

*Consiglio Nazionale delle Ricerche*

**ISTITUTO DI ELABORAZIONE  
DELLA INFORMAZIONE**

**PISA**

IMPLEMENTAZIONE DI UN ALGORITMO  
PER L'ELIMINAZIONE DELLE HIDDEN-LINE

**Enrico Fantini**

Nota interna B4-50  
Settembre 1986

**IMPLEMENTAZIONE DI UN ALGORITMO  
PER L'ELIMINAZIONE DELLE HIDDEN-LINE**

Enrico Fantini

La necessita` di rappresentare delle matrici di dati in un modo piu` adeguato alla interpretazione ha portato alla stesura di un programma che implementa un algoritmo per l'eliminazione delle hidden-line.

Tramite tale algoritmo e` possibile determinare quali righe della matrice, rappresentante una qualsiasi superficie, siano visibili da un particolare punto di vista.

L'algoritmo in questione e` stato implementato su un elaboratore GOULD 32/27 con sistema operativo MPX-32 Rev. 3.2C in linguaggio FORTRAN-77.

Il programma fa uso di routines esterne per la gestione dell'I/O verso le periferiche grafiche e per l'organizzazione dei dati su disco. Dette routines sono residenti nelle librerie GEPIM.LIB e SVP.LIB poste nella direttrice LIBRERIE del disco di sistema.

Come risorse hardware il programma necessita oltre che di una memoria estesa di almeno 12 pagine (96 Kbytes), di alcune periferiche grafiche per l'uscita dell'elaborato. Tali periferiche sono:

- plotter piano Calcomp-81 a media velocita` (max 30 cm/sec), con formato carta di 338 mm per l'asse X e 280 mm per l'asse Y con una risoluzione di 0.1 mm, collegato all'elaboratore tramite una interfaccia seriale RS-232;

- terminale grafico Tektronix 4014 a fosfori verdi, con formato video di 1024 punti per l'asse X e 768 punti per l'asse Y, collegato all'elaboratore tramite una interfaccia seriale RS-232;
- stazione grafico/pittorica SVP1000, formata da 6 memorie quadro di 512x512 pixel con la possibilita` di avere 256 colori o livelli di grigio, collegata all'elaboratore tramite interfaccia parallela MPCII (MultiPurpose Custom Interface).

Vale la pena precisare che data la particolare struttura di chiamata del programma (tramite una macro JCL), al momento del RUN vengono allocate soltanto le risorse hardware che si ritengono necessarie al buon fine dell'operazione richiesta dall'utilizzatore.

Questo vuol dire che se, per esempio, si desiderasse l'uscita dell'elaborato su plotter, la procedura non impegnerebbe ne` il Tektronix ne` l'SVP1000 che verrebbero lasciati liberi per gli altri utenti.

### DESCRIZIONE

L'algoritmo permette di avere una visione con proiezione a 45 gradi dell'oggetto rappresentato nella matrice. Quest'ultima non puo` avere dimensioni piu` grandi di 128x128 e risiede interamente in memoria centrale al fine di accelerare i tempi di calcolo.

Facendo il confronto tra le coordinate del punto preso in esame e quelle del punto di massimo locale si determinano le sole parti visibili della riga matrice che contiene detto punto, iterando il processo per il

numero delle righe della matrice, e poi per le colonne, si ottiene l'intera visione della superficie contenuta nella matrice.

### GUIDA ALL'UTILIZZO

Evocando da TSM la macro HIDLINE viene per prima cosa chiesto il tipo di periferica che si vuole usare per l'uscita dell'elaborato.

Le risposte possibili sono:

- il carattere P per il plotter
- il carattere T per il tektronix
- il carattere S per l'SVP1000
- il carattere F per la chiusura dell'intera sessione;

qualsiasi carattere diverso da quelli descritti genera un messaggio di errore.

Prendiamo ora in considerazione i singoli casi descrivendo la prima parte del programma in comune a tutte le periferiche di uscita.

Il programma si apre chiedendo il numero di punti totali sull'asse X. Con questo si intende l'ingombro che viene determinato dalla grandezza max dell'asse X, espresso in numero di punti.

La massima grandezza accettata e` di 2000 punti per il plotter, di 1000 punti per il tektronix e di 512 punti per l'SVP1000, praticamente e` il doppio della misura del lato del tubo contenente l'intera scena tridimensionale. Per default viene preso il valore massimo per ogni singola periferica.

Vengono poi richiesti i dati relativi al file su disco contenente la matrice; si ricorda che tali dati devono essere di tipo intero su 2 bytes e scritti con le routines per la gestione dati su disco GEPIM.

Il passo successivo e' la determinazione della sottomatrice di ingresso fissando il numero di righe e di colonne da prendere in considerazione. A questo punto viene chiesto il guadagno dell'asse Z il quale altro non e' che un fattore di moltiplicazione che permette di riscaldare a piacimento i dati a disposizione. Il default di questo parametro e' 100.

Viene poi determinato il punto di vista rispetto alla prima riga della matrice contenente la superficie.

I possibili punti di vista sono:

- anteriore (partendo dalla prima riga)
- posteriore (partendo dall'ultima riga)
- destra (partendo dall'ultima colonna)
- sinistra (partendo dalla prima colonna).

Per default viene presa in considerazione la vista anteriore.

A questo punto termina la parte in comune a tutte le periferiche e il programma prosegue in maniera diversa a seconda del tipo di uscita richiesto.

#### Uscita plotter

Viene chiesto il numero della penna con la quale si vuole eseguire il disegno (da 1 a 8). Default penna 1.

Effettuata la scelta il programma inizia a disegnare, data la non elevata velocita` della periferica questa fase potrebbe risultare abbastanza lunga.

#### Uscita Tektronix

Il programma inizia subito ad eseguire il disegno senza chiedere altro.

#### Uscita SVP1000

L'uscita su questa periferica permette diverse opportunita` dovute alla possibilita` di intervenire sulle varie scale del colore.

La prima cosa che viene chiesta e` la conferma alla possibilita` di effettuare un RESET hardware della periferica; se si risponde affermativamente vengono cancellate tutte le memorie di quadro e i registri di stato della periferica stessa con la conseguente perdita di tutte le immagini presenti in essa.

A questo punto viene chiesto se si vogliono caricare le tavole di look-up per la determinazione dei colori. A risposta affermativa vengono richiesti i parametri per il reperimento dei files contenenti le tre tavole (R,G,B) residenti su disco; se non si desidera caricare delle tavole questa parte non viene assolutamente eseguita e si prosegue con la richiesta del valore dello sfondo del disegno.

