

INTRODUZIONE AL PROBLEMA DELLE LINEE NASCOSTE

È noto, e chi segue questa rubrica di Computer Grafica lo ha letto e riletto più volte, che uno dei problemi più complessi nei processi di rappresentazione bidimensionale di figure tridimensionali è costituito dalla individuazione delle "linee nascoste".

Nei numeri scorsi abbiamo presentato alcune tecniche di visualizzazione (prospettiva, assonometria e proiezione ortogonale) con le quali abbiamo anche realizzato dei programmi dimostrativi con output su plotter o su monitor grafico.

Sviluppando queste tecniche, però, abbiamo previsto la visualizzazione di tutte le linee costituenti la figura, senza la individuazione, e conseguente non visualizzazione quindi, di quelle nascoste dagli altri elementi della figura interposti rispetto al punto di vista dell'osservatore.

Il problema è estremamente complesso

visione ed eseguire il test di visibilità su ogni elemento della figura e questo test può dare tre risposte: elemento visibile, elemento non visibile, elemento parzialmente visibile.

In quest'ultimo caso l'analisi si appesantisce ancora di più, in quanto occorre stabilire che porzione dell'elemento è visibile e che porzione non lo è.

I metodi per l'individuazione delle "Hidden Lines"

I metodi generalizzati di individuazione delle linee nascoste lavorano nello spazio oggetto, ovvero il test di visibilità viene eseguito direttamente nello spazio reale ove sono posti osservatore, schermo di visualizzazione e figura da visualizzare.

Il sistema più semplificato è quello di lavorare nello spazio soggetto ovvero sul

disegnare e quale no. Questo procedimento semplificato rende il problema più accessibile dal punto di vista del programmatore e soprattutto limita i tempi di elaborazione.

Prima di descriverne le caratteristiche e di mostrare gli esempi realizzati, sarà bene ricapitolare nel suo complesso il problema della prospettiva/hidden lines suddividendolo nei suoi passi logici.

Le fasi della procedura

La procedura per la visualizzazione di oggetti tridimensionali, con la eliminazione delle linee nascoste, si compone di alcuni passi logici:

1 - Immissione dei dati dell'oggetto da visualizzare. Questi dati sono riferiti al sistema di riferimento assoluto e valgono indipendentemente dal sistema di rappresentazione prescelto.

1-1 Dati immessi direttamente da programma. Si può utilizzare tale metodo quando i dati sono pochi, ad esempio quando si vuol visualizzare una funzione $Y = Y(X, Z)$, in genere inseribile in una unica istruzione.

1-2 Dati immessi via input (da tastiera o da digitizer) quando i dati non sono molti e vanno utilizzati una sola volta.

1-3 Dati immessi via file. Nella generalità dei casi, per dati numerosi e da utilizzare

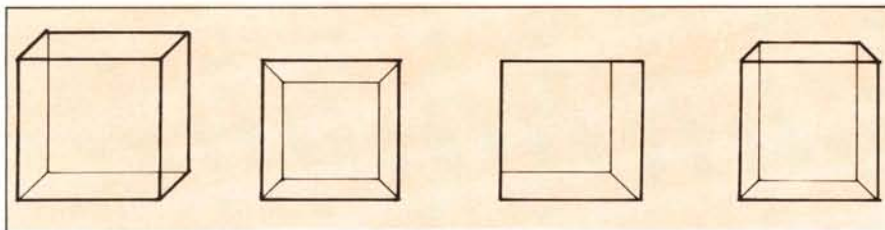


Figura 1 - Le linee nascoste nella vista prospettica di un cubo. A seconda della posizione relativa tra osservatore e cubo, le linee nascoste possono essere da tre a otto.

non solo a livello di programmazione ma soprattutto a livello di analisi. È difficilmente generalizzabile a causa della infinita varietà di casi possibili.

Ne studieremo per ora alcune semplificazioni che permetteranno di ottenere, anche se per casi particolari, risultati interessanti.

Nella rappresentazione prospettica di una figura si pone il "problema della visibilità", ovvero il problema di determinare quali facce e/o quali parti di facce e quali spigoli e/o quali parti di spigoli siano realmente visibili all'osservatore e quali no.

