

COMPUTER GRAFICA APPLICATA - GRAFICA TRIDIMENSIONALE

Vi sono vari metodi per la rappresentazione di un oggetto tridimensionale su di una superficie bidimensionale, tutti ampiamente illustrati nei libri di disegno oppure nei libri di geometria descrittiva.

Per chi voglia affrontare il problema con un microcomputer "equipaggiato" con uscita grafica su monitor o su carta, la lettura di questi libri è certo utile ma, a mio parere, non risolutiva. Infatti i libri di disegno trattano l'argomento riferendosi a metodi grafici difficilmente traducibili in programmi, così come i libri di geometria descrittiva, trattando l'argomento da un punto di vista generale, tramite sistemi di vettori posti in sistemi di riferimento differenti, allontanano troppo dalla visione pratica del problema.

In altre parole, con un microcomputer non si possono utilizzare procedure solo grafiche, se non sono traducibili in un algoritmo che permetta il calcolo delle coordinate, né è conveniente utilizzare sistemi complessi di geometria descrittiva che appesantiscono inutilmente la programmazione e rallentano l'esecuzione.

Abbiamo quindi realizzato un sistema di rappresentazione che semplifica al massimo la parte "calcolo" con gran beneficio sui tempi di esecuzione e, come si potrà comprendere nel testo e vedere dalle illustrazioni, risolve in maniera brillante quasi tutti se non tutti i problemi tipici di rappresentazione prospettica di oggetti tridimensionali.

Un po' di teoria

I metodi classici di rappresentazione di un oggetto tridimensionale sono tre:

- metodo delle proiezioni ortogonali
- metodo delle proiezioni assonometriche
- metodo della prospettiva centrale

Il primo metodo consiste nel proiettare l'oggetto da distanza infinita su piani tra di loro ortogonali, ottenendo così tre immagini in ciascuna delle quali l'oggetto appare rappresentato in due dimensioni. È un metodo particolarmente utilizzato per il

disegno tecnico in quanto fornisce le reali dimensioni dell'oggetto. L'esame della figura 1 chiarirà il concetto.

Il secondo metodo consiste nel proiettare l'oggetto su un piano da distanza infinita. L'oggetto deve essere posto in posizione tale da far risultare una immagine a tre dimensioni. Tale rappresentazione è alquanto espressiva ma del tutto insufficiente a far percepire la reale forma dell'oggetto. Vedi figura 2.

Il terzo metodo, vedi figura 3, consiste nel riprodurre su un piano (detto quadro) una immagine simile a quella percepita

dall'occhio dell'osservatore posto a distanza finita.

Nell'articolo svilupperemo questo ultimo metodo secondo un algoritmo supersemplificato, che però permette tutte le implementazioni classiche necessarie per far muovere l'oggetto rispetto all'osservatore.

Il nostro metodo

L'oggetto che vogliamo rappresentare è una casa (vedi fig. 4) che da un punto di vista grafico è costituita da 15 vertici uniti tra di loro con 26 segmenti. Cioè se la vogliamo disegnare su un foglio di carta con una matita dobbiamo unire i 15 vertici con 26 tratti.

Ma poiché ogni segmento è individuato da due vertici, l'oggetto è in realtà individuato solo dai vertici e dalla informazione di quali coppie di vertici sono unite tra di loro.

Il primo problema fondamentale è quindi quello di trovare una regola matematica che permetta di tradurre ogni vertice della casa fornito in coordinate tridimensionali $P(X,Y,Z)$ in opportune coordinate schermo $P(XS,YS)$.

Ipotizzando gli assi posti come nel disegno, ovvero posizione iniziale dell'osservatore sull'asse Z, piano di rappresentazione XS, YS parallelo al piano XY, troveremo delle facili formule che esprimono le coordinate $P(XS, YS)$ sullo schermo di ciascun

