'Dimensione della finestra di Smoothing ? : '\)
    READ(*,*,ERR=395) N
    IF(NNC.GT.1) GOTO 42
    WRITE(*,33)
33  FORMAT(/5X,'Nome File punti filtrati in Uscita ? : '\)
    READ(*,32) FNAM
32  FORMAT(A)
    OPEN(3,FILE=FNAM,STATUS='NEW')
42  CALL SMOOT(VX,VY,NUM,N)
    WRITE(3,77) NUM
    WRITE(3,77)(VX(L),VY(L),L=1,NUM)
77  FORMAT(20I4)
    CALL CLS
63  WRITE(*,60)
60  FORMAT(/5X,'Ancora ? (y/n) : '\)
    READ(*,10) SINO
    IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 62
    IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 63
    CLOSE (3)
    CALL ESTRAI(VX,VY,FNAM,IMM,NNC)
    GOTO 124
111 write(*,112)
112 format(/' ERRORE. Coordinate -X- fuori range.'/)
    stop
222 write(*,223)
223 format(/' ERRORE. Coordinate -Y- fuori range.'/)
    stop
333 write(*,334)
334 format(/' ERRORE. Parametro TASTO fuori range.'/)
    stop
124 CONTINUE
    END
C *****
SUBROUTINE INRET(A,B,C,D,IX,IY,LUN)
  IMPLICIT INTEGER*2 (I-N)
  DIMENSION IX(1),IY(1)
C
  IA=A
  IB=B
  IC=C
  ID=D
  LUN=0
C
C  ATTIVAZIONE E VETTORIZZAZIONE DEI PUNTI DI INTERPOLAZIONE
C
  F=D-B
  F1=ABS(F)
  G=C-A
  G1=ABS(G)
  IF(F1.GT.G1) GOTO 1
  IPS=1
  IF(IA.GT.IC) IPS=-1
  DO 2 I=IA,IC,IPS
  LUN=LUN+1
  IX(LUN)=I
  H=FLOAT(I)-A
  IY(LUN)=(F*H)/G+B+0.5
2  CONTINUE
  GOTO 6
1  IPS=1
  IF(IB.GT.ID) IPS=-1
  DO 4 I=IB,ID,IPS
  LUN=LUN+1
  IY(LUN)=I

```

```

H=FLOAT(I)-B
IX(LUN)=(G*H)/F+A+0.5
4 CONTINUE
6 RETURN
END)
C*****
SUBROUTINE SMOOT(VX,VY,NUM,N)
C
C SOTTOPROGRAMMA DI SMOOTHING
C
C IMPLICIT INTEGER*2 (I-N)
C INTEGER*2 VX(1),VY(1)
C
C FN=FLOAT(N)
M=N/2
L=M+1
N1=N-2
NFINE=NUM+M-1
K=0
DO 1 I=1,N1
K=K+1
NUM1=NUM+K
VX(NUM1)=VX(I+1)
VY(NUM1)=VY(I+1)
1 CONTINUE
K=0
DO 2 I=L,NFINE
K=K+1
I1=I-M
I2=I+M
X=0.
Y=0.
DO 3 J=I1,I2
X=X+FLOAT(VX(J))
Y=Y+FLOAT(VY(J))
3 CONTINUE
VX(K)=X/FN+0.5
VY(K)=Y/FN+0.5
2 CONTINUE
VX(NUM)=VX(1)
VY(NUM)=VY(1)
RETURN
END)
C *****
SUBROUTINE ESTRAI(IX,IY,FNAM,IMM,NNC)
C
C ESTRAZIONE DA UN IMMAGINE DI UN PARTICOLARE CONTORNATO
C
C IMPLICIT INTEGER*2 (I-N)
C INTEGER*2 IX(1),IY(1),VAL,XYMIN,XYMAX
C INTEGER*4 AREA,MISPER,ISTOG(256),MAXX
C CHARACTER*64 FNAM
C CHARACTER*26 IEI
C CHARACTER*1 SINO,MJ
C
C IEI=' *** I.E.I. *** '
100 MISPER=0
CALL CLS
CALL DISPL(IMM)
WRITE(*,1)
1 FORMAT(///,25X,'*** ESTRAZIONE DEL PARTICOLARE ***')
WRITE(*,2)
2 FORMAT(//5X,'*** INGRESSO DATI ***'/)
8 WRITE(*,7)
7 FORMAT(5X,'Memoria di Quadro di lavoro (0-3) ?: \')
READ(*,*.ERR=9) IM1

```

```

IF(IM1.LT.0.OR.IM1.GT.3.OR.IM1.EQ.IMM) GOTO 8
CALL DISPL(IM1)
WRITE(*,5)
5  FORMAT(5X,'OK ? (y/n): '\)
   READ(*,6) SINO
6  FORMAT(A1)
   IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 8
   VAL=255
   OPEN(3,FILE=FNAM,STATUS='OLD')
   DO 62 JL=1,NNC
   READ(3,11) N
   READ(3,11) (IX(I),IY(I),I=1,N)
11  FORMAT(20I4)
   DO 13 I=2,N
   I1=I-1
   LUNG=SQRT((IX(I)-IX(I1))**2+(IY(I)-IY(I1))**2)
   CALL LINE(IX(I1),IY(I1),IX(I),IY(I),VAL)
   MISPER=MISPER+LUNG
13  CONTINUE
62  CONTINUE
   CLOSE (3)
   IX0=255
   IY0=255
   IF(NJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 399
   WRITE(*,398)
398  FORMAT(5X,'Seleziona col Joy-stick un punto interno al contorno')
     CALL GETXY(IX0,IY0)
     GOTO 397
399  XYMIN=0
     XYMAX=511
     CALL RESM
     WRITE(*,31)
31  FORMAT(5X,'Seleziona col Mouse un punto interno al contorno')
     M2=-1
     CALL MS_COO(IX0,IY0,XYMIN,XYMAX,XYMIN,XYMAX,M2,*111,*222,*333)
397  WRITE(*,14)
14  FORMAT(/5X,'*** USCITA DATI ***'/)
15  WRITE(*,16)
16  FORMAT(5X,'Memoria di Quadro (0-3) ? : '\)
     READ(*,*,ERR=15) IM2
     IF(IM2.LT.0.OR.IM2.GT.3) GOTO 15
     IF(IM2.EQ.IM1.OR.IM2.EQ.IM1) GOTO 15
     CALL DISPL(IM2)
     WRITE(*,5)
     READ(*,6) SINO
     IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 15
     CALL FILR4(IM1,IM1,IM2,IX0,IY0,AREA,ISTOG)
52  WRITE(*,50)
50  FORMAT(/5X,'Disegno Istogramma del particolare (y/n) ? : '\)
     READ(*,6) SINO
     IF(SINO.EQ.'N'.OR.SINO.EQ.'n') GOTO 51
     IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 52
53  WRITE(*,16)
     READ(*,*,ERR=53) IM3
     IF(IM3.LT.0.OR.IM3.GT.3) GOTO 53
     CALL DISPL(IM3)
     WRITE(*,5)
     READ(*,6) SINO
     IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 53
     WRITE(*,54)
54  FORMAT(/5X,'Azzeramento (y/n) ? : '\)
     READ(*,6) SINO
     IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') CALL SET(0)
     DO 55 I=1,256
     IF(ISTOG(I).NE.0) GOTO 56
55  CONTINUE

```