Ad esempio esaminiamo la figura 1. Del cubo, a seconda della posizione relativa tra osservatore, schermo e figura, sono visibili 9, 4, 7, ecc. dei 12 spigoli che individuano la figura nello spazio e quindi sul foglio. Adirittura per figure particolari, ad esempio per il solido complesso della figura 2, alcune linee sono nascoste solo parzialmente.

Il problema da un punto di vista analitico risulta particolarmente difficile, anche per figure molto semplici, in quanto occorre simulare il complesso meccanismo della

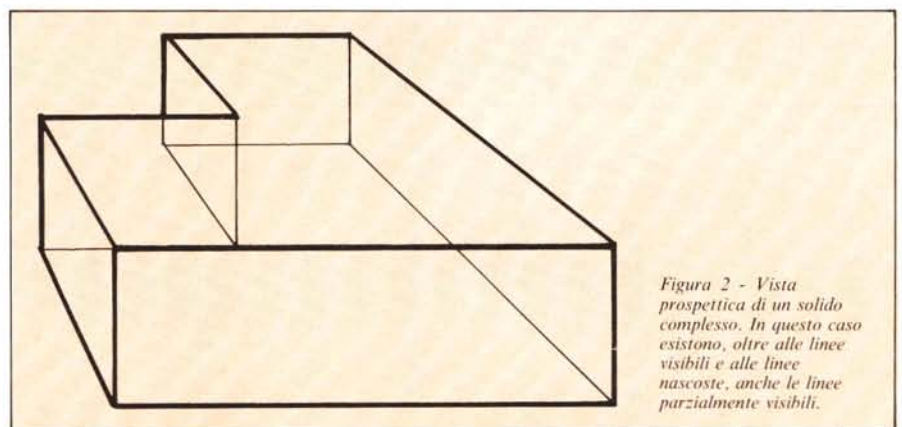


Figura 2 - Vista prospettica di un solido complesso. In questo caso esistono, oltre alle linee visibili e alle linee nascoste, anche le linee parzialmente visibili.

piano bidimensionale dove si concretizza la rappresentazione, direttamente sugli elementi geometrici disegnabili (che sono solamente punti e linee).

In altre parole: immaginiamo di aver già risolto il problema di calcolare tutti gli elementi della figura in coordinate schermo, al momento di disegnarli eseguiamo il test di visibilità per decidere quale elemento

più volte, conviene realizzare un programma apposito per la creazione e la gestione (modifiche, correzioni, inserimenti, cancellazioni, ecc.) dei dati grafici. Il programma di output leggerà il file.

2 - Immissione dati relativi a osservatore e schermo. Immessi i dati individuanti l'oggetto nel riferimento assoluto, vanno immessi, riferiti allo stesso riferimento i


```

100 XM = - 9999: XN = 9999: YM = - 9999: YN = 9999
110 HGR : HCOLOR= 3: GOSUB 120
120 DIM X%(300), Y%(300), F%(300)
130 J = 1 + 1: HOME
140 VTab (210): HTab (240): INVERSE : PRINT "SCHERMA ": SC
150 NORMAL : VTab (23): PRINT " PUNTO ": I:
160 HTab (15): INPUT "X,Y,F" : (X%(I), Y%(I), F%(I))
170 REM RICERCA MIN/MAX
180 IF X%(I) < XM THEN XM = X%(I)
190 IF X%(I) > XN THEN XN = X%(I)
200 IF Y%(I) < YM THEN YM = Y%(I)
210 IF Y%(I) > YN THEN YN = Y%(I)
220 DX = XM - XN + .0001: DY = YM - YN + .0001
230 SX = 260 / DX: SY = 150 / DY
240 SC = SX: IF SY < SX THEN SC = SY
250 HGR : GOSUB 320: FOR J = 1 TO I
260 IF J = 1 THEN F%(J) = 0
270 X% = (X%(J) - XM - DX / 2) * SC + 139
280 Y% = (Y%(J) - YM - DY / 2) * SC + 80
290 IF F%(J) = 0 THEN HPLLOT X%, Y%: GOTO 310
300 HPLLOT TO X%, Y%
310 NEXT J: GOTO 130
320 HPLLOT 0,0 TO 279,0 TO 279,159 TO 0,159 TO 0,0 RETURN