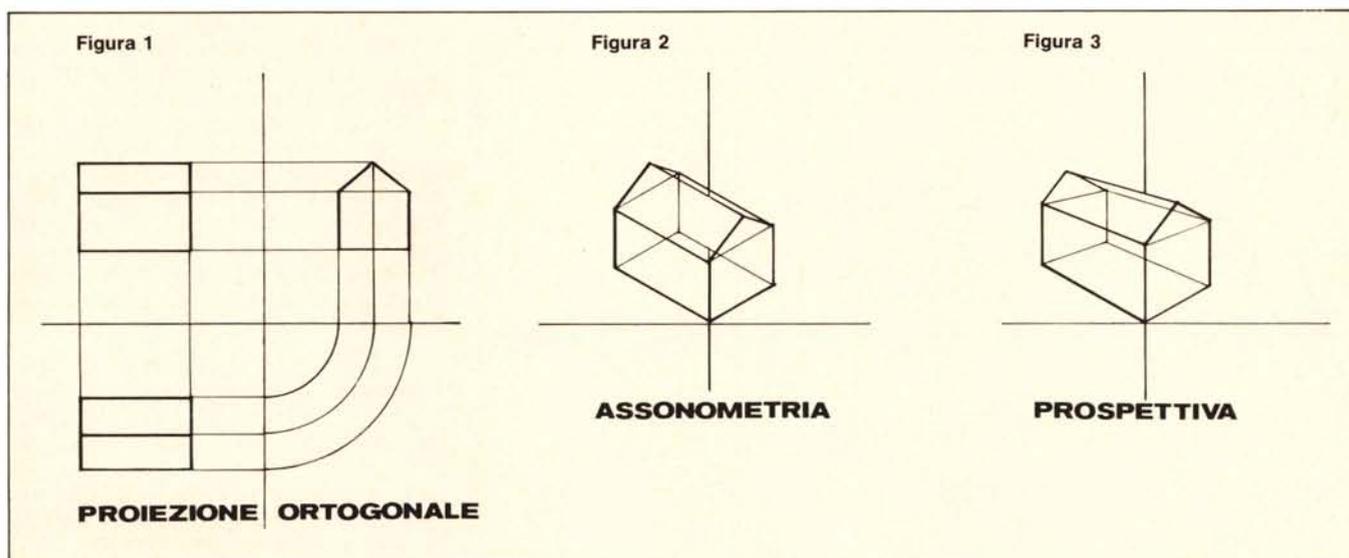


Figura 1 - PROIEZIONE ORTOGONALE - Metodo di rappresentazione di un oggetto tridimensionale su una superficie bidimensionale, particolarmente utilizzato per il disegno tecnico, in quanto fornisce le reali dimensioni dell'oggetto.

Figura 2 - ASSONOMETRIA - Le dimensioni lineari dell'oggetto sono quelle reali lungo i tre assi, sono viceversa falsati gli angoli.

Figura 3 - PROSPETTIVA - È il metodo, trattato in questo articolo, che fornisce l'immagine più vicina a quella percepita dall'occhio umano.

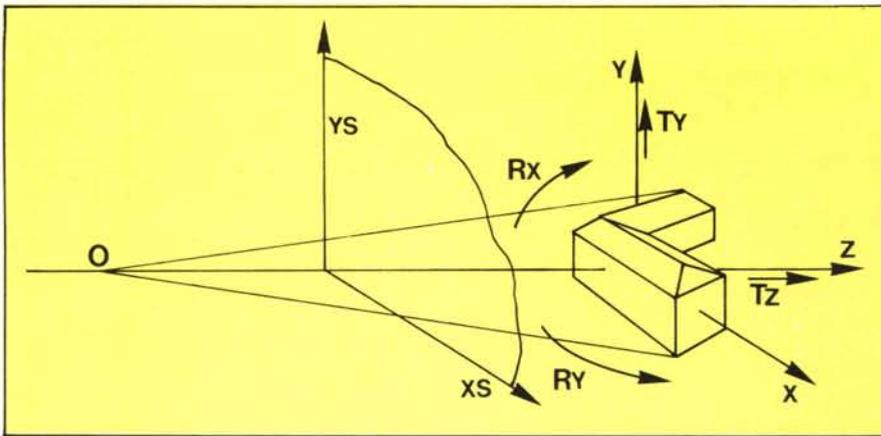


Figura 4 - L'OGGETTO NELLO SPAZIO TRIDIMENSIONALE - Analisi nello spazio del problema della prospettiva. O è l'osservatore, XS, YS il piano schermo e TZ, TY, RX, RY i movimenti dell'oggetto implementati nei programmi.

punto P(XYZ) dell'oggetto nello spazio tridimensionale:

$$XS = \frac{X * D1}{D2 + Z}$$

$$YS = \frac{Y * D1}{D2 + Z}$$

Il tutto si comprende facilmente esaminando la figura 5, dove il problema viene limitato al caso bidimensionale.

Queste formule sono valide anche se le grandezze in gioco sono legate a opportune leggi di variazione. È con queste leggi che possiamo simulare tutte le posizioni relative possibili nello spazio tra osservatore, schermo e oggetto.