```

56   MIN=I-1
      DO 57 I=256,1,-1
      IF(ISTOG(I).NE.0) GOTO 58
57   CONTINUE
58   MAX=I-1
      MAXX=0
      DO 63 I=1,256
      IF(ISTOG(I).GT.MAXX) MAXX=ISTOG(I)
63   CONTINUE
      WRITE(*,59) MIN,MAX
59   FORMAT(/5X,'Valore minimo del particolare = ',I4/5X,
*'Valore massimo del particolare = ',I4/)
      CALL DHISTO(MAXX,50,450,180,255,300,ISTOG)
      IT1=22
      IT2=56
      IT3=26
      IT4=2
      IT5=0
      IT6=255
      CALL TEXT(IT1,IT2,IEI,IT3,IT4,IT5,IT6)
51   WRITE(*,20) MISPER,AREA
20   FORMAT(/5X,'Misura Perimetro del Contorno (in Pixels) = ',I6/
*5X,'Misura Area del Riempimento (in Pixels) = ',I6//)
23   WRITE(*,21)
21   FORMAT(5X,'Scrittura Misure su Memoria di Quadro ? (y/n) : '\)
      READ(*,6) SINO
      IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 22
      IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 23
      GOTO 124
22   CALL WRTA(MJ)
      GOTO 124
111  write(*,112)
112  format(/' ERRORE. Coordinate -X- fuori range.'/)
      stop
222  write(*,223)
223  format(/' ERRORE. Coordinate -Y- fuori range.'/)
      stop
333  write(*,334)
334  format(/' ERRORE. Parametro TASTO fuori range.'/)
      stop
124  RETURN
      END

```

C *****

```

SUBROUTINE FILR4(IMM,IM1,IM2,IX0,IY0,AREA,ISTOG)
IMPLICIT INTEGER*2 (I-N)
DIMENSION IST(1000,2)
INTEGER*2 VAL,BUF(2),CONT
INTEGER*4 AREA,ISTOG(256)

```

C

```

      DO 17 JJ=1,256
17   ISTOG(JJ)=0
      AREA=0
      CALL SETSCR(IM1)
      K=0
      CONT=1
      VAL=255
      ITMP=IX0
      IX=IX0
      IY=IY0
3   CALL WPIX(IX,IY,VAL)
      CALL SETSCR(IMM)
      CALL ROWRI(IX,IY,CONT,BUF)
      CALL SETSCR(IM2)
      AREA=AREA+1
      JJ=BUF(1)+1
      ISTOG(JJ)=ISTOG(JJ)+1

```

```

CALL WPIX(IX,IY,BUF(1))
CALL SETSCR(IM1)
IX=IX+1
CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 3
IA=IX-1
IX=ITMP-1
4 CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).EQ.VAL) GOTO 5
16 CALL WPIX(IX,IY,VAL)
CALL SETSCR(IMM)
CALL ROWRI(IX,IY,CONT,BUF)
CALL SETSCR(IM2)
AREA=AREA+1
JJ=BUF(1)+1
ISTOG(JJ)=ISTOG(JJ)+1
CALL WPIX(IX,IY,BUF(1))
CALL SETSCR(IM1)
IX=IX-1
GOTO 4
5 IB=IX+1
IY=IY-1
IX=IB
6 CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 7
10 IX=IX+1
GOTO 6
7 IF(IX.GT.IA) GOTO 11
8 IX=IX+1
CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 8
K=K+1
IST(K,1)=IX-1
IST(K,2)=IY
GOTO 10
11 IY=IY+2
IX=IB
9 CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 12
15 IX=IX+1
GOTO 9
12 IF(IX.GT.IA) GOTO 13
14 IX=IX+1
CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 14
K=K+1
IST(K,1)=IX-1
IST(K,2)=IY
GOTO 15
13 IF(K.EQ.0) RETURN
IX=IST(K,1)
IY=IST(K,2)
K=K-1
IA=IX
GOTO 16
END)
C *****
SUBROUTINE WRTA(MJ)
  implicit integer*2(i-n)
  character*56 st1
  character*1 ies,mj
  integer*2 rpix,xymin,xymax
100 WRITE(*,4)
4  FORMAT(5X,'Memoria di Quadro (0-3) ?: '\)
  READ(*,*,ERR=100) IM
  IF(IM.LT.0.OR.IM.GT.3) GOTO 100

```

```

CALL DISPL(IM)
WRITE(*,5)
5  FORMAT(5X,'Ok ? (y/n): '\)
   READ(*,6) SINO
6  FORMAT(A1)
   IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 100
   il=255
   idim=1
   idir=0
   nch=56
       write(*,15)
15  format(5x,'- Inserire il testo -'/56('*'))
   read(*,16) st1
16  format(a)
   ix=255
   iy=255
   if(mj.eq.'M'.or.mj.eq.'m') goto 3
   write(*,1)
1  format(/5x,'Determina col Joy-stick l''inizio di scrittura'/)
   call getxy(ix,iy)
   goto 2
3  write(*,444)
444 format(/5x,'Determina col Mouse l''inizio di scrittura',/)
   xymin=0
   xymax=511
   call resm
   m2=-1
   call ms_coo(ix,iy,xymin,xymax,xymin,xymax,m2,*111,*222,*333)
2  call text(ix,iy,st1,nch,idim,idir,il)
   write(*,40)
40  format(/5x,'Ancora ? (y/n): ',)$)
   read(*,14) ies
14  format(a1)
   if (ies.eq.'y' .OR. ies.eq.'Y') goto 100
   goto 415
111 write(*,112)
112 format(/' ERRORE. Coordinate -X- fuori range.'/)
   stop
222 write(*,223)
223 format(/' ERRORE. Coordinate -Y- fuori range.'/)
   stop
333 write(*,334)
334 format(/' ERRORE. Parametro TASTO fuori range.'/)
   stop
415 continue
   end
C *****
C SUBROUTINE RESM
C IMPLICIT INTEGER*2 (I-N)
C
C INIZIALIZZAZIONE CON RESET DEL PC-MOUSE
C
   m1=0
   call mouses(m1,m2,m3,m4)
   if(m1.eq.0) then
   write(*,1)
1  format(/5x,'ERRORE!! : Mouse non installato')
   stop
   endif
   m1=7
   m3=0
   m4=511
   call mouses(m1,m2,m3,m4)
   m1=8
   m3=0
   m4=511

```

```
call mouses(m1,m2,m3,m4)
return
end
```



```

PROGRAM VETT
C
C VETTORIZZAZIONE DI UN CONTORNO
C
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 VX(2500),VY(2500),IA(800),IB(800),XYMIN,XYMAX
INTEGER*4 M1,M2,M3,M4
DIMENSION X(3),Y(3),A(3,4)
CHARACTER*64 FNAM
CHARACTER*1 IMOV,SINO,MJ,T
C
C
IAF=0
62 CALL CLS
LMAXV=2499
IT=3
IQ=4
NUM=0
WRITE(*,2)
2 FORMAT(////,25X,'*** VETTORIZZAZIONE DI UN CONTORNO ***'/28X,
*1X,'con Interpolazione Circolare'/)
WRITE(*,3)
3 FORMAT(/5X,'*** INGRESSO DATI ***'/)
7 WRITE(*,5)
5 FORMAT(5X,'Memoria di Quadro (0-3) ?: '\)
READ(*,*,ERR=7) IMM
IF(IMM.LT.0.OR.IMM.GT.3) GOTO 7
CALL DISPL(IMM)
WRITE(*,6)
6 FORMAT(5X,'Ok ? (y/n): '\)
READ(*,10) SINO
10 FORMAT(A1)
IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 7
IF(IAF.EQ.1) GOTO 41
WRITE(*,31)
31 FORMAT(/5X,'Nome File di Uscita ?: '\)
READ(*,32) FNAM
32 FORMAT(A)
OPEN(2,FILE=FNAM,STATUS='NEW')
41 WRITE(*,40)
40 FORMAT(/5X,'Selezione punti da Joy-stick (J) o Mouse (M) ?: '\)
READ(*,10) MJ
IF(MJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 45
WRITE(*,56)
56 FORMAT(/5X,'Istruzioni per l'uso del Joy-stick :'/5X,'ESC',
*4X,'per acquisire Coordinate'/5X,'FRECCHE per spostamenti'/5X,
*'BARRA',2X,'per spostamenti veloci'/5X,'CR per terminare'/)
GOTO 14
45 WRITE(*,46)
46 FORMAT(/5X,'Istruzioni per l'uso del Mouse :'/8X,'TASTO 4 per',
*1X,'rilevamento Coordinate'/7X,' TASTO 1 per terminare'/)
14 WRITE(*,4)
4 FORMAT(/22X,'*** INIZIO RILEVAMENTO PUNTI ***'/)
K=0
ICX=255
ICY=255
IF(ISEL.EQ.1) GOTO 1
XYMIN=0
XYMAX=511
M1=0
CALL MOUSES(M1,M2,M3,M4)
IF(M1.EQ.0) THEN
WRITE(*,321)

```