Sono ammessi tutti i valori compresi tra 0 e 255 (default = 0).

Viene poi chiesto il tipo di colorazione del disegno da scegliere fra 3 possibilita`:

- 1 - monocromatico, cioè tutto il disegno verrà eseguito con un solo colore
- 2 - colori corrispondenti alle tavole di look-up
- 3 - a fasce di valori corrispondenti ai valori posti sulle tavole di look-up (default = 1).

Se si sceglie il tipo 1 viene chiesto il valore di default (default= 0) con cui eseguire il disegno dopo di che il programma inizia a tracciare il disegno.

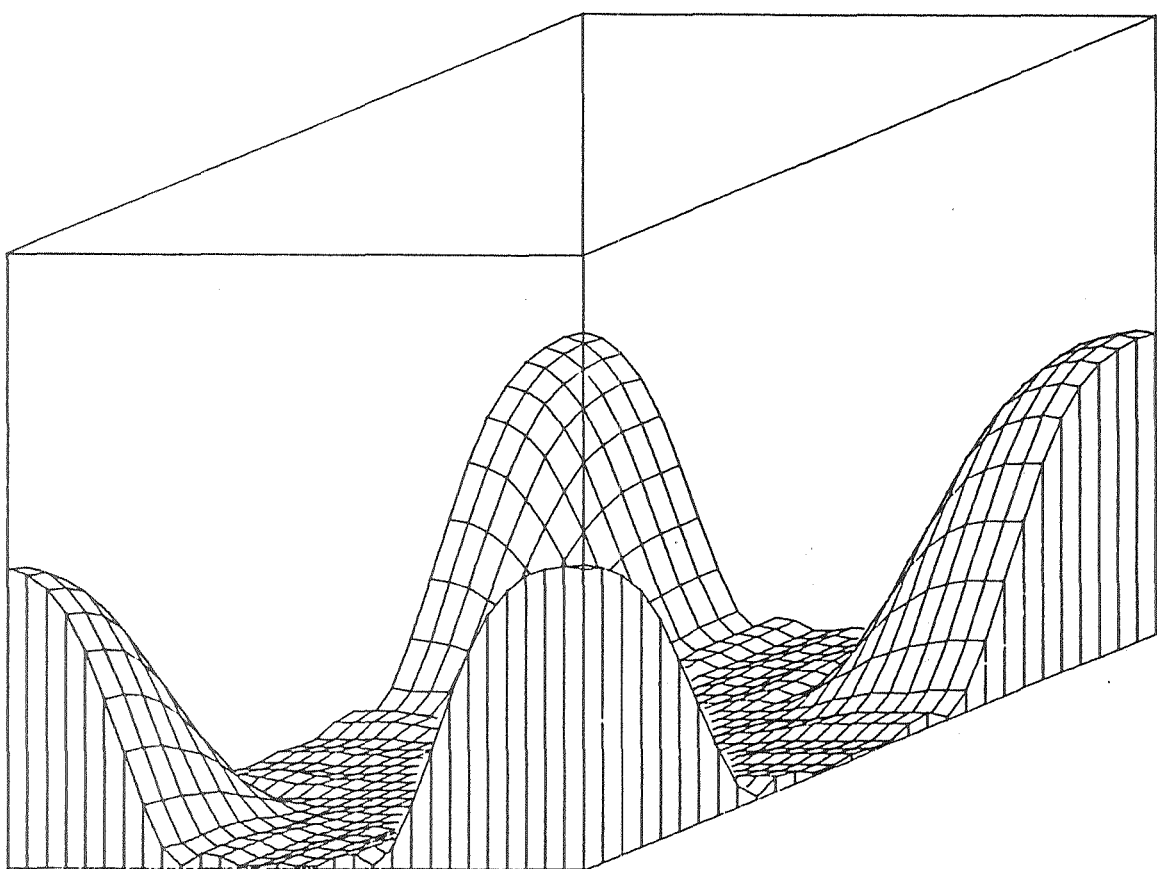
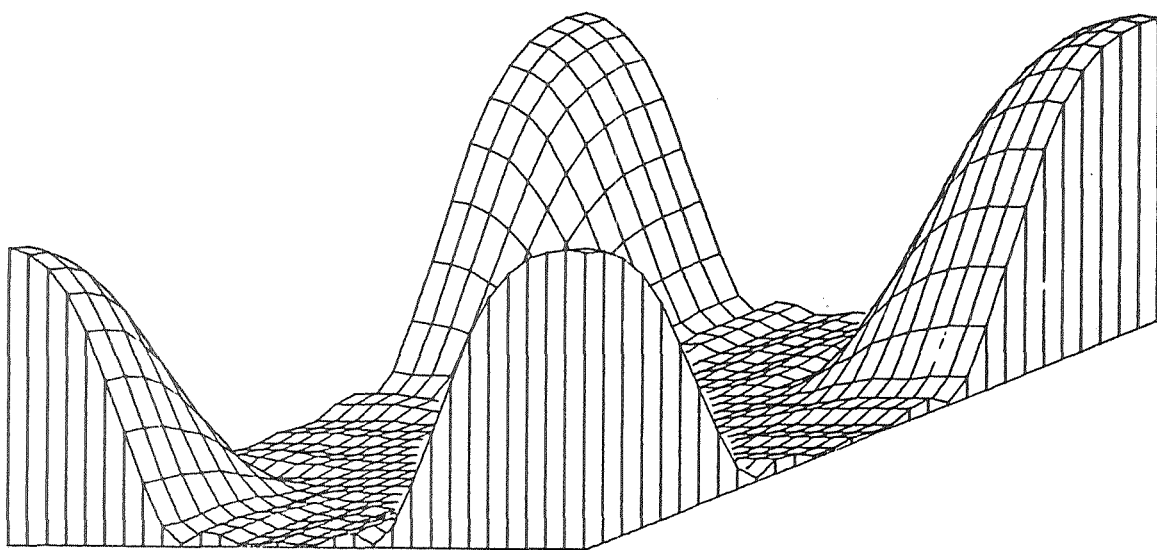
Se si sceglie il tipo 2 viene eseguito il disegno utilizzando come colori i valori posti nelle tavole di look-up all'indirizzo indentificato dal valore del punto da disegnare.

Se si sceglie il tipo 3 viene chiesto il valore di default per eventuali pixel che eccedessero come valore le fasce che verranno determinate in seguito (default = 0).

Le fasce vengono determinate chiedendo all'operatore i valori min e max su cui articularle.