Nei programmi realizzati abbiamo trovato quattro di queste leggi:

- Traslazione lungo l'asse Z
l'allontanamento dell'osservatore dall'oggetto si ottiene aumentando la grandezza D2

- Traslazione lungo l'asse Y
l'innalzamento dell'osservatore lungo la

verticale si ottiene abbassando il soggetto. Basta quindi diminuire via via il valore della coordinata Y del punto.

- Rotazione rispetto all'asse X
la rotazione si ottiene ponendo le coordinate XZ dei punti della casa in un loop di rotazione attorno all'asse Y. Le coordinate Y rimangono invariate.

Il punto P(X1, Y1, Z1) originario, ruotando il sistema di riferimento attorno all'asse Y, di un angolo A, assume la posizione P(X2, Y2, Z2) individuata dalle formule:

$$X2 = X1 * \cos(A) + Z1 * \sin(A)$$

$$Y2 = Y1$$

$$Z2 = -X1 * \sin(A) + Z1 * \cos(A)$$

- Rotazione rispetto all'asse X
è equivalente alla precedente, solo che la rotazione riguarda le coordinate YZ, mentre le coordinate X rimangono fisse.

Il programma base

Prima di approfondire gli argomenti ora accennati, esaminiamo nel dettaglio il pro-

gramma BASE, che è quello nel quale sono riassunti i concetti esposti e che servirà appunto come BASE per le numerose implementazioni che vogliamo realizzare.

Nelle righe 10-90 vanno immesse tutte le caratteristiche sia dell'oggetto che vogliamo visualizzare, sia del punto di vista dal quale vogliamo vederlo, sia del formato della "finestra" entro il quale vogliamo vederlo.

ND numero di punti.

X%(I), Y%(I), Z%(I), F%(I) coordinate dei punti nello spazio e flag di informazione di inizio/fine spezzata.

A angolo di rotazione dell'oggetto rispetto all'asse Y.

Y2 altezza dell'osservatore rispetto al piano XZ.

D1 distanza tra l'osservatore e lo schermo.

D2 distanza tra l'osservatore e l'oggetto.

X9,Y9 dimensione output del disegno

X8,Y8 centro dell'output del disegno.

: Nelle righe 1000 e successive vanno inserite le coordinate di tutti i punti delle spezzate.

La parte elaborativa vera e propria comincia con la trasformazione di tutti i dati spaziali in dati bidimensionali e con la ricerca del massimo e del minimo delle coordinate XS%(I), YS%(I) necessarie per lo scaling del disegno (righe 100 - 220).

Nelle righe 230 - 260 viene trovata la scala per la quale moltiplicare ciascuna coordinata bidimensionale trovata, per farla ricadere nella dimensione voluta del disegno e con il centro voluto.

Infine dalla riga 300 in poi viene disegnata la vista prospettica desiderata della casetta, mediante le spezzate il cui inizio e la cui fine sono riconosciute dal flag F%(I), e mediante le coordinate bidimensionali trovate opportunamente scalate e traslate per farle entrare nel formato output prescelto.

Il programma ha un output su monitor APPLE II e quindi nel listato pubblicato in figura 6 prevede il classico formato grafico APPLE di 280 per 192 pixel.

Provate il programma dapprima copiando, per comodità, i dati già inseriti e cominciando la necessaria familiarizzazione cambiando a caso, o con vostre regole, le grandezze A, Y2, D1, D2 per spostare il punto di osservazione, oppure cambiando le grandezze X9, Y9 e X8, Y9 per spostare il punto di uscita sul monitor e per variare la finestra.

Con lo stesso programma BASE sono state realizzate, ma questa volta con uscita su plotter, anche le figure 7 e 8, che rappresentano da due punti di vista differenti la stessa barca, individuata tramite sezioni verticali rispetto alla linea di carena e tramite linee d'acqua (ovvero sezioni parallele alla superficie dell'acqua).

Per la cronaca la barca è il cutter a deriva mobile (infatti nel disegno non c'è) SAMORIC 41, progettata dall'architetto francese André Moric, delle dimensioni di