```

321  FORMAT(5X,'ERRORE!! : Mouse non installato')
      STOP
      ELSE
      CALL RESM
      ENDIF
1    IF (MJ.EQ.'J'.OR.MJ.EQ.'j') THEN
      CALL GETXY(ICX,ICY)
      GOTO 44
      ENDIF
      MM2=-1
      CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
44   K=K+1
      X(K)=ICX
      Y(K)=ICY
      IF (MJ.EQ.'M'.OR.MJ.EQ.'m') CALL RESM
      IM=MOD(K,3)
      IF (IM.NE.0) GOTO 1
C   INTERPOLAZIONE
      A(1,1)=X(1)
      A(1,2)=Y(1)
      A(1,3)=1.
      A(1,4)=-(X(1)*X(1)+Y(1)*Y(1))
      A(2,1)=X(2)
      A(2,2)=Y(2)
      A(2,3)=1.
      A(2,4)=-(X(2)*X(2)+Y(2)*Y(2))
      A(3,1)=X(3)
      A(3,2)=Y(3)
      A(3,3)=1.
      A(3,4)=-(X(3)*X(3)+Y(3)*Y(3))
      CALL GAUS(A,IT,IQ,B)
      AA=A(1,1)
      BB=A(2,1)
      CC=A(3,1)
      T=MJ
      IF (NUM.EQ.0) CALL INCIR(AA,BB,CC,X(1),Y(1),X(3),Y(3),IA,IB,LUN,B,T)
      IF (NUM.NE.0) CALL INCIR(AA,BB,CC,X(2),Y(2),X(3),Y(3),IA,IB,LUN,B,T)
      IF (LUN.GT.0) GOTO 125
      K=2
      GOTO 1
125  IFINE=NUM+LUN
      IF (IFINE.GT.LMAXV) THEN
      WRITE(*,532)
532  FORMAT(' ATTEZIONE !: Area memorizzazione contorno SATURA'/
*15X,'Ci si ARRESTA ! ')
      STOP
      ENDIF
      DO 24 KL=1,LUN
      NUM=NUM+1
      VX(NUM)=IA(KL)
      VY(NUM)=IB(KL)
24   CONTINUE
      K=2
      X(1)=X(2)
      Y(1)=Y(2)
      X(2)=X(3)
      Y(2)=Y(3)
      IF (MJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 132
      WRITE(*,210)
210  FORMAT(5X,'CR          per Terminare ')
      CALL LETCAR(IMOV)
      IU=ICHAR(IMOV)
      IF (IU.NE.13) GOTO 1
      GOTO 133
132  WRITE(*,211)
211  FORMAT(5X,'Tasto 1    per Terminare')

```

```

MM2=-1
CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
MM3=MM2
CALL RESM
IF(MM3.NE.1) GOTO 1
133 NUM=NUM+1
VX(NUM)=VX(1)
VY(NUM)=VY(1)
WRITE(2,77) NUM
WRITE(2,77)(VX(L),VY(L),L=1,NUM)
77  FORMAT(20I4)
GOTO 124
111  write(*,112)
112  format(/' ERRORE. Coordinate -X- fuori range.'/)
stop
222  write(*,223)
223  format(/' ERRORE. Coordinate -Y- fuori range.'/)
stop
333  write(*,334)
334  format(/' ERRORE. Parametro TASTO fuori range.'/)
stop
124  CALL CLS
63  WRITE(*,60)
60  FORMAT(5X,'Ancora ? (y/n): '\)
READ(*,10) SINO
IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 64
IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 63
GOTO 65
64  WRITE(*,66)
66  FORMAT(5X,'Altro File ? (y/n): '\)
READ(*,10) SINO
IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 67
IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 64
IAF=1
GOTO 62
67  CLOSE (2)
IAF=0
GOTO 62
65  CLOSE (2)
END
C *****
SUBROUTINE GAUS(A,N,N1,B)
IMPLICIT INTEGER*2 (I-N)
DIMENSION A(3,4)
C
C RISOLUZIONE DI UN SISTEMA LINEARE 3X4 COL METODO DI GAUSS
C
C
C K1 SERVE PER METTERE A POSTO IL SEGNO DEL DETERMINANTE
K1=0
N3=N-1
M=N1-N
N2=N1-1
C RICERCA IL MASSIMO ELEMENTO IN VALORE ASSOLUTO SULLA PARTE DELLA
C COLONNA J=MA.GE.J. SE ZERO IL DETERMINANTE E' NULLO E SI RITORNA
DO 11 J=1,N
B=0.
DO 3 I=J,N
R=ABS(A(I,J))
IF (R-B) 3,3,5
5  B=R
K=I
3  CONTINUE
IF(B) 21,17,21
C SCAMBIO DI RIGHE SE L'ELEMENTO NON E' SULLA J-MA RIGA
21  IF(K-J) 23,25,23

```

```

23   DO 7 L=J,N1
      A1=A(K,L)
      A(K,L)=A(J,L)
7     A(J,L)=A1
      IF(K1) 235,233,235
233  K1=1
      GOTO 25
235  K1=0
C   TRIANGOLARIZZAZIONE SUPERIORE DELLA MATRICE
25   DO 9 I=J,N2
      A(J,I+1)=A(J,I+1)/A(J,J)
      IF(N-J) 41,31,41
41   DO 110 I=J,N3
      DO 111 K=J,N2
      A(I+1,K+1)=A(I+1,K+1)-A(J,K+1)*A(I+1,J)
111  CONTINUE
110  CONTINUE
11   CONTINUE
C   CALCOLO DEL DETERMINANTE IN CONSIDERAZIONE DI K1
31   B=1.
      DO 666 I=1,N
666  B=B*A(I,I)
      IF(K1.EQ.0) GOTO 33
      B=-B
C   PASSO INDIETRO
33   DO 333 J=1,N3
      DO 333 I=1,M
      N5=N
      K2=N-J
      I3=N-I
      DO 333 K=1,J
      A(K2,K3)=A(K2,K3)-A(N5,K3)*A(K2,N5)
      N5=N5-1
333  CONTINUE
C   TRASFERIMENTO DELLE SOLUZIONI
      DO 27 J=1,M
      K=N+J
      DO 27 I=1,N
27   A(I,J)=A(I,K)
17   RETURN
      END
C   *****
      SUBROUTINE INCIR(AA,BB,CC,XI,YI,XF,YF,IA,IB,LUN,B,MJ)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 VAL,COUNT,BUF(2),XYMN,XYMX
      INTEGER*4 M1,M2,M3,M4
      DIMENSION IA(1),IB(1),KOMD(200)
      CHARACTER*1 SIND,MJ
C
C   ATTIVAZIONE E VETTORIZZAZIONE DEI PUNTI DI INTERPOLAZIONE
C
      ICX=0
      ICY=0
      IS=0
      COUNT=1
      VAL=255
      IF(R.EQ.0.) GOTO 100
      XC=AA/2.
      YC=BB/2.
      R=SQRT(XC*XC+YC*YC-CC)
      IF(R.GT.150000.) GOTO 100
C   CALCOLO DI: ANGOLO INIZIALE (T1) SUO SENNO (ST1) E SUO COSENO (C1)
      ST1=(YC-YI)/R
      C1=(XC-XI)/R
      IF(C1.GT.1.) C1=1.
      IF(C1.LT.-1.) C1=-1.

```