```

Figura 3 - Programma SCALING - DEMO. Listato. Le operazioni di scaling e di visualizzazione vengono eseguite per ogni nuova immissione. Nella riga di testo sotto l'immagine viene riportato il fattore di scala calcolato.

dati dell'osservatore (posizione nel riferimento, orientazione e angolazione della visuale) e i dati dello schermo (posizione nello stesso riferimento).

In genere il programma, eseguita la prima visualizzazione, torna a questo passo per il cambio del punto di osservazione dello stesso oggetto.

3 - Elaborazione dei dati della figura sullo schermo. Sulla base dei dati relativi alla visualizzazione, il programma trasforma le coordinate nello spazio assoluto della figura in coordinate sul piano di visualizzazione.

Tale elaborazione può dare risultati non reali, ad esempio se la direzione verso la quale è orientato l'osservatore non incontra il piano schermo o non incontra il soggetto.

4 - Scaling dei dati.

Una volta tradotti i dati dallo spazio assoluto al piano di visualizzazione va eseguito lo scaling, per riportare le dimensioni della figura al formato voluto, comunque

non superiore al formato del supporto output, e per centrarla rispetto allo stesso sistema output.

Questa routine è standard, ma rallenta moltissimo la procedura in quanto comporta la necessità di scorrere ed elaborare necessariamente tutti i dati per cercare i valori: XM, XN, YM, YN massimi e minimi dei valori schermo X,Y: DX,DY inter-

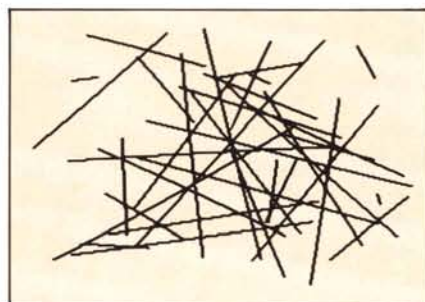


Figura 5 - Programma Linee. Output. La visualizzazione dei segmenti, che avviene punto per punto, è molto rallentata.

```

100 HOME : PRINT "TRACCIAMENTO SEGMENTI ": PRINT
110 PRINT " 1 - PER LINEE": INPUT " 2 - PER PUNTI ": PP
120 IF PP < > 1 AND PP < > 2 THEN 100
130 HGR2 : HCOLOR= 3
140 X1 = RND (1) * 280: Y1 = RND (1) * 192
150 X2 = RND (1) * 280: Y2 = RND (1) * 192
160 ON PP GOSUB 170, 190: GOTO 140
170 REM TRACCIAMENTO SEGMENTI
180 HPLLOT X1, Y1 TO X2, Y2: RETURN
190 REM TRACCIAMENTO PER PUNTI
200 IF X1 = X2 THEN T = 99999: GOTO 220
210 T = (Y2 - Y1) / (X2 - X1): S = 1
220 IF ABS (T) > 1 THEN 240
230 IF X1 > X2 THEN S = - 1
240 FOR X = X1 TO X2 STEP S
250 Y = Y1 + (X - X1) * T + .5
260 HPLLOT X, Y
270 NEXT X: RETURN
280 IF Y1 > Y2 THEN S = - 1
290 FOR Y = Y1 TO Y2 STEP S
300 X = X1 + (Y - Y1) / T + .5
310 HPLLOT X, Y
320 NEXT Y: RETURN

```

Figura 6 - Programma Linee. Listato. Si distinguono i due casi di loop sulla X, per segmenti con inclinazione minore di 45° sull'orizzontale, e sulla Y per gli altri, rispettivamente righe 230, 280.