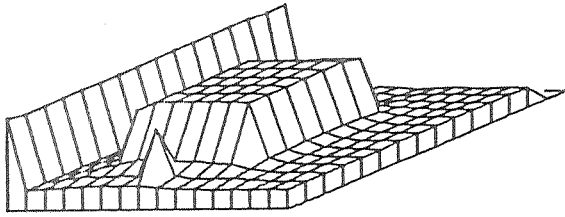
Per ogni fascia viene poi chiesto il massimo valore rappresentabile dalla fascia stessa e il valore delle tavole di look-up (colore) che rappresenterà tale fascia. A questo punto inizia la fase di tracciatura del disegno tenendo conto di tutti i parametri impostati in precedenza.

In tutti e tre i casi alla fine dell'esecuzione del disegno viene data la possibilità di tracciare il cubo che racchiude l'intera scena ottenuta, in qualsiasi caso il passo successivo del programma è di ripartire dall'inizio chiedendo di nuovo la periferica che si vuole usare per l'uscita di un altro eventuale disegno; a questo punto è possibile uscire dalla procedura digitando la lettera F.

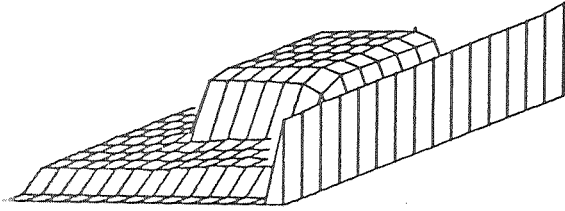


Es. 1 Rappresentazione della stessa superficie.  
Nel secondo caso è stato tracciato anche il  
cubo circoscritto alla scena ottenuta.

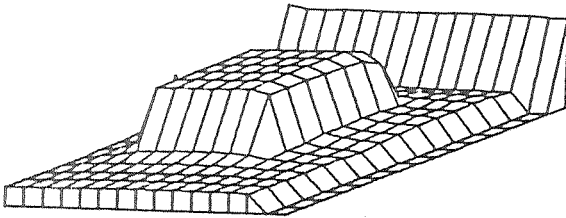




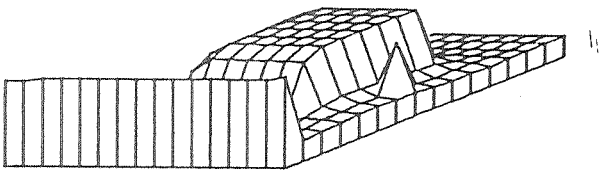
Vista anteriore



Vista posteriore

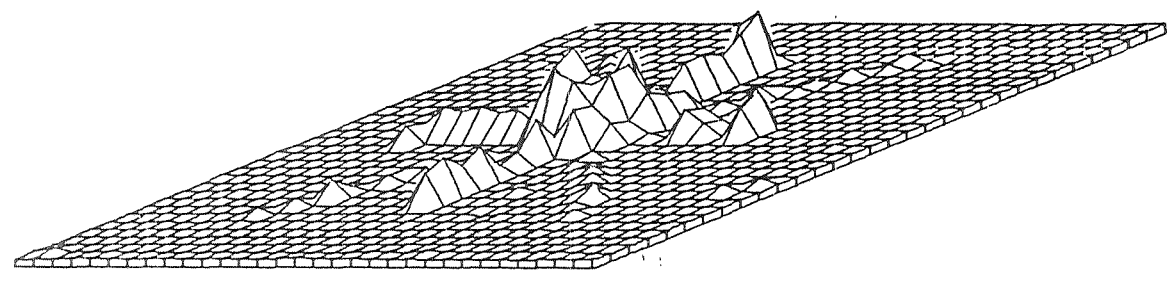
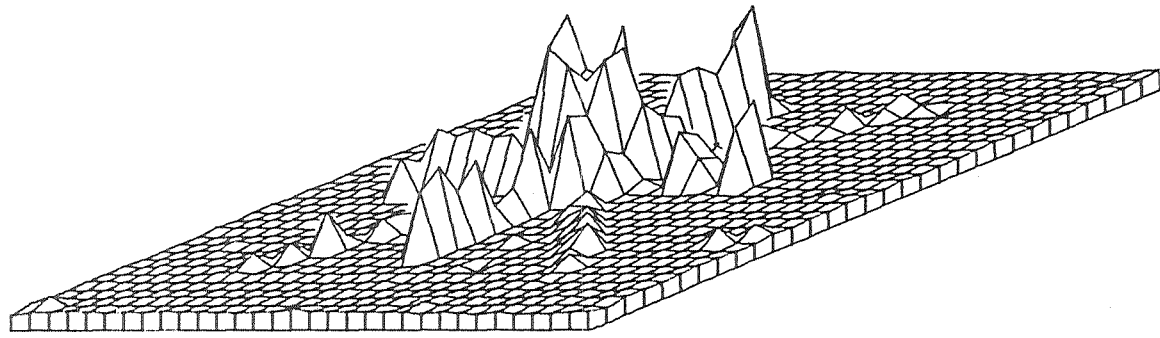
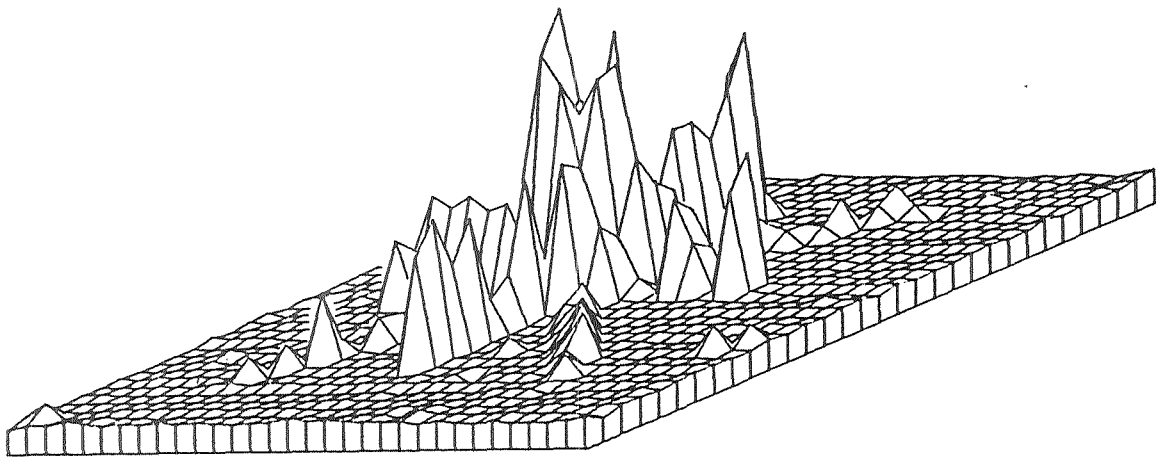


Vista da destra



Vista da sinistra

Es. 2 Rappresentazione della stessa superficie  
con le quattro diverse possibili viste.