```

      T1=ACOS(C1)
      IF(ST1.GE.0.) GOTO 41
      T1=6.2831852-T1
41  CONTINUE
C   CALCOLO DI: ANGOLO FINALE (T2) SUO SENO (ST2) E SUO COSENO (C2)
      ST2=(YC-YF)/R
      C2=(XC-XF)/R
      IF(C2.GT.1.) C2=1.
      IF(C2.LT.-1.) C2=-1.
      T2=ACOS(C2)
      IF(ST2.GE.0.) GOTO 42
      T2=6.2831852-T2
C   DETERMINAZIONE DEL VERSO DI INTERPOLAZIONE
42  TO=T2-T1
      IF(TO.LT.3.1416.AND.TO.GT.3.1415) GOTO 100
      IF(TO.GT.-3.1416.AND.TO.LT.-3.1415) GOTO 100
      IF(TO.LT.3.1415926) GOTO 60
      TO=TO-6.2831852
      GOTO 61
60  IF(TO.GT.0.) GOTO 62
      IF(TO.GT.-3.1415926) GOTO 61
      TO=TO+6.2831852
      GOTO 62
C   INIZIO CICLO CALCOLO COORDINATE VERSO ANTIORARIO
61  K5=0
      GR=0.00872664
      NP=ABS(TO)/GR
      DO 46 NN=1,NP
      N=NN-1
      TETA=T1-FLOAT(N)*GR
      S=COS(TETA)
      S1=XC-R*S
      IA1=S1+0.5
      S=SIN(TETA)
      S1=YC-R*S
      IB1=S1+0.5
      IF(NN.EQ.1) GOTO 336
      IF(IA1.EQ.IA(K5).AND.IB1.EQ.IB(K5)) GOTO 46
336  K5=K5+1
      IA(K5)=IA1
      IB(K5)=IB1
      KS=KS+1
      CALL ROWRI(IA(K5),IB(K5),COUNT,BUF)
      KOMD(K5)=BUF(1)
      CALL WPIX(IA(K5),IB(K5),VAL)
46  CONTINUE
      GOTO 99
C   INIZIO CICLO CALCOLO COORDINATE VERSO ORARIO
62  K5=0
      GR=0.00872664
      NP=ABS(TO)/GR
      DO 21 NN=1,NP
      N=NN-1
      TETA=T1+FLOAT(N)*GR
      S=COS(TETA)
      S1=XC-R*S
      IA1=S1+0.5
      S=SIN(TETA)
      S1=YC-R*S
      IB1=S1+0.5
      IF(NN.EQ.1) GOTO 337
      IF(IA1.EQ.IA(K5).AND.IB1.EQ.IB(K5)) GOTO 21
337  K5=K5+1
      IA(K5)=IA1
      IB(K5)=IB1
      KS=KS+1

```

```

CALL ROWRI(IA(K5),IB(K5),COUNT,BUF)
KOND(K5)=BUF(1)
CALL WPIX(IA(K5),IB(K5),VAL)
21  CONTINUE
    GOTO 99
C  INTERPOLAZIONE LINEARE
100  DX=XF-XI
    DY=YF-YI
    IF(DX.EQ.0.) GOTO 84
    OM=DY/DX
    OC=YI-XI*OM
    OL=SQRT(DX*DX+DY*DY)
    NPAS=OL/0.1+0.5
    IF(DX) 83,84,88
83   QTS=-1.
    GOTO 80
88   QTS=1.
80   K5=0
    DO 81 IK=1,NPAS
    KI=IK-1
    COM=XI+QTS*FLOAT(KI)*0.1
    IA1=COM+0.5
    IB1=OM*COM+OC+0.5
    IF(IK.EQ.1) GOTO 47
    IF(IA1.EQ.IA(K5).AND.IB1.EQ.IB(K5)) GOTO 81
47   IS=IS+1
    IA(IS)=IA1
    IB(IS)=IB1
    KS=IS+1
    CALL ROWRI(IA(K5),IB(K5),COUNT,BUF)
    KOND(K5)=BUF(1)
    CALL WPIX(IA(K5),IB(K5),VAL)
81   CONTINUE
    GOTO 99
C  RAPPORTO VERTICALE
84   IF(DY.LT.0.) GOTO 86
    ITS=1
87   III=YI+0.5
    IFI=YF+0.5
    IS=0
    DO 85 IK=III,IFI,ITS
    IS=IS+1
    IA(IS)=XI+0.5
    IB(IS)=IK
    KS=IS+1
    CALL ROWRI(IA(K5),IB(K5),COUNT,BUF)
    KOND(K5)=BUF(1)
    CALL WPIX(IA(K5),IB(K5),VAL)
85   CONTINUE
    GOTO 99
86   ITS=-1
    GOTO 87
99   LUH=IS
    IF(MJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 160
8   WRITE(*,7)
7   FORMAT(5X,'Interpolazione OK ? (y/n) : \')
    READ(*,6) SINO
6   FORMAT(A1)
    IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 4
    IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 8
    GOTO 9
160  WRITE(*,2)
2   FORMAT(5X,'Interpolazione OK ? : '/5X,'SI --> Tasto 2',6X.';',
*6X,'NO --> Tasto 4')
    IM2=-1
    CALL CMS COO(ICX,ICY,XYMN,XYMX,XYMN,XYMX,MM2,*111,*222,*333)

```

```

IF(MM2.EQ.2) GOTO 4
IF(MM2.EQ.4) GOTO 9
CALL RESM
GOTO 160
9 DO 3 KJ=1,LUN
VAL=KOMD(KJ)
3 CALL WPIX(IA(KJ),IB(KJ),VAL)
LUN=0
GOTO 4
111 write(*,112)
112 format(/' ERRORE. Coordinate -X- fuori range. '/')
stop
222 write(*,223)
223 format(/' ERRORE. Coordinate -Y- fuori range. '/')
stop
333 write(*,334)
334 format(/' ERRORE. Parametro TASTO fuori range. '/')
124 stop
4 CALL RESM
RETURN
END
C *****
SUBROUTINE RESM
C
C INIZIALIZZAZIONE CON RESET DEL PC-MOUSE
C
M1=0
CALL MOUSES(M1,M2,M3,M4)
M1=7
M3=0
M4=511
CALL MOUSES(M1,M2,M3,M4)
M1=8
M3=0
M4=511
CALL MOUSES(M1,M2,M3,M4)
RETURN
END

```

```

PROGRAM VETL
C
C VETTORIZZAZIONE DI UN CONTORNO CON INTERPOLAZIONE LINEARE
C E SMOOTHING CON FINESTRA DEFINIBILE
C
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 VX(3000),VY(3000),IX(800),IY(800),XYMIN,XYMAX
CHARACTER*64 FNAM,FNAM1
CHARACTER*1 IMOV,SINO,MJ
C
C
IAF=0
62 CALL CLS
LMAXV=2999
NUM=0
WRITE(*,2)
2 FORMAT(//////,25X,'*** VETTORIZZAZIONE DI UN CONTORNO ***'/30X,
*1X,'con Interpolazione Lineare'/)
WRITE(*,3)
3 FORMAT(/5X,'*** INGRESSO DATI ***'/)
7 WRITE(*,5)
5 FORMAT(5X,'Memoria di Quadro (0-3) ?: '\)
READ(*,*,ERR=7) IMM
IF(IMM.LT.0.OR.IMM.GT.3) GOTO 7
CALL DISPL(IMM)
WRITE(*,6)
6 FORMAT(5X,'Ok ? (y/n): '\)
READ(*,10) SINO
10 FORMAT(A1)
IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 7
IF(IAF.EQ.1) GOTO 41
WRITE(*,31)
31 FORMAT(/5X,'Nome File punti interpolati in Uscita ?: '\)
READ(*,32) FNAM
32 FORMAT(A)
OPEN(2,FILE=FNAM,STATUS='NEW')
41 WRITE(*,40)
40 FORMAT(/5X,'Selezione punti da Joy-stick (J) o Mouse (M) ?: '\)
READ(*,10) MJ
K=0
ICX=255
ICY=255
IF(MJ.EQ.'N'.OR.MJ.EQ.'m') GOTO 1
C
C USO DEL JOESTICK
C
WRITE(*,56)
56 FORMAT(/5X,'Istruzioni per l'uso del Joy-stick :'/5X,'ESC',
*4X,'per acquisire Coordinate'/5X,'FRECCE per spostamenti'/5X,
*'BARRA per spostamenti veloci'/5X,'CR per terminare'/)
WRITE(*,4)
4 FORMAT(//22X,'*** INIZIO RILEVAMENTO PUNTI ***'/)
C
CALL GETXY(ICX,ICY)
A=ICX
B=ICY
125 CALL GETXY(ICX,ICY)
C=ICX
D=ICY
CALL INRET(A,B,C,D,IX,IY,LUN,MJ)
IF(LUN.LE.0) GOTO 125
IF INF=NUM+LUN
IF(IFINE.GT.LMAXV) THEN

```