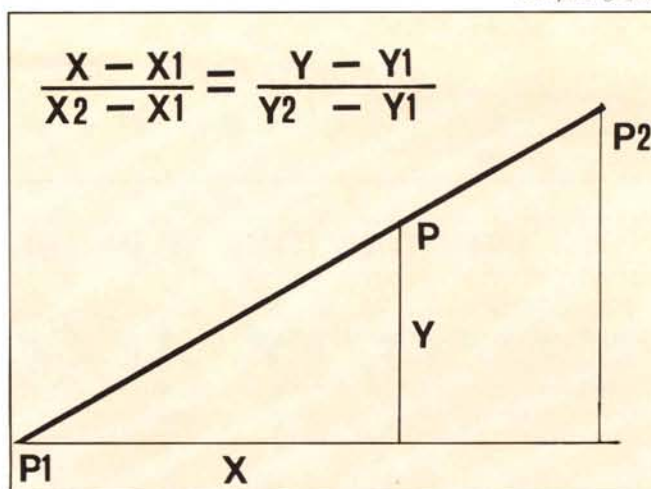


Figura 4 - Schizzo illustrativo. Per individuare il punto P(X,Y) giacente sul segmento tra i due punti P1 e P2 si usa una semplice proporzione.

vallo tra valori massimi e minimi e di trovare, mediante la proporzione con i dati relativi al formato output voluto (ad es. 279 per 191 per il monitor Apple II) il fattore di scala SC (il più piccolo tra SX e SY).

Per quanto riguarda questo aspetto particolare (scaling) presentiamo (listato fig. 3) un nuovo programma Demo, che esegue lo scaling e visualizza dati immessi da tastiera.

Si possono immettere dati bidimensionali, coordinate X,Y e flag 0/1 per inizio/continuazione linea spezzata.

Per ogni nuovo dato immesso viene rieseguita la routine di scaling e riformattato il disegno sul monitor Apple II.

Poiché il risultato della subroutine di scaling sono solo alcuni dati (SC, XM, YN, DX, DY), se li precalcoliamo (ovviamente per un dato soggetto e per un dato punto di vista) li potremo inserire direttamente nel programma di visualizzazione come costanti, alleggerendolo notevolmente.

```

100 DIM YM(279), YN(279)
110 REM INIZIALIZZAZIONE YMAX, YMIN
120 FOR I = 0 TO 279: YM(I) = 110: YN(I) = 80: NEXT I
130 HGR2 : HCOLOR= 3
140 HPLLOT 0,0 TO 279,0 TO 279,191
150 HPLLOT TO 0,191 TO 0,0
160 FOR X = 0 TO 279: HPLLOT X, YM(X): NEXT X
170 FOR X = 0 TO 279: HPLLOT X, YN(X): NEXT X
180 X1 = RND (1) * 280: Y1 = RND (1) * 80
190 X2 = RND (1) * 280: Y2 = RND (1) * 82 + 110
200 IF X1 = X2 THEN T = 99999: GOTO 220
210 T = (Y2 - Y1) / (X2 - X1): S = 1
220 IF ABS (T) > 1 THEN 240
230 IF X1 > X2 THEN S = - 1
240 FOR X = X1 TO X2 STEP S
250 Y = Y1 + (X - X1) * T + .5
260 GOSUB 310: NEXT X: GOTO 180
270 IF Y1 > Y2 THEN S = - 1
280 FOR Y = Y1 TO Y2 STEP S
290 X = X1 + (Y - Y1) / T + .5
300 GOSUB 310: NEXT Y: GOTO 180
310 REM TEST DENTRO FASCIA
320 IF Y > YM(X) AND Y < YN(X) THEN RETURN
330 HPLLOT X, Y: RETURN

```

Figura 7 - Programma Fascia. Listato. Inizializzati i vettori contenenti le Ymax e le Ymin, e visualizzata la fascia tra essi compresa si esegue per ogni punto di ogni segmento da tracciare il test di dentro/fuori fascia.

Linee nascoste e linee visibili

In un processo di visualizzazione dobbiamo scomporre in segmenti tutte le figure, di tali segmenti alcuni sono nascosti; altri sono totalmente visibili altri sono parzialmente visibili.