Es. 3 Rappresentazione della stessa superficie cambiando il guadagno sull'asse Z.

!	H	H	I	I	I	I	D	D	D	D	D	D	L	L	L	I	I	I	I	I	I	N	N	N	N	E	E	E	E	E	E	E	E	!																											
!	H	H	I	I	I	I	D	D	D	D	D	D	L	L	L	I	I	I	I	I	I	N	N	N	N	E	E	E	E	E	E	E	E	!																											
!	H	H	I	I	D	D	D	D	L	L	I	I	N	N	N	N	E	E	!																																										
!	H	H	I	I	D	D	D	D	L	L	I	I	N	N	N	N	E	E	!																																										
!	H	H	H	H	H	H	H	I	I	D	D	D	D	L	L	L	L	I	I	N	N	N	N	E	E	E	E	E	E	!																															
!	H	H	H	H	H	H	H	I	I	D	D	D	D	L	L	L	L	I	I	N	N	N	N	E	E	E	E	E	E	!																															
!	H	H	I	I	D	D	D	D	L	L	I	I	N	N	N	N	E	E	!																																										
!	H	H	I	I	D	D	D	D	L	L	I	I	N	N	N	N	E	E	!																																										
!	H	H	I	I	I	I	I	I	D	D	D	D	D	D	D	D	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	I	I	I	I	I	I	I	I	I	I	N	N	N	N	E	E	E	E	E	E	E	E	E	E	E	E	E	E	!
!	H	H	I	I	I	I	I	D	D	D	D	D	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	I	I	I	I	I	I	I	I	I	I	N	N	N	N	E	E	E	E	E	E	E	E	E	E	E	E	E	E	!		

Determinazione e tracciatura delle sole righe visibili di una matrice contenente una qualsiasi superficie osservata da un particolare punto di vista ( 45 gradi ).

```

DEFM TIPO
OPTION NOCOM
PAGE 0
@SYSTEM(SYSTEM)CLEVI
IFP %TIPO CONTIP
LIST HIDMASK
NOTE
NOTE *   Plotter, tektronix, svp1000 (P/T/S) (F=Fine) =====>
SET %TIPO
@SYSTEM(SYSTEM)CLEVI
%CONTIP
PAGE
IFT %TIPO EQ F ENDE
IFT %TIPO EQ T TEKT
IFT %TIPO EQ S SVP
IFT %TIPO NE P ERRO
AS TSC TO DEV=NU
AS TEK TO DEV=NU
@SYSTEM(SYSTEM)HIDENLN
P
GOTO FINE
%TEKT
AS TSC TO DEV=NU
AS PLO TO DEV=NU
AS PLP TO LFC=PLO
@SYSTEM(SYSTEM)HIDENLN
T
GOTO FINE
%SVP
AS PLO TO DEV=NU
AS PLP TO LFC=PLO
AS TEK TO DEV=NU
@SYSTEM(SYSTEM)HIDENLN
S
GOTO FINE
%ERRO
NOTE
NOTE *-----*
NOTE !   EPROPE NEL PARAMETRO. !
NOTE !   VALORI POSSIBILI : !
NOTE !           T = Output su tektronix !
NOTE !           P = " " plotter !
NOTE !           S = " " svp1000 !
NOTE !           ! !
NOTE !           F = Fine !
NOTE !           ! !
NOTE !   Premi RETURN per continuare !
NOTE !           ! !
NOTE *-----*
NOTE
SET %TIPO
%FINE
SELECT @SYSTEM(SYSTEM)HIDLINE
ENDM

```

Nome progr. : HIDENLN - Data : 15/7/1936 E. Fantini

Determinazione e tracciatura delle sole righe visibili di una matrice contenente una qualsiasi superficie osservata da un particolare punto di vista.

Descrizione dei Logical Files Codes usati :

LFC	1	assign	UT (Terminale utente)
"	99	"	SYC (Macro chiamante)
"	TEK	"	tektronix 4014
"	PLO	"	plotter Calcomp-31
"	PLP	"	" "
"	TSC	"	SVP1000
"	TSR	"	" "

Librerie usate :