```

WRITE(*,532)
532  FORMAT(' ATTENZIONE !: Area memorizzazione contorno SATURA' /
*15X,'Ci si ARRESTA ! ')
STOP
ENDIF
DO 24 KL=1,LUN
NUM=NUM+1
VX(NUM)=IX(KL)
VY(NUM)=IY(KL)
24  CONTINUE
A=C
B=D
WRITE(*,210)
210  FORMAT(5X,'CR          per Terminare ')
CALL LETCAR(IMOV)
IU=ICHAR(IMOV)
IF(IU.NE.13) GOTO 125
GOTO 133

C
C  USO DEL MOUSE
C
1  XYMIN=0
XYMAX=511
M1=0
CALL Mouses(M1,M2,M3,M4)
IF(M1.EQ.0) THEN
WRITE(*,321)
321  FORMAT(5X,'ERRORE!! : Mouse non installato')
STOP
ENDIF
WRITE(*,46)
46  FORMAT(/5X,'Istruzioni per l''uso del Mouse : '//8X,'TASTO 4   per ',
*1X,'rilevamento Coordinate'/7X,' TASTO 1   per terminare'/)
WRITE(*,4)

C
MM2=-1
CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
A=ICX
B=ICY
126  MM2=-1
CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
C=ICX
D=ICY
CALL INRET(A,B,C,D,IX,IY,LUN,MJ)
IF(LUN.LE.0) GOTO 126
IFINE=NUM+LUN
IF(IFINE.GT.LMAXV) THEN
WRITE(*,532)
ENDIF
DO 25 KL=1,LUN
NUM=NUM+1
VX(NUM)=IX(KL)
VY(NUM)=IY(KL)
25  CONTINUE
A=C
B=D
WRITE(*,211)
211  FORMAT(5X,'Tasto 1   per Terminare')
MM2=-1
CALL MS_COO(ICX,ICY,XYMIN,XYMAX,XYMIN,XYMAX,MM2,*111,*222,*333)
IF(MM2.NE.1) GOTO 126

C
133  NUM=NUM+1
VX(NUM)=VX(1)
VY(NUM)=VY(1)
WRITE(2,77) NUM

```

```

WRITE(2,77)(VX(L),VY(L),L=1,NUM)
77  FORMAT(20I4)
C
C  SMOOTHING
C
395  WRITE(*,666)
666  FORMAT(/5X,'Dimensione della finestra di Smoothing ?: '\)
      READ(*,*,ERR=395) N
      IF(IAF.EQ.1) GOTO 42
      WRITE(*,33)
33   FORMAT(/5X,'Nome File punti filtrati in Uscita ?: '\)
      READ(*,32) FNAM1
      OPEN(3,FILE=FNAM1,STATUS='NEW')
42   CALL SMOOT(VX,VY,NUM,N)
      WRITE(3,77) NUM
      WRITE(3,77)(VX(L),VY(L),L=1,NUM)
      GOTO 124
111  write(*,112)
112  format(/' ERRORE. Coordinate -X- fuori range. '/')
      stop
222  write(*,223)
223  format(/' ERRORE. Coordinate -Y- fuori range. '/')
      stop
333  write(*,334)
334  format(/' ERRORE. Parametro TASTO fuori range. '/')
      stop
124  CALL CLS
63   WRITE(*,60)
60   FORMAT(/5x,'Ancora ? (y/n): '\)
      READ(*,10) SINO
      IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 64
      IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 63
      GOTO 65
64   WRITE(*,66)
66   FORMAT(5X,'Altro File ? (y/n): '\)
      READ(*,10) SINO
      IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 67
      IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 64
      IAF=1
      GOTO 62
67   CLOSE (2)
      CLOSE (3)
      IAF=0
      GOTO 62
65   CLOSE (2)
      CLOSE (3)
      END
C  *****
      SUBROUTINE INRET(A,B,C,D,IX,IY,LUN,MJ)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 VAL,COUNT,BUF(2),XYMN,XYMX
      DIMENSION IX(1),IY(1),KOMD(725)
      CHARACTER*1 SINO,MJ
C
      XYMN=0
      XYMX=511
      VAL=255
      COUNT=1
      IA=A
      IB=B
      IC=C
      ID=D
      LUN=0

```

```

C
C  ATTIVAZIONE E VETTORIZZAZIONE DEI PUNTI DI INTERPOLAZIONE
C

```

```

F=D-B
F1=ABS(F)
G=C-A
G1=ABS(G)
IF(F1.GT.G1) GOTO 1
IPS=1
IF(IA.GT.IC) IPS=-1
DO 2 I=IA,IC,IPS
LUN=LUN+1
IX(LUN)=I
H=FLOAT(I)--A
IY(LUN)=(F*H)/G+B+0.5
CALL ROWRI(IX(LUN),IY(LUN),COUNT,BUF)
KOMD(LUN)=BUF(1)
CALL WPIX(IX(LUN),IY(LUN),VAL)
2 CONTINUE
GOTO 3
1 IPS=1
IF(IB.GT.ID) IPS=-1
DO 4 I=IB,ID,IPS
LUN=LUN+1
IY(LUN)=I
H=FLOAT(I)-B
IX(LUN)=(G*H)/F+A+0.5
CALL ROWRI(IX(LUN),IY(LUN),COUNT,BUF)
KOMD(LUN)=BUF(1)
CALL WPIX(IX(LUN),IY(LUN),VAL)
4 CONTINUE
3 IF(MJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 12
WRITE(*,8)
8 FORMAT(5X,'Interpolazione OK ? (y/n) : '\)
READ(*,5) SINO
5 FORMAT(A1)
IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 6
IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 3
GOTO 11
12 WRITE(*,13)
13 FORMAT(5X,'Interpolazione OK ? : '/5X,'SI --> Tasto 2',6X,':',
*6X,'NO --> Tasto 4')
MM2=-1
CALL MS_COO(ICX,ICY,XYMN,XYMX,XYMN,XYMX,MM2,*111,*222,*333)
IF(MM2.EQ.2) GOTO 6
IF(MM2.EQ.4) GOTO 11
GOTO 12
11 DO 7 KJ=1,LUN
VAL=KOMD(KJ)
7 CALL WPIX(IX(KJ),IY(KJ),VAL)
LUN=0
GOTO 6
111 write(*,112)
112 format(/' ERRORE. Coordinate -X- fuori range. '/')
stop
222 write(*,223)
223 format(/' ERRORE. Coordinate -Y- fuori range. '/')
stop
333 write(*,334)
334 format(/' ERRORE. Parametro TASTO fuori range. '/')
124 stop
6 RETURN
END

```

```

C*****

```

```

SUBROUTINE SMOOT(VX,VY,NUM,N)

```

```

C

```

```

C SUTTOPROGRAMMA DI SMOOTHING

```

```

C

```

```

IMPLICIT INTEGER*2 (I-N)

```

C

INTEGER*2 VX(1),VY(1)

FN=FLOAT(N)

M=N/2

L=M+1

N1=N-2

NFINE=NUM+M-1

K=0

DO 1 I=1,N1

K=K+1

NUM1=NUM+K

VX(NUM1)=VX(I+1)

VY(NUM1)=VY(I+1)

1

CONTINUE

K=0

DO 2 I=L,NFINE

K=K+1

I1=I-M

I2=I+M

X=0.

Y=0.

DO 3 J=I1,I2

X=X+FLOAT(VX(J))

Y=Y+FLOAT(VY(J))

3

CONTINUE

VX(K)=X/FN+0.5

VY(K)=Y/FN+0.5

2

CONTINUE

VX(NUM)=VX(1)

VY(NUM)=VY(1)

RETURN

END