Per essere pronti a considerare anche la terza possibilità, dobbiamo essere in grado di tracciare un segmento che unisce due punti, punto per punto (vedi figura 4). La formula per trovare il singolo punto intermedio tra P1 e P2 è

$$Y = Y1 + (X - X1) * T \text{ per } X \text{ che va da } X1 \text{ a } X2$$

dove T è l'angolazione del segmento

$$T = (Y2 - Y1) / (X2 - X1)$$

inoltre se $X2 < X1$ il loop va eseguito al

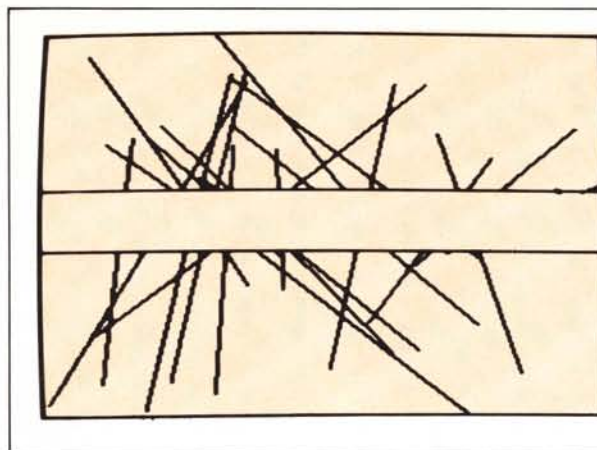


Figura 8 - Programma Fascia. Output. L'esecuzione è molto lenta, in quanto per ogni punto P(X,Y) va eseguita la routine di test.

```
100 DIM Y(279): HGR2: HCOLOR= 3
110 REM CARICAMENTO VETTORE YMAX
120 FOR I = 0 TO 279:Y(I) = 191: NEXT
130 HPLLOT 0,191 TO 279,191
140 REM LOOP PRINCIPALE
150 FOR K = 190 TO 10 STEP -10: HCOLOR= 3
160 X1 = RND (1) * 200:Y1 = RND (1) * K
170 X2 = X1 + RND (1) * 70 + 10:Y2 = Y1
180 REM TEST LINEE VERTICALI
190 IF Y1 < Y(X1) THEN HPLLOT X1,Y1 TO X1,Y(X1)
200 IF Y2 < Y(X2) THEN HPLLOT X2,Y2 TO X2,Y(X2)
210 FOR X = X1 + 1 TO X2:Y = Y1: GOSUB 250: NEXT X: NEXT K
220 REM TRACCIAMENTO SKYLINE
230 HGR2: HCOLOR= 3: HPLLOT 0,191 TO 279,191: HPLLOT 0,191
240 FOR X = 1 TO 279: HPLLOT TO X,Y(X): NEXT: END
250 REM TEST YMAX
260 HCOLOR= 0: IF Y < Y(X) THEN Y(X) = Y: HCOLOR= 3
270 HPLLOT X,Y: RETURN
```

Figura 9 - Programma SKYLINE. Listato. Il test punto nascosto è nelle righe 190, 200 per le linee verticali del rettangolo raffigurante il grattacielo e in riga 250 e successive per i punti relativi ai segmenti orizzontali.

contrario, ponendo lo step =-1.

Ma se l'inclinazione del segmento è elevata accade che per un piccolo incremento della X c'è un incremento grande della Y e quindi il segmento tracciato risulterà "sgranato". Uno dei modi per evitare tale inconveniente è quello di eseguire il loop sulla Y nel caso che l'inclinazione sia maggiore di 45°.

Abbiamo realizzato un programma che esegue in modo random una serie di segmenti, tracciandoli punto per punto sul monitor dell'Apple II. (fig. 5 l'output e fig. 6 il listato), si notano i due casi di inclinazione del segmento, inoltre si nota la protezione contro il caso DIVISION BY ZERO, che si verificherebbe per coppie di punti in cui $X1 = X2$.

Il metodo dei massimi e dei minimi

In questo articolo vogliamo introdurre il procedimento dei massimi e dei minimi per l'eliminazione delle linee nascoste.