@SYSTEM(LIBRERIE)GEPIM.LIB (.DIR)  
@SYSTEM(LIBRERIE)SVP.LIB (.DIR)

```
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 LFC
INTEGER*1 IUP,IETX,IV(2)
INTEGER*2 IVC(2,255),IVMIN,IVMAX,IM,IB
INTEGER*2 YR1,XR1,YR2,XR2,YR3,XR3,YR4,XR4
INTEGER*2 YR5,YR6,YR7,YR8
INTEGER*2 MAT(128,128),MAT1(128,128),MAT2(128,128)
INTEGER*2 ICH(5),ICR(8),IFILE(3),IB(,12),XI,XF,YI,YF
REAL M,M1,XO,YO,MAXP,Y1,Y2,Y3,Y4,X1,X2,IE,IFF,IC,IA,IG4E
REAL IH,IS,IU,COMO,TA,IMAX
CHARACTER TTY*5,IPT*1,RIS*1
COMMON /A/ MAT,MAT1,MAT2
EXTENDED BASE /A/256
IUP=72 ; IETX=3
IPEN=1 ; ISPIA=0
LFCTTY=LU=1
CALL OPENTTY(LFCTTY,TTY)
NPAG=12
DO UNTIL(NPAG.LE.0)
  CALL X:GDSPCE(NPAG,NGET,,K)
  NPAG=NPAG-NGET
ENDDO
DO K=1,128 ; DO J=1,128
  MAT(K,J)=MAT1(K,J)=MAT2(K,J)=0
ENDDO
ENDDO
READ(99,2) IPT
FORMAT(A1)
IF(IPT.NE.'P'.AND.IPT.NE.'T'.AND.IPT.NE.'S') THEN
  TYPE *, ' Errore nel parametro di individuazione della periferica
- di output :_',IPT
STOP
ENDIF
IMAX=500.
IFC=X'0054454R'
```

```

IF(IPT.EQ.'S') IMAX=256.
IF(IPT.EQ.'P') THEN
    IMAX=1000.
    LFC=X'00504C4F'
ENDIF
1   K=IMAX*2
    WRITE(LU,199) K
199  FORMAT(/,' Numero di punti totali asse -X- [',I4,'] :_')
    READ(LU,3) K
3   FORMAT(I5)
    IF(K.EQ.0) K=IMAX*2
    MAXP=K/2
    IF(MAXP.LT.1.OR.MAXP.GT.IMAX) THEN
        K=IMAX*2
        WRITE(LU,669) K
669  FORMAT(/,' Dimensione troppo grande. Max dimensione = ',I6)
        GOTO 1
    ENDIF
    IE=IFF=0. ; IOP=IS=1 ; IDCB=10 ; IC=IA=0.
10  CALL APRE(IDCB,IFILE,ICH,ICR,IOP,IFO,IB,LU,NREC,LUN,IS)
    IF(IFO.EQ.'I ') GOTO 20
    TYPE *,' ### I dati non sono di tipo -INTERI- ###'
    CALL CHIUD(IDCB,0)
    GOTO 10
20  WRITE(LU,21) NREC
21  FORMAT(/,' ### Riga inizio, riga fine, passo [',I4,'] :_')
    READ(LU,22) IRI,IRF,IRP
22  FORMAT(3I5)
    IF(IRI.EQ.0) IRI=1
    IF(IRF.EQ.0) IRF=NREC
    IF(IRP.EQ.0) IRP=1
    IF(IRI.LT.1.OR.IRF.GT.NREC.OR.iri.GT.IRF.OR.IRP.LT.1) GOTO 20
30  WRITE(LU,31) LUN
31  FORMAT(' ### Colonna inizio, colonna fine, passo [',I4,']:_')
    READ(LU,22) ICI,ICF,ICP
    IF(ICI.EQ.0) ICI=1
    IF(ICF.EQ.0) ICF=LUN
    IF(ICP.EQ.0) ICP=1
    IF(ICI.LT.1.OR.ICF.GT.LUN.OR.ICI.GT.ICF.OR.ICP.LT.1) GOTO 30
    TYPE *,' ### Guadagno in -Z- [100] :_ '
    READ(LU,3) K
    IGAIN=FLOAT(K)
    IF(K.EQ.0) IGAIN=100.

C
C   Lettura sottomatrice selezionata e
C   Vista anteriore
C
1111 IR=0
    DO I=IRI,IRF,IRP
        CALL LESC(IDCB,0,IFO,IB,I,LUN)
        IR=IR+1 ; JC=0
        DO J=ICI,ICF,ICP
            JC=JC+1
            MAT(IR,JC)=MAT1(IR,JC)=I3(J)
            IF(I3(J).GT.IE) IE=I3(J)
            IF(I3(J).LT.IFF) IFF=I3(J)
        ENDDO
    ENDDO
    LC=JC ; LR=IR
35  WRITE(LU,40)
40  FORMAT(/,' ### PUNTO DI VISTA ###',//,
-      '      1 = Anteriore',//,
-      '      2 = Posteriore',//)

```

3 = Destra',/'  
4 = Sinistra [1] :\_'

```
READ(LU,3) IVIS  
IF(IVIS.EQ.0) IVIS=1  
IF(IVIS.LT.1.OR.IVIS.GT.4) GOTO 35  
IF(IPT.EQ.'P') THEN
```

45 TYPE \*,' Che penna uso (1-3) [1] :\_'

```
READ(LU,22) IPEN  
IF(IPEN.EQ.0) IPEN=1  
IF(IPEN.LT.1.OR.IPEN.GT.3) GOTO 45
```

```
ENDIF  
GOTO (8888,2222,3333,4444) IVIS
```

C  
C Calcoli determinazione linee nascoste  
C e tratteggio orizzontale  
C

50

```
IVMAX=IE+.5 ; IVMIN=IFF+.5 ; IE=IE-IFF  
IH=MAXP/FLOAT(LC) ; IS=IH*.4  
XI=YI=YF=0 ; XF=IH*(LC-1)+.5  
IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)  
IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)  
IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,  
-ISPIA,IFF,IE,IC,IGAIN)
```

```
DO J=1,LC-1  
VI=(FLOAT(MAT(1,J))-IFF)/IE*IGAIN+.5  
YF=(FLOAT(MAT(1,J+1))-IFF)/IE*IGAIN+.5  
Z1=FLOAT(J-1)*IH ; Z2=FLOAT(J)*IH  
XI=Z1+.5 ; XF=Z2+.5  
IF(J.EQ.1) THEN
```

```
YR1=YI  
XR1=XI  
ENDIF  
IF(J.EQ.(LC-1)) THEN  
YR2=YF  
XR2=XF  
ENDIF
```

```
IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)  
IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)  
IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,  
-ISPIA,IFF,IE,IC,IGAIN)
```

```
MAT2(1,J)=YI  
YF=YI ; XF=XI ; YI=0  
IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)  
IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)  
IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,  
-ISPIA,IFF,IE,IC,IGAIN)  
ENDDO
```

```
YI=0  
Z1=(FLOAT(MAT(1,LC))-IFF)/IE*IGAIN  
YF=MAT2(1,LC)=Z1+.5  
Z2=FLOAT(LC-1)*IH  
XI=XF=Z2+.5  
IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)  
IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)  
IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,  
-ISPIA,IFF,IE,IC,IGAIN)
```

```
DO K=1,LC-1  
MAT2(1,K)=MAT2(1,(K+1))  
ENDDO
```

```
IC=IC+IS ; IA=IA+IH  
DO 90 I=2,LR  
DO 80 J=1,LC-1  
Y1=(FLOAT(MAT(I,J))-IFF)/IE*IGAIN+IC
```

```

Y2=(FLOAT(MAT(I,J+1))-IFF)/IE*IGAIN+IC
Y3=FLOAT(MAT2(I-1,J))
IU=Y3
Y4=FLOAT(MAT2(I-1,J+1))
X1=FLOAT(J-1)*IH+IA
X2=FLOAT(J)*IH+IA
COMO=MAXP+IC
IF(Y1.GT.COMO) Y1=COMO
IF(Y2.GT.COMO) Y2=COMO
IF(Y1.GE.Y3) THEN
    IU=Y1
    GOTO 130
ENDIF
GOTO 150
30 XI=X1+.5 ; YI=Y1+.5 ; XF=X2+.5 ; YF=Y2+.5
IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,
- ISPIA,IFF,IE,IC,IGAIN)
70 MAT2(I,J)=IU+.5
IF(J.EQ.1) THEN
    IF(Y1.GT.YR3) YR3=Y1
ENDIF
IF(I.EQ.LR.AND.J.EQ.(LC-1)) THEN
    IF(YF.GT.YR4) YR4=YF
ENDIF
80 CONTINUE
MAT2(I,LC)=Y2+.5
Z1=FLOAT(I-1)*IS ; Z2=Y2
XI=X2+.5 ; YI=Z1+.5 ; XF=X2+.5 ; YF=Z2+.5
IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,
- ISPIA,IFF,IE,IC,IGAIN)
DO K=1,LC-1
    MAT2(I,K)=MAT2(I,(K+1))
ENDDO
90 IA=IA+IH ; IC=IC+IS
CONTINUE
XI=(LC-1)*IH+.5 ; YI=0
XF=X2+.5 ; YF=(LR-1)*IS+.5
IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,
- ISPIA,IFF,IE,IC,IGAIN)
C
C INIZIO TRATTEGGIO VERTICALE
C
99 IC=0. ; IA=0.
DO 120 J=1,LC
DO 110 I=1,LR-1
IF(MAT(I,J).NE.-32000) GOTO 100
IF(MAT(I+1,J).EQ.-32000) GOTO 110
100 Y1=(FLOAT(MAT(I,J))-IFF)/IE*IGAIN+FLOAT(I-1)*IS
Y2=(FLOAT(MAT(I+1,J))-IFF)/IE*IGAIN+FLOAT(I)*IS
IC=FLOAT(I-1)*IS
X1=FLOAT(I-1)*IH+IA ; X2=FLOAT(I)*IH+IA
COMO=MAXP+FLOAT(I-1)*IS
IF(Y1.GT.COMO) Y1=COMO
COMO=MAXP+FLOAT(I)*IS
IF(Y2.GT.COMO) Y2=COMO
IF(MAT(I,J).EQ.-32000) GOTO 180
IF(MAT(I+1,J).EQ.-32000) GOTO 190