```

PROGRAM SMOOT
C
C SMOOTHING MONODIMENSIONALE CON FINESTRA DEFINIBILE.
C
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 VX(3000),VY(3000),VAL
CHARACTER*64 FNAM,FNAM1
CHARACTER*1 SINO
C
IAF=0
62 CALL CLS
WRITE(*,20)
20 FORMAT(/,31X,'*** PROGRAMMA DI SMOOTHING MONODIMENSIONALE ***'//)
WRITE(*,21)
21 FORMAT(5X,'*** INGRESSO DATI ***'//)
VAL=255
IF(IAF.EQ.1) GOTO 30
WRITE(*,8)
8 FORMAT(/5X,'Nome File punti acquisiti : ',\ )
READ(*,7) FNAM
7 FORMAT(A)
OPEN(2,FILE=FNAM,STATUS='OLD')
30 READ(2,13) NUM
READ(2,13) (VX(I),VY(I),I=1,NUM)
13 FORMAT(20I4)
15 WRITE(*,14)
14 FORMAT(/5X,'Dimensione finestra di smoothing : '\ )
READ(*,*,ERR=15) N
C
NN1=N/2
NN2=N-(NN1*2)
IF(NN2.EQ.0) GOTO 15
C
VAL=255
FN=FLOAT(N)
M=N/2
L=M+1
N1=N-2
NFINE=NUM+M-1
K=0
DO 1 I=1,N1
K=K+1
NUM1=NUM+K
VX(NUM1)=VX(I+1)
VY(NUM1)=VY(I+1)
1 CONTINUE
K=0
DO 2 I=L,NFINE
K=K+1
I1=I-M
I2=I+M
X=0.
Y=0.
DO 3 J=I1,I2
X=X+FLOAT(VX(J))
Y=Y+FLOAT(VY(J))
3 CONTINUE
VX(K)=X/FN+0.5
VY(K)=Y/FN+0.5
2 CONTINUE
VX(NUM)=VX(1)
VY(NUM)=VY(1)
WRITE(*.23)

```

```

23  FORMAT(/5X,'*** USCITA RISULTATI ***'/)
    IF(IAF.EQ.1) GOTO 31
    WRITE(*,27)
27  FORMAT(/5X,'Nome File punti filtrati ? : '\)
    READ(*,7) FNAM1
    OPEN(3,FILE=FNAM1,STATUS='NEW')
31  WRITE(3,13) NUM
    WRITE(3,13) (VX(I),VY(I),I=1,NUM)
10  WRITE(*,9)
9   FORMAT(/5X,'Memoria di Quadro (0-3) ? : '\)
    READ(*,*,ERR=10) IMM
    IF(IMM.LT.0.OR.IMM.GT.3) GOTO 10
    CALL DISPL(IMM)
    WRITE(*,11)
11  FORMAT(5X,'Ok ? (y/n) : '\)
    READ(*,12) SINO
12  FORMAT(A1)
    IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 10
26  WRITE(*,24)
24  FORMAT(/5X,'Cancellazione Memoria (y/n) : '\)
    READ(*,12) SINO
    IF(SINO.EQ.'N'.OR.SINO.EQ.'n') GOTO 25
    IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 26
    CALL SET(0)
25  DO 22 I=2,NUM
    I1=I-1
    CALL LINE(VX(I1),VY(I1),VX(I),VY(I),VAL)
22  CONTINUE
    CALL CLS
63  WRITE(*,60)
60  FORMAT(/5X,'Ancora ? (y/n) : '\)
    READ(*,12) SINO
    IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 64
    IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 63
    GOTO 65
64  WRITE(*,66)
66  FORMAT(5X,'Altro File ? (y/n) : '\)
    READ(*,12) SINO
    IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 67
    IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 64
    IAF=1
    GOTO 62
67  CLOSE (2)
    CLOSE (3)
    IAF=0
    GOTO 62
65  CLOSE (2)
    CLOSE (3)
    END

```

PROGRAM FILRUN

```

C
C PROGRAMMA DI FILLING DI SUPERFICI POLIGONALI CONVESSE,
C CONCAVE, E CON PRESENZA DI BUCHI DOVUTI A NIDIFICAZIONE.
C
  IMPLICIT INTEGER*2 (I-N)
  INTEGER*2 IX(2000), IY(2000), VAL, VAL1, XYMIN, XYMAX
  INTEGER*4 AREA, MISPER
  CHARACTER*64 FNAM
  CHARACTER*1 SINO, MJ, MF

C
100 MISPER=0
  CALL CLS
  WRITE(*,22)
22  FORMAT(///,28X,'*** PROGRAMMA DI FILLING ***')
  WRITE(*,21)
21  FORMAT(/5X,'*** INGRESSO DATI ***'/)
25  WRITE(*,23)
23  FORMAT(/5X,'Memoria di quadro (M) o File di disco (F) ? : '\)
  READ(*,12) MF
  IF(MF.EQ.'M'.OR.MF.EQ.'m') GOTO 6
  IF(MF.EQ.'F'.OR.MF.EQ.'f') GOTO 24
  GOTO 25
24  WRITE(*,8)
  8  FORMAT(/5X,'Nome File Ingresso punti di contorno ? : '\)
  READ(*,7) FNAM
  7  FORMAT(A)
  OPEN(2,FILE=FNAM,STATUS='OLD')
  5  WRITE(*,4)
  4  FORMAT(/5X,'Numero di Poligoni di contorno ? : '\)
  READ(*,*,ERR=5) NNC
  6  WRITE(*,2)
  2  FORMAT(/5X,'Valore luminanza di rappresentazione contorni ? : '\)
  READ(*,*,ERR=6) VAL
77  WRITE(*,66)
66  FORMAT(5X,'Valore luminanza del riempimento ? : '\)
  READ(*,*,ERR=77) VAL1
  IF(MF.EQ.'F'.OR.MF.EQ.'f') WRITE(*,27)
27  FORMAT(/5X,'*** USCITA RISULTATI ***'/)
10  WRITE(*,9)
  9  FORMAT(/5X,'Memoria di Quadro (0-3) ? : '\)
  READ(*,*,ERR=10) IMM
  IF(IMM.LT.0.OR.IMM.GT.3) GOTO 10
  CALL DISPL(IMM)
  WRITE(*,11)
11  FORMAT(5X,'Ok ? (y/n) : '\)
  READ(*,12) SINO
12  FORMAT(A1)
  IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 10
  IF(MF.EQ.'M'.OR.MF.EQ.'m') GOTO 26
  DO 13 JJ=1,NNC
  READ(2,1) N
  READ(2,1) (IX(I),IY(I),I=1,N)
  1  FORMAT(20I4)
  DO 3 I=2,N
  I1=I-1
  LUNG=SQRT((IX(I)-IX(I1))**2+(IY(I)-IY(I1))**2)
  CALL LINE(IX(I1),IY(I1),IX(I),IY(I),VAL)
  MISPER=MISPER+LUNG
  3  CONTINUE
13  CONTINUE
  CLOSE(2)
24  IXC=255

```