Una prima applicazione, la più semplice possibile che utilizziamo a scopo esemplificativo è costituita dalla implementazione del precedente programma. Immaginiamo che sul foglio su cui tracciamo le nostre linee random ci sia una fascia che nasconde una porzione del monitor. Per vedere se

ciascun punto di ciascun segmento sia "nascosto" dalla fascia basta eseguire un test dentro/fuori fascia.

Nel nostro caso la fascia è individuata da due vettori in cui sono inseriti per ogni X possibile sul monitor il valore del bordo superiore della fascia YM(X) e il valore del bordo inferiore YN(X).

Il programma relativo è listato in figura 7 e l'output è in figura 8. Le routine in più rispetto al programma precedente sono quelle dell'inizializzazione delle fasce (riga 120) e del test dentro/fuori eseguita su ogni punto da tracciare (riga 320).

Il programma skyline

L'ulteriore passo logico è intuitivo. Se via via che viene realizzato sullo schermo il disegno "aggiorniamo" il vettore dei massimi e dei minimi, possiamo eseguire per ogni punto il test di visibilità. Questo sistema funzionerà a condizione che i segmenti siano tracciati in ordine di vicinanza dall'osservatore.

Una esemplificazione è il programma SKYLINE, con il quale ipotizziamo di disegnare lo skyline di una città con grattacieli sul monitor Apple II.

I grattacieli sono larghi da 10 a 80 e sono

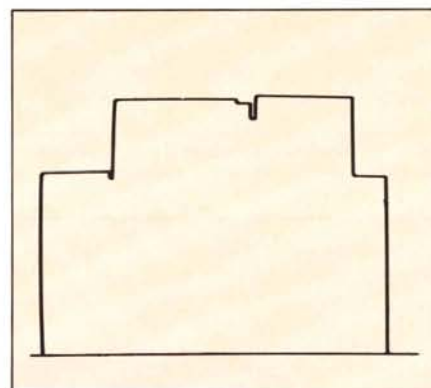
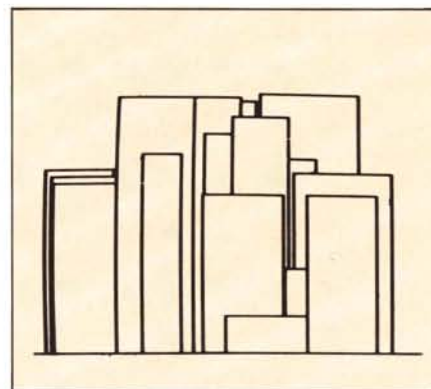


Figure 10 e 11 - Programma SKYLINE. Output. Nel primo output vengono disegnati uno dopo l'altro tutti i 19 grattacieli a partire da quello più vicino, nel secondo viene tracciato lo skyline vero e proprio corrispondente alla linea individuata da tutti gli YM.

alti da 10 a 180 pixel Apple II. Ogni punto P(X,Y) da tracciare viene confrontato con il valore del vettore dei massimi, se il valore della Y è minore di YM(X) allora il punto è nascosto, se è maggiore non è nascosto, allora viene disegnato e viene aggiornato il vettore dei massimi. Analogamente si esegue un test per le linee verticali dei grattacieli.

Disegnatili tutti, viene tracciata la linea skyline, che non è altro che la traccia del vettore dei massimi.