```



```

        XI=X1+.5 ; YI=Y1+.5 ; XF=X2+.5 ; YF=Y2+.5
        IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
        IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
        IF(IPT.EQ.'S') CALL PLOSVPL(U,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,
- ISPIA,IFF,IE,IC,IGAIN)
110  CONTINUE
        IA=IA+IH
120  CONTINUE
179  TYPE *, ' ### Vuoi tracciare il cubo circoscritto (S/N) :_'
        READ (LU,2) RIS
        IF(RIS.NE.'S'.AND.RIS.NE.'N') GOTO 179
        IF(RIS.EQ.'N') GOTO 1337
        XR4=XR2*2 ; XR3=XR2
        YR5=YR6=MAXP
        YR7=YR8=(MAXP+FLOAT(LR)*IS)+.5
        XI=XF=XR1 ; YI=YR1 ; YF=YR5
        K=1 ; GOTO 1330
1320  YI=YF ; XF=XR2 ; K=2 ; GOTO 1330
1321  XI=XF ; YI=YR2 ; K=3 ; GOTO 1330
1322  YI=YF ; XF=XR4 ; YF=YR7 ; K=4 ; GOTO 1330
1323  XI=XF ; YI=YR4 ; K=5 ; GOTO 1330
1324  XF=XR2 ; YI=YF ; K=6 ; GOTO 1330
1325  XI=XF ; YI=YR6 ; K=7 ; GOTO 1330
1326  XF=XR1 ; YF=YR5 ; YI=YR8 ; XI=XR2 ; K=8
1330  IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
        IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
        IF(IPT.EQ.'S') CALL PLOSVPL(U,IM,BB,IVC,IVMAX,IVMIN,XI,XF,YI,YF,
- ISPIA,IFF,IE,IC,IGAIN)
        GOTO(1320,1321,1322,1323,1324,1325,1326,1337) K
1357  IF(IPT.EQ.'P') THEN
            ENCODE(2,125,IV(1)) IUP,IETX
125  FORMAT(2A1)
            CALL RWB(LFC,2,IV,2,IST)
        ELSE
            IV(1)=31
            CALL RWB(LFC,2,IV,1,IST)
        ENDIF
        STOP
C
C
130  IF(Y2.GE.Y4) GOTO 60
        N99=1
140  IFUN=INTER(M,M1,X0,Y0,X1,X2,Y1,Y2,Y3,Y4,TA)
        IF(IFUN.EQ.1) THEN
            COMO=MIXP+FLOAT(I-1)*IS
            IF(Y0.GT.COMO) Y0=COMO
            XI=X1+.5 ; YI=Y1+.5 ; XF=X0+.5 ; YF=Y0+.5
            IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
            IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
            IF(IPT.EQ.'S') CALL PLOSVPL(U,IM,BB,IVC,IVMAX,IVMIN,
- XI,XF,YI,YF,ISPIA,IFF,IE,IC,IGAIN)
        ENDIF
1449  GOTO (70,110) N99
150  IF(Y2.LE.Y4) GOTO 170
        IFUN=INTER(M,M1,X0,Y0,X1,X2,Y1,Y2,Y3,Y4,TA)
        IF(MAT(I-1,J+1).NE.0) GOTO 160
        IF(IFUN.EQ.1) THEN
            XI=X0+.5 ; YI=Y0+.5 ; XF=X2+.5 ; YF=Y2+.5
            IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
            IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
            IF(IPT.EQ.'S') CALL PLOSVPL(U,IM,BB,IVC,IVMAX,IVMIN,
- XI,XF,YI,YF,ISPIA,IFF,IE,IC,IGAIN)
        ENDIF

```

```

160 CONTINUE
170 MAT(I,J)=-32000
    GOTO 70
180 Y1=(FLOAT(MAT1(I,J))-IFF)/IE*IGAIN+FLOAT(I-1)*IS
    IF(I.EQ.1.OR.J.EQ.LC) GOTO 110
    Y3=FLOAT(MAT2(I-1,J))
    Y4=FLOAT(MAT2(I-1,J+1))
    IF(MAT2(I,J).GT.MAT2(I-1,J+1)) Y4=FLOAT(MAT2(I,J))
    IFUN=INTER(M,M1,XO,YO,X1,X2,Y1,Y2,Y3,Y4,TA)
    IF(IFUN.EQ.1) THEN
        XI=XO+.5 ; YI=YO+.5 ; XF=X2+.5 ; YF=Y2+.5
        IF(IPT.EQ.'P') CALL PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
        IF(IPT.EQ.'T') CALL PLOTEK(LFC,XI,YI,XF,YF,ISPIA)
        IF(IPT.EQ.'S') CALL PLOSVP(LU,IM,BB,IVC,IVMAX,IVMIN,
- XI,XF,YI,YF,ISPIA,IFF,IE,IC,IGAIN)
    ENDIF
    GOTO 110
190 Y2=(FLOAT(MAT1(I+1,J))-IFF)/IE*IGAIN+FLOAT(I)*IS
    IF(Y1.GE.Y2) GOTO 110
    IF(I.EQ.1) GOTO 11J
    IF(MAT2(I,J).GT.Y2) GOTO 110
    Y4=FLOAT(MAT2(I,J)) ; Y3=FLOAT(MAT2(I-1,J))
    N99=2
    GOTO 140