```

-- IYO=255
396 WRITE(*,395)
395 FORMAT(/5X,'Usa del Joy-stick (J) o del Mouse (M) ? : '\)
READ(*,12) MJ
IF(MJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 399
IF(MJ.NE.'J'.AND.MJ.NE.'j') GOTO 396
WRITE(*,398)
398 FORMAT(5X,'Seleziona col Joy-stick un punto interno al contorno')
CALL GETXY(IXO,IYO)
GOTO 397
399 XYMIN=0
XYMAX=511
CALL RESM
WRITE(*,31)
31 FORMAT(/5X,'Seleziona col Mouse un punto interno al contorno')
M2=-1
CALL MS_COO(IXO,IYO,XYMIN,XYMAX,XYMIN,XYMAX,M2,*111,*222,*333)
397 CALL FILR4(IXO,IYO,VAL,VAL1,AREA)
CALL CLS
IF(MF.EQ.'F'.OR.MF.EQ.'f') WRITE(*,20) MISPER,AREA
20 FORMAT(/5X,'Misura Perimetro del Contorno (in Pixels) = ',I6/
*5X,'Misura Area del Riempimento (in Pixels) = ',I6//)
43 WRITE(*,40)
40 FORMAT(/5X,'Ancora ? (y/n): '\)
READ(*,12) SINO
IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 42
IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 43
GOTO 44
42 WRITE(*,41)
41 FORMAT(/5X,'Cancellazione della Memoria ? (y/n): '\)
READ(*,12) SINO
IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 45
IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 42
GOTO 100
45 CALL SET(0)
GOTO 100
111 write(*,112)
112 format(/' ERRORE. Coordinate -X- fuori range.'/)
stop
222 write(*,223)
223 format(/' ERRORE. Coordinate -Y- fuori range.'/)
stop
333 write(*,334)
334 format(/' ERRORE. Parametro TASTO fuori range.'/)
stop
44 CONTINUE
END

```

C *****

```

SUBROUTINE FILR4(IXO,IYO,VAL,VAL1,AREA)
IMPLICIT INTEGER*2 (I-N)
DIMENSION IST(1000,2)
INTEGER*2 VAL,VAL1,BUF(2),CONT
INTEGER*4 AREA

```

C

```

AREA=0
K=0
CONT=1
ITMP=IXO
IX=IXO
IY=IYO
3 AREA=AREA+1
CALL WPIX(IX,IY,VAL1)
IX=IX+1
CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 3
IA=IX-1

```



```

      IX=ITMP-1
4     CALL ROWRI(IX,IY,CONT,BUF)
      IF(BUF(1).EQ.VAL) GOTO 5
16    AREA=AREA+1
      CALL WPPIX(IX,IY,VAL1)
      IX=IX-1
      GOTO 4
5     IB=IX+1
      IY=IY-1
      IX=IB
6     CALL ROWRI(IX,IY,CONT,BUF)
      IF(BUF(1).NE.VAL.AND.BUF(1).NE.VAL1) GOTO 7
10    IX=IX+1
      GOTO 6
7     IF(IX.GT.IA) GOTO 11
8     IX=IX+1
      CALL ROWRI(IX,IY,CONT,BUF)
      IF(BUF(1).NE.VAL) GOTO 8
      K=K+1
      IST(K,1)=IX-1
      IST(K,2)=IY
      GOTO 10
11    IY=IY+2
      IX=IB
9     CALL ROWRI(IX,IY,CONT,BUF)
      IF(BUF(1).NE.VAL.AND.BUF(1).NE.VAL1) GOTO 12
15    IX=IX+1
      GOTO 9
12    IF(IX.GT.IA) GOTO 13
14    IX=IX+1
      CALL ROWRI(IX,IY,CONT,BUF)
      IF(BUF(1).NE.VAL) GOTO 14
      K=K+1
      IST(K,1)=IX-1
      IST(K,2)=IY
      GOTO 15
13    IF(K.EQ.0) RETURN
      IX=IST(K,1)
      IY=IST(K,2)
      K=K-1
      IA=IX
      GOTO 16
      END

```

```

C *****
C SUBROUTINE RESM

```

```

C
C INIZIALIZZAZIONE CON RESET DEL PC-MOUSE
C

```

```

      M1=0
      CALL MOUSES (M1,M2,M3,M4)
      IF(M1.EQ.0) THEN
        WRITE(*,321)
321  FORMAT(5X,'ERRORE!! : Mouse non installato')
      STOP
      ENDIF
      M1=7
      M3=0
      M4=511
      CALL MOUSES(M1,M2,M3,M4)
      M1=8
      M3=0
      M4=511
      CALL MOUSES(M1,M2,M3,M4)
      RETURN
      END

```

PROGRAM ESTRAZ

```

C
C ESTRAZIONE DA UN IMMAGINE DI UN PARTICOLARE CONTORNATO
C
  IMPLICIT INTEGER*2 (I-N)
  INTEGER*2 IX(2000),IY(2000),VAL,XYMIN,XYMAX
  INTEGER*4 AREA,MISPER,ISTOG(256),MAXX
  CHARACTER*64 FNAM
  CHARACTER*26 IEI
  CHARACTER*1 SINO,MJ

C
  IEI='    ***   I.E.I.   *** '

100 MISPER=0
  CALL CLS
  WRITE(*,1)
  1  FORMAT(/,,21X,'*** PROGRAMMA DI ESTRAZIONE PARTICOLARE ***')
  WRITE(*,2)
  2  FORMAT(/5X,'*** INGRESSO DATI ***'/)
  3  WRITE(*,4)
  4  FORMAT(5X,'Prima Memoria di Quadro (0-3) ? : '\)
  READ(*,*,ERR=3) IMM
  IF(IMM.LT.0.OR.IMM.GT.3) GOTO 3
  CALL DISPL(IMM)
  WRITE(*,5)
  5  FORMAT(5X,'Ok ? (y/n) : '\)
  READ(*,6) SINO
  6  FORMAT(A1)
  IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 3
  8  WRITE(*,7)
  7  FORMAT(5X,'Memoria di Quadro di lavoro (0-3) ? : '\)
  READ(*,*,ERR=8) IM1
  IF(IM1.LT.0.OR.IM1.GT.3.OR.IM1.EQ.IMM) GOTO 8
  CALL DISPL(IM1)
  WRITE(*,5)
  READ(*,6) SINO
  IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 8
  VAL=255
  WRITE(*,9)
  9  FORMAT(/5X,'Nome File Ingresso dei punti di contorno : '\)
  READ(*,10) FNAM
  10  FORMAT(A)
  OPEN(3,FILE=FNAM,STATUS='OLD')
  60  WRITE(*,61)
  61  FORMAT(/5X,'Numero di Poligoni di contorno ? : '\)
  READ(*,*,ERR=60) NNC
  VAL=255
  DO 62 JL=1,NNC
  READ(3,11) N
  READ(3,11) (IX(I),IY(I),I=1,N)
  11  FORMAT(20I4)
  DO 13 I=2,N
  I1=I-1
  LUNG=SQRT((IX(I)-IX(I1))**2+(IY(I)-IY(I1))**2)
  CALL LINE(IX(I1),IY(I1),IX(I),IY(I),VAL)
  MISPER=MISPER+LUNG
  13  CONTINUE
  62  CONTINUE
  CLOSE (3)
  IX0=255
  IY0=255
  396  WRITE(*,395)
  395  FORMAT(/5X,'Usa del Joy-stick (J) o del Mouse (M) ? : '\)
  READ(*,6) MJ

```