In figura 9 c'è il listato del programma, in figura 10 e 11 rispettivamente il disegno del "panorama" della città e la traccia dello skyline della stessa.


```

100 REM INIZIALIZZAZIONI
110 HL = 1: REM FLAG HIDDEN LINES SI/NO
120 HGR2: HCOLOR= 3: HPLLOT 0,0: CALL 62454: HCOLOR= 0
130 NI = 15: NJ = 15: XC = 140: YC = 90
140 DIM Y(280): FOR I = 0 TO 279: Y(I) = 191: NEXT I
150 D1 = - 20: D2 = - 35
160 PI = 3.14159: SP = PI / 5: IZ = - 18
170 REM DATI SCALING PRECALCOLATI
180 SC = 17.64: XN = - 7.37: YN = 8.64: DX = 14.74: DY = 6.2
190 REM ROUTINE DISEGNO
200 FOR B = - NJ TO NJ: FOR A = - NI TO NI
210 X0 = A * SP: Z0 = B * SP
220 R = SQR ((X0 ^ 2 + Z0 ^ 2)) / 2
230 Y0 = SIN (5 * R) / (R + .002) + 10
240 I = H + NI: J = B + NJ
250 X = (D1 * X0) / (D2 - Z0)
260 X = (X - XN - DX / 2) * SC + XC
270 Y = - (D1 * Y0) / (D2 - Z0)
280 Y = (Y - YN - DY / 2) * SC + YC
290 IF A = - NI THEN X2 = X: Y2 = Y: GOTO 310
300 X1 = X2: Y1 = Y2: X2 = X: Y2 = Y: GOSUB 320
310 NEXT A, B: END
320 REM RICERCA HIDDEN
330 IF HL THEN HPLLOT X1, Y1 TO X2, Y2: RETURN
340 IF X1 = X2 THEN T = 99999: GOTO 360
350 T = (Y2 - Y1) / (X2 - X1): S = 1
360 IF ABS (T) > 1 THEN 420
370 IF X1 > X2 THEN S = - 1
380 FOR X = X1 TO X2 STEP S
390 Y = Y1 + (X - X1) * T + .5
400 GOSUB 470: IF F THEN HPLLOT X, Y
410 NEXT X: RETURN
420 IF Y1 > Y2 THEN S = - 1
430 FOR Y = Y1 TO Y2 STEP S
440 X = X1 + (Y - Y1) / T + .5
450 GOSUB 470: IF F THEN HPLLOT X, Y
460 NEXT Y: RETURN
470 F = 0: IF Y < Y(X) THEN Y(X) = Y: F = 1
480 RETURN

```

Figura 12 - Programma funzione. Listato. La funzione da visualizzare è posta nelle righe 220/230, mentre in riga 180 sono posti i valori precalcolati di scaling.

scaling e la conseguente immissione delle cinque costanti necessarie. L'aver precalcolato le costanti di scaling evita la necessità di ricorrere a vettori e matrici per l'immagazzinamento dei dati da visualizzare.

In sostanza il programma calcola il disegno e lo visualizza direttamente.

In riga 110 c'è il flag HL che, se posto uguale a 1, esclude il test delle hidden lines dal disegno.

In riga 130 ci sono i valori NI e NJ che indicano la lunghezza dei loop sulla A e sulla B. Poiché i loop vanno da -NI a NI e da -NJ a NJ i punti spaziali calcolati per NI=NJ=15 sono 31^2 cioè 961.

Inoltre notiamo che, poiché stiamo visualizzando una funzione trigonometrica, i nostri loop principali, eseguiti con valori interi di A e di B, vanno tradotti in valori angolari e questo lo otteniamo semplicemente moltiplicando A e B per una frazione di Pgreco.

In riga 140 c'è la inizializzazione dei valori Y massimi. Nel caso Apple II in realtà si tratta di minimi poiché, come noto, l'asse Y va dall'alto verso il basso.

La funzione $Y = Y(X, Z)$ da visualizzare è posta nelle righe 220, 230 utilizzando la variabile intermedia $R = R(X, Z)$ che in pratica è la distanza del punto P(XYZ) dall'origine delle coordinate X e Z ovvero dall'asse Y; viene in definitiva calcolata una funzione $Y = Y(R)$.

Ricavati i valori di due punti in sequenza P1 e P2 e quindi da collegare, tra questi vengono tracciati i segmenti, con il sistema punto punto visto nel programma precedente.