```

C  
C Riconfigurazione della matrice dati a seconda del punto di vista  
C  
C

C Vista posteriore  
C

```

2222 IR=0
    DO I=LR,1,-1
        IR=IR+1 ; JC=0
        DO J=LC,1,-1
            JC=JC+1
            MAT(IR,JC)=MAT1(I,J)
        ENDDO
    ENDDO
    GOTO 9999

```

C  
C Vista destra  
C

```

3333 DO I=1,LR
        DO J=1,LC
            IR=LC-J+1
            MAT(IR,I)=MAT1(I,J)
        ENDDO
    ENDDO
    IR=LC ; LC=LR ; LR=IR
    GOTO 9999

```

C  
C Vista sinistra  
C

```

4444 DO I=1,LR
        DO J=1,LC
            JC=LR-I+1
            MAT(J,JC)=MAT1(I,J)
        ENDDO
    ENDDO
    IR=LC ; LC=LR ; LR=IR
9999 DO I=1,LR
        DO J=1,LC
            MAT1(I,J)=MAT(I,J)

```

```

      ENDDO
      ENDDO
8888  CALL CHIUD(IDC8,U)
      GOTO 50
      END

```

```

C
C
C
C
C

```

```

##### SUBROUTINES E FUNCTIONS #####

```

```

FUNCTION CALCOLO PUNTO INTERSEZIONE SEGMENTI SEMINASCOSTI

```

```

INTEGER*2 FUNCTION INTER(M,M1,X0,Y0,X1,X2,Y1,Y2,Y3,Y4,TA)
INTEGER*2 XI,XF,YI,YF
REAL M,M1,X0,Y0,Y1,Y2,Y3,Y4,TA
INTER=0
M=(Y2-Y1)/(X2-X1)
M1=(Y4-Y3)/(X2-X1)
X0=(Y1-M*X1-Y3+M1*X1)/(M1-M)
Y0=((Y4-Y3)*(Y1-M*X1)-(Y2-Y1)*(Y3-M1*X1))/(Y4-Y3-Y2+Y1)
TA=((X2-X1)*(Y3-Y1))/(-(X2-X1)*(Y4-Y3)+(Y2-Y1)*(X2-X1))
IF(TA.GE.0.AND.TA.LE.1) INTER=1
RETURN
END

```

```

C
C
C

```

```

SUBROUTINE PER TRACCIATURA SU PLOTTER

```

```

SUBROUTINE PLOTTA(LFC,XI,YI,XF,YF,IPEN,ISPIA)
IMPLICIT INTEGER*2 (I-N)
INTEGER*1 ISOH,IP,ISP,ISET(25),IETX,IV(50),IX1,IBL,
-ILL,IS,IK,IUP,IDW
INTEGER*2 XI,YI,XF,YF
INTEGER*4 LFC,IST,IT1,IT2,LFC1
CHARACTER ICAR*25
EQUIVALENCE (ICAR,ISET(1))
DATA ISOH/01/,IP/8/,ISP/70/,IETX/03/,IX1/17/,IBL/32/,
-ILL/76/,IK/75/,IS/47/,IUP/72/,IDW/73/
LFC1=LFC+1
ICAR='3 17 0 250 0 13 0 J 67 63'
IF(ISPIA.NE.0) GOTO 20
CALL RWB(LFC,0,IV,1,IST)
CALL RWB(LFC1,0,IV,1,IST)
10  ENCODE(23,10,IV(1)) ISOH,IP,(ISET(1),I=1,25),IETX
    FORMAT(23A1)
    CALL RWB(LFC,2,IV,28,IST)
20  ENCODE(3,25,IV(1)) ISOH,IP,IX1
25  FORMAT(3A1)
    CALL RWB(LFC,2,IV,3,IST)
    CALL WAIT(50,1,KKK)
    CALL RWB(LFC1,1,IV,2,IST)
    IF(IV(1).EQ.68) GOTO 20
    IF(ISPIA.EQ.IPEN) GOTO 35
    ISPIA=IPEN
    ENCODE(2,30,IV(1)) ISP,IPEN
30  FORMAT(A1,I1)
    CALL RWB(LFC,2,IV,2,IST)
35  ENCODE(11,40,IV(1)) ILL,IT1,IBL,IT2,IBL
40  FORMAT(2(A1,I4),A1)
    CALL RWB(LFC,2,IV,11,IST)
    ENCODE(11,40,IV(1)) IUP,XI,IS,YI,IK
    CALL RWB(LFC,2,IV,11,IST)
    ENCODE(11,40,IV(1)) IDW,XF,IS,YF,IK
    CALL RWB(LFC,2,IV,11,IST)
RETURN

```

END

C  
C  
C

SUBROUTINE PER TRACCIATURA SU TEKTRONIX

SUBROUTINE PLOTEK(LFC,XI,YI,XF,YF,ISPIA)

IMPLICIT INTEGER\*2 (I-N)

DIMENSION IV(5),IVX(2),IVY(2)

INTEGER\*4 LFC,IST,IR(5)

INTEGER\*2 XI,XF,YI,YF

INTEGER\*1 IVC(2)

DATA IVC(1)/27/,IVC(2)/12/

IF(ISPIA.NE.0) GOTO 10

CALL RWB(LFC,0,IVC,1,IST)

CALL RWB(LFC,2,IVC,2,IST)

CALL WAIT(3000,1,KKK)

10 IV(1)=29

IVX(1)=XI ; IVX(2)=XF ; IVY(1)=YI ; IVY(2)=YF

CALL GMP(2,IVX,IVY,IV(2))

CALL RWB(LFC,2,IV,10,IST)

ISPIA=9999

RETURN

END

C  
C  
C

C\*\*\*\*

SUBROUTINE GMP(IN1,IN2,IN3,IN4)

IMPLICIT INTEGER\*2 (I-N)

DIMENSION IN2(1),IN3(1),IN4(1)