```

IF(MJ.EQ.'M'.OR.MJ.EQ.'m') GOTO 399
IF(MJ.NE.'J'.AND.MJ.NE.'j') GOTO 396
WRITE(*,398)
398 FORMAT(5X,'Seleziona col Joy-stick un punto interno al contorno')
CALL GETXY(IXO,IYO)
GOTO 397
399 XYMIN=0
XYMAX=511
CALL RESM
WRITE(*,31)
31 FORMAT(5X,'Seleziona col Mouse un punto interno al contorno')
M2=-1
CALL MS_COO(IXO,IYO,XYMIN,XYMAX,XYMIN,XYMAX,M2,*111,*222,*333)
397 WRITE(*,14)
14 FORMAT(/5X,'*** USCITA DATI ***'/)
15 WRITE(*,16)
16 FORMAT(5X,'Memoria di Quadro (0-3) ?: '\)
READ(*,*,ERR=15) IM2
IF(IM2.LT.0.OR.IM2.GT.3) GOTO 15
IF(IM2.EQ.IMM.OR.IM2.EQ.IM1) GOTO 15
CALL DISPL(IM2)
WRITE(*,5)
READ(*,6) SINO
IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 15
CALL FILR4(IMM,IM1,IM2,IXO,IYO,AREA,ISTOG)
52 WRITE(*,50)
50 FORMAT(/5X,'Disegno Istogramma del particolare (y/n) ?: '\)
READ(*,6) SINO
IF(SINO.EQ.'N'.OR.SINO.EQ.'n') GOTO 51
IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 52
53 WRITE(*,16)
READ(*,*,ERR=53) IM3
IF(IM3.LT.0.OR.IM3.GT.3) GOTO 53
CALL DISPL(IM3)
WRITE(*,5)
READ(*,6) SINO
IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 53
WRITE(*,54)
54 FORMAT(/5X,'Azzeramento (y/n) ?: '\)
READ(*,6) SINO
IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') CALL SET(0)
DO 55 I=1,256
IF(ISTOG(I).NE.0) GOTO 56
55 CONTINUE
56 MIN=I-1
DO 57 I=256,1,-1
IF(ISTOG(I).NE.0) GOTO 58
57 CONTINUE
58 MAX=I-1
MAXX=0
DO 63 I=1,256
IF(ISTOG(I).GT.MAXX) MAXX=ISTOG(I)
63 CONTINUE
WRITE(*,59) MIN,MAX
59 FORMAT(/5X,'Valore minimo del particolare = ',I4/5X,
*'Valore massimo del particolare = ',I4/)
CALL DHISTO(MAXX,50,450,180,255,300,ISTOG)
IT1=22
IT2=56
IT3=26
IT4=2
IT5=0
IT6=255
CALL TEXT(IT1,IT2,IEI,IT3,IT4,IT5,IT6)
51 WRITE(*,20) MISPER,AREA
20 FORMAT(/5X,'Misura Perimetro del Contorno (in Pixels) = ',I6/

```

```

*5X,'Misura Area del Riempimento (in Pixels)      = ',16//)
23  WRITE(*,21)
21  FORMAT(5X,'Scrittura Misure su Memoria di Quadro ? (y/n) : '\)
    READ(*,6) SINO
    IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 22
    IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 23
    GOTO 124
22  CALL WRTA(MJ)
    GOTO 124
111  write(*,112)
112  format(/' ERRORE. Coordinate -X- fuori range.'/)
    stop
222  write(*,223)
223  format(/' ERRORE. Coordinate -Y- fuori range.'/)
    stop
333  write(*,334)
334  format(/' ERRORE. Parametro TASTO fuori range.'/)
    stop
124  CALL CLS
43  WRITE(*,40)
40  FORMAT(/5X,'Altra Elaborazione ? (y/n): ',#)
    READ(*,6) SINO
    IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 42
    IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 43
    GOTO 44
42  WRITE(*,41)
41  FORMAT(/5X,'Cancellazione delle Memorie di Lavoro e di Uscita',
*1X,'? (y/n): '\)
    READ(*,6) SINO
    IF(SINO.EQ.'Y'.OR.SINO.EQ.'y') GOTO 45
    IF(SINO.NE.'N'.AND.SINO.NE.'n') GOTO 42
    GOTO 100
45  CALL SETSCR(IM2)
    CALL SET(0)
    CALL SETSCR(IM1)
    CALL SET(0)
    GOTO 100
44  CONTINUE
    END
C  *****
    SUBROUTINE FILR4(IMM,IM1,IM2,IXO,IYO,AREA,ISTOG)
    IMPLICIT INTEGER*2 (I-N)
    DIMENSION IST(1000,2)
    INTEGER*2 VAL,BUF(2),CONT
    INTEGER*4 AREA,ISTOG(1)
C
    DO 17 JJ=1,256
17  ISTOG(JJ)=0
    AREA=0
    CALL SETSCR(IM1)
    K=0
    CONT=1
    VAL=255
    ITMP=IXO
    IX=IXO
    IY=IYO
3  CALL WPIX(IX,IY,VAL)
    CALL SETSCR(IMM)
    CALL ROWRI(IX,IY,CONT,BUF)
    CALL SETSCR(IM2)
    AREA=AREA+1
    JJ=BUF(1)+1
    ISTOG(JJ)=ISTOG(JJ)+1
    CALL WPIX(IX,IY,BUF(1))
    CALL SETSCR(IM1)
    IX=IX+1

```

```

CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 3
IA=IX-1
IX=ITMP-1
4 CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).EQ.VAL) GOTO 5
16 CALL WPIX(IX,IY,VAL)
CALL SETSCR(IMM)
CALL ROWRI(IX,IY,CONT,BUF)
CALL SETSCR(IM2)
AREA=AREA+1
JJ=BUF(1)+1
ISTOG(JJ)=ISTOG(JJ)+1
CALL WPIX(IX,IY,BUF(1))
CALL SETSCR(IM1)
IX=IX-1
GOTO 4
5 IB=IX+1
IY=IY-1
IX=IB
6 CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 7
10 IX=IX+1
GOTO 6
7 IF(IX.GT.IA) GOTO 11
8 IX=IX+1
CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 8
K=K+1
IST(K,1)=IX-1
IST(K,2)=IY
GOTO 10
11 IY=IY+2
IX=IB
9 CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 12
15 IX=IX+1
GOTO 9
12 IF(IX.GT.IA) GOTO 13
14 IX=IX+1
CALL ROWRI(IX,IY,CONT,BUF)
IF(BUF(1).NE.VAL) GOTO 14
K=K+1
IST(K,1)=IX-1
IST(K,2)=IY
GOTO 15
13 IF(K.EQ.0) RETURN
IX=IST(K,1)
IY=IST(K,2)
K=K-1
IA=IX
GOTO 16
END
C *****
SUBROUTINE WRTA(MJ)
  implicit integer*2(i-n)
  character*56 st1
  character*1 ies,mj
  integer*2 rpix,xymin,xymax
100 WRITE(*,4)
4 FORMAT(5X,'Memoria di Quadro (0-3) ? : '\)
READ(*,*,ERR=100) IM
IF(IM.LT.0.OR.IM.GT.3) GOTO 100
CALL DISPL(IM)
WRITE(*,5)
5 FORMAT(5X,'Ok ? (v/n): '\)

```

```

-
6 READ(*,6) SINO
  FORMAT(A1)
  IF(SINO.NE.'Y'.AND.SINO.NE.'y') GOTO 100
  il=255
  idim=1
  idir=0
  nch=56
    write(*,15)
15   format(5x,'- Inserire il testo -'/56('C*'))
    read(*,16) st1
16   format(a)
    ix=255
    iy=255
    if(mj.eq.'M'.or.mj.eq.'m') goto 3
    write(*,1)
1   format(/5x,'Determina col Joy-stick l''inizio di scrittura'/)
    call getxy(ix,iy)
    goto 2
3   write(*,444)
444  format(/5x,'Determina col Mouse l''inizio di scrittura',/)
    xmin=0
    xmax=511
    call resm
    m2=-1
    call ms_coo(ix,iy,xmin,xmax,xmin,xmax,m2,*111,*222.*333)
2   call text(ix,iy,st1,nch,idim,idir,il)
    write(*,40)
40   format(/5ix,'Ancora ? (y/n): ',%)
    read(*,14) ies
14   format(a1)
    if (ies.eq.'y' .OR. ies.eq.'Y') goto 100
    goto 415
111  write(*,112)
112  format(/' ERRORE. Coordinate -X- fuori range.'/)
    stop
222  write(*,223)
223  format(/' ERRORE. Coordinate -Y- fuori range.'/)
    stop
333  write(*,334)
334  format(/' ERRORE. Parametro TASTO fuori range.'/)
    stop
415  continue
    end
C *****
  SUBROUTINE RESM
C
C INIZIALIZZAZIONE CON RESET DEL PC-MOUSE
C
  m1=0
  call mouses(m1,m2,m3,m4)
  if(m1.eq.0) then
    write(*,1)
1   format(/5x,'ERRORE!! : Mouse non installato')
    stop
  endif
  m1=7
  m3=0
  m4=511
  call mouses(m1,m2,m3,m4)
  m1=8
  m3=0
  m4=511
  call mouses(m1,m2,m3,m4)
  return
  end

```