Nella subroutine da riga 320 c'è il test hidden, che viene evitato ponendo HL = 1. Dopo la suddivisione dei segmenti da tracciare in punti contigui, viene eseguito il test

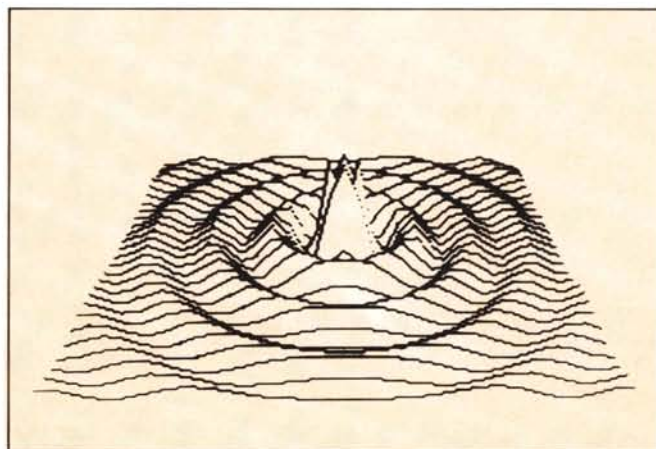
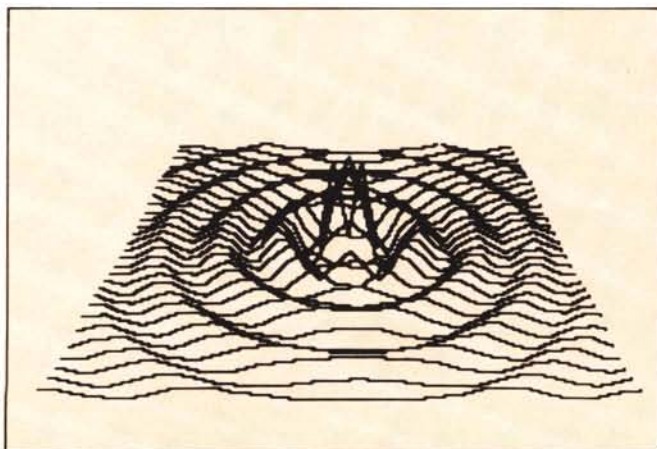


Figure 13 e 14 - Programma Funzione. Output. La differenza tra i due output è che nel primo non sono eliminate le linee nascoste e nel secondo sì.

Il programma funzione tridimensionale

Lo stesso procedimento dei massimi e dei minimi può essere applicato con successo alla visualizzazione delle funzioni $Y = Y(X, Z)$.

Il singolo punto spaziale della funzione P(XYZ) si calcola inserendo la funzione

$Y = Y(X, Z)$ all'interno di due loop sulla X e sulla Z. Per ogni punto così calcolato si esegue, con il metodo della prospettiva illustrato nei numeri scorsi, la traduzione in coordinate schermo. A questo punto si esegue il test di visibilità sul punto-schermo.

Esaminando il listato di figura 12, notiamo l'eliminazione della pesante routine di

per ognuno di questi punti in riga 470.

Viene posto uguale a zero il flag F, se il punto P(XY) è esterno alla YM(X) la YM viene aggiornata, il flag settato a 1 e quindi disegnato il punto.

Di tale programma pubblichiamo sia l'output con le linee nascoste, sia quello senza le linee nascoste (figg. 13 e 14). 

RIVENDITORI, SOFTWARE-HOUSE, CONSULENTI EDP

Eurocom vi propone



Rainbow 100

personal computer a 8-16 bit (è basato su CPU Z80 e 8088), con memoria RAM fino a 256 K bytes, sistemi operativi CP/M* e MSDOS e 1,6 M bytes di memoria di massa su drives 5" 1/4 più disco Winchester esterno opzionale da 5 M bytes.

*CP/M™ è un prodotto della Digital Research.

digital

Professional 325 Professional 350

personal computers 16 bit basati su CPU F11 (PDP11/23), con memoria RAM fino a 256 K bytes, sistema operativo P/OS (derivato dal collaudatissimo RSX della famiglia PDP-11).

Il modello 325 lavora con memoria di massa di 1,6 M bytes su drives 5" 1/4; il modello 350 utilizza invece un disco Winchester da 5" 1/4, per un totale di 5 M bytes in linea, più 2 drives 5" 1/4, per minifloppy, per ulteriori 800 K bytes.

la nuova famiglia di personal computer Digital

Eurocom vi garantisce il suo supporto tecnico e commerciale ed un vasto listino di software applicativo per le esigenze dei vostri clienti



eurocom