K=0

DO J=1,IN1

K=K+1

IF(IN3(J).LT.0) IN3(J)=0

IF(IN3(J).GT.1023) IN3(J)=1023

IN4(K)=(IN3(J)/32+32)\*256+IAND(IN3(J),0\*37\*8)+96

K=K+1

IF(IN2(J).LT.0) IN2(J)=0

IF(IN2(J).GT.1023) IN2(J)=1023

IN4(K)=(IN2(J)/32+32)\*256+IAND(IN2(J),0\*37\*8)+64

ENDDO

RETURN

END

C  
C  
C

SUBROUTINE PER TRACCIATURA SU SVP

SUBROUTINE PLOSVP(LU,IM,IB,IVC,IVMAX,IVMIN,X1,X2,Y1,Y2,ISPIA,  
-IFF,IE,IC,IGAIN)

IMPLICIT INTEGER\*2 (I-N)

REAL IFF,IE,IC,IGAIN,Y9

INTEGER\*2 XI,XF,YI,YF,Y,X,IVMAX,IVMIN,IV(2),IVC(2,256)

INTEGER\*2 X1,X2,Y1,Y2,Y8

DIMENSION ICH(3),ICR(8),IFILE(3),IA(256)

CHARACTER IRIS\*1,ICO\*1(3)

IF(ISPIA.NE.0) GOTO 13

ISPIA=9999

1 TYPE \*, " # Reset dell' SVP1000 (S/N) ? :\_"

READ(LU,2) IRIS

2 FORMAT(A1)

IF(IRIS.EQ."N") GOTO 3

IF(IRIS.NE."S") GOTO 1

CALL TINIZ

CALL TRSFW

CALL TRHRW

```

CALL WAIT(1000,1,KKK)
CALL ASINT
3 CALL MEBA(IM,IB,99)
4 TYPE *, " ## Caricamento tavole di look-up (S/N) ? :_"
READ(LU,2) IRIS
IF(IRIS.EQ."N") GOTO 81
IF(IRIS.NE."S") GOTO 4
ICO(1)="R" ; ICO(2)="G" ; ICO(3)="B"
DO 1000 K=1,3
WRITE(LU,5) ICO(K)
5 FORMAT(/" ### Caricamento tavola -",A1,"- ####",/)
IDCB=77
6 CALL APRE(IDCB,IFILE,ICH,ICR,1,IF,IA,LU,NR,LR,LU)
IF(NR.EQ.1.AND.LR.EQ.256.AND.IF.EQ."I") GOTO 7
TYPE *, " ### ERRORE ###. Questo file non contiene look-up."
GOTO 6
7 CALL LESC(IDCB,0,IF,IA,1,LR)
CALL CHIUD(IDCB,0)
CALL PACK(IA,-LR,IA)
IV(1)=0"4402"
IV(2)=K-1
CALL TOUT(IV,2,IERR)
CALL TOUT(IA,128,IERR)
1000 CONTINUE
81 TYPE *, " ### Valore per il fondo (0-255) [0] ? :_"
READ(LU,2) K
82 FORMAT(I3)
IF(K.LT.0.OR.K.GT.255) GOTO 81
CALL MEMO(IM,K)
71 WRITE(LU,72)
72 FORMAT(/" ## COLORAZIONE DEL GRAFICO #",/,
- " 1 = monocromatico",/,
- " 2 = corrisp. valori tavole look-up",/,
- " 3 = a fasce di valori (max 256)",/,
- "[1] =====> ")
READ(LU,32) ITD
IF(ITD.EQ.0) ITD=1
IF(ITD.LT.1.OR.ITD.GT.3) GOTO 71
GOTO (3,73,8) ITD
73 DO K=1,256 ; IVC(1,K)=K-1 ; IVC(2,K)=K-1 ; ENDDO
GOTO 13
8 TYPE *, " ## Colore di default (0-255) [0] ? :_"
READ(LU,32) L
IF(L.LT.0.OR.L.GT.255) GOTO 8
DO K=1,256 ; IVC(1,K)=IVMAX ; IVC(2,K)=L ; ENDDO
IF(ITD.NE.3) GOTO 13
9 WRITE(LU,10) IVMIN,IVMAX
10 FORMAT(/" ## Valori MIN., MAX. =",2(2X,I4),/)
K=1
11 WRITE(LU,12) K
12 FORMAT(/" Fascia num. ",I3," - Valore, luminanza ? :_" )
READ(LU,*) IVC(1,K),IVC(2,K)
IF(IVC(1,K).LT.0.OR.IVC(2,K).LT.0) THEN
IVC(1,K)=IVMAX ; IVC(2,K)=L
GOTO 13
ENDIF
IF(IVC(1,K).LT.IVMIN.OR.IVC(1,K).GT.IVMAX.OR.
-IVC(2,K).LT.0.OR.IVC(2,K).GT.255) GOTO 11
K=K+1
IF(K.LT.257) GOTO 11
13 XI=XI+1 ; XF=XF+1 ; YI=YI+1 ; YF=YF+1
IDX=XF-XI ; IDY=YF-YI ; LC=0
X=XI-1 ; Y=YI-1 ; K=1

```

```

14   Y9=IFF-(IE+IC/IGAIN)-(IE*.5/IGAIN)+(IE*FLOAT(511-Y)/IGAIN)
     Y8=Y9
141  IF(IVC(1,K).GE.Y8) THEN
     ICOL=IVC(2,K)
     GOTO 15
     ENDIF
     K=K+1
     IF(K.LT.257) GOTO 141
     ICOL=L
15   K99=1
     GOTO 2000
16   MX=IABS(IDX) ; MY=IABS(IDY)
     IF(MX.GE.MY) GOTO 222
     KC=MX ; MX=MY ; MY=KC
222  RM=MX/2
     IDX=ISIGN(1,IDX)
     IDY=ISIGN(1,IDY)
52   IF(MX.EQ.LC) GOTO 111
     RM=RM+MY
     ID1=IDX ; ID2=IDY
     IF(RM.GT.MX) GOTO 50
     IF(MX.NE.IABS(XF-XI)) GOTO 61
     ID2=0
     GOTO 60
61   ID1=0
     GOTO 60
50   RM=RM-MX
50   X=X+ID1 ; Y=Y+ID2 ; K=1
56   Y9=IFF-(IE+IC/IGAIN)-(IE*.5/IGAIN)+(IE*FLOAT(511-Y)/IGAIN)
     Y8=Y9
     IF(IVC(1,K).GE.Y8) THEN
     ICOL=IVC(2,K)
     GOTO 67
     ENDIF
     K=K+1
     IF(K.LT.257) GOTO 56
     ICOL=L
67   K99=2
     GOTO 2000
68   LC=LC+1
     GOTO 52
2000 ML=ICOL*256+IM-1
     IA(1)=0*40)4
     IA(2)=Y
     IA(3)=X
     IA(4)=ML
     CALL TOUT(IA,4,IERR)
     GOTO(16,58) K99
111  RETURN
     END

```