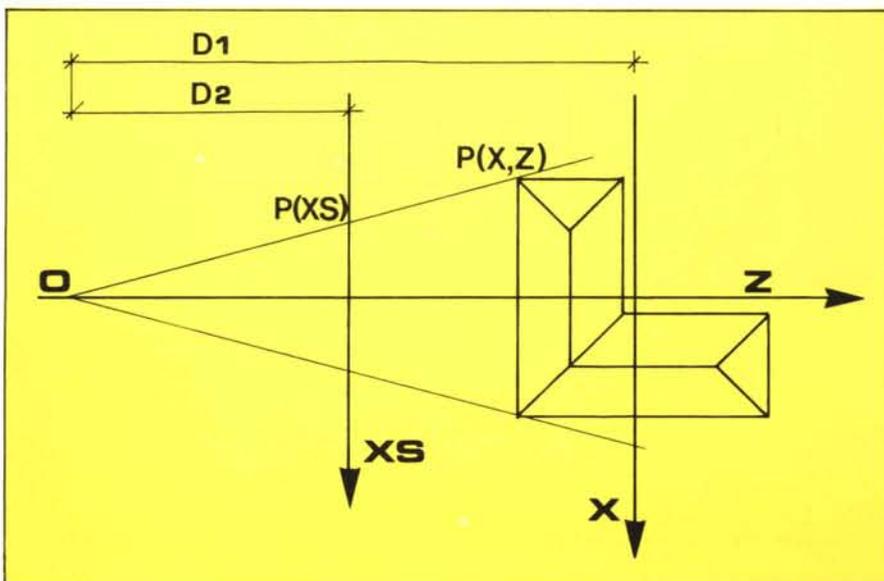


Figura 5 - Problema della prospettiva limitato al caso bidimensionale.

```

10 ND = 30: REM NUMERO DEI PUNTI
20 DIR X(ND), YS(ND), VS(ND), YSS(ND), ZS(ND), FX(ND)
30 FOR I = 1 TO ND: READ X(I), YS(I), VS(I), ZS(I), FX(I): NEXT I
40 A = 3: C = 105: R = 5: S = SIN(A): REM ROTAZIONE RX DELL'OGGETTO
50 X9 = 279: Y9 = 191: REM DIMENSIONI DISEGNO
60 X8 = 139: Y8 = 95: REM CENTRO DEL DISEGNO
70 Y2 = 600: REM ALTEZZA OSSERVATORE DAL PIANO XZ
80 D1 = 400: REM DISTANZA OSSERVATORE SCHERMO
90 D2 = 800: REM DISTANZA OSSERVATORE OGGETTO
100 REM CALCOLO PUNTI
110 XM = -9999: XN = 9999: YH = -9999: YN = 9999
120 FOR I = 1 TO ND
130 X1 = X(I) + C + Z(I) + S
140 Y1 = YS(I) - Y2
150 Z1 = -X(I) + S + Z(I) + C
160 XS(I) = (D1 + X1) / (D2 - Z1)
170 YS(I) = -(D1 + Y1) / (D2 - Z1)
180 IF XS(I) < XN THEN XN = XS(I)
190 IF XS(I) > XM THEN XM = XS(I)
200 IF YS(I) < YN THEN YN = YS(I)
210 IF YS(I) > YH THEN YH = YS(I)
220 NEXT I
230 REM CALCOLO TATT, SCALA E TRASLAZIONE
240 DX = XM - XN: DY = YH - YN
250 SX = X9 / DX: SY = Y9 / DY
260 SC = SX: IF SX > SY THEN SC = SY
300 REM DISEGNO
310 HGR2: HCOLOR = 3: FOR I = 1 TO ND
320 X = (XS(I) - XN - DX / 2) * SC + X8
330 Y = (YS(I) - YN - DY / 2) * SC + Y8
340 IF NOT FX(I) THEN HPLLOT X:Y: GOTO 360
350 HPLLOT TO X:Y
360 NEXT I: END
1000 REM DATA
1010 DATA 0, -50, 0, 0
1020 DATA 0, -50, 150, 1
1030 DATA 100, -50, 150, 1
1040 DATA 100, -50, -100, 1
1050 DATA -100, -50, -100, 1
1060 DATA -100, -50, 0, 1
1070 DATA 0, -50, 0, 1
1080 DATA 0, 40, 0, 1
1090 DATA 0, 40, 150, 1
1100 DATA 100, 40, 150, 1
1110 DATA 100, 40, -100, 1
1120 DATA -100, 40, -100, 1
1130 DATA -100, 40, 0, 1
1140 DATA 0, 40, 0, 1
1150 DATA 50, 100, -50, 1
1160 DATA 100, 40, -100, 1
1170 DATA 100, -50, -100, 1
1180 DATA -100, -50, -100, 0
1190 DATA -100, 40, -100, 1
1200 DATA -50, 100, -50, 1
1210 DATA -100, 40, 0, 1
1220 DATA -100, -50, 0, 1
1230 DATA 0, -50, 150, 0
1240 DATA 0, 40, 150, 1
1250 DATA 50, 100, 100, 1
1260 DATA 100, 40, 150, 1
1270 DATA 100, -50, 150, 1
1280 DATA -50, 100, -50, 0
1290 DATA 50, 100, -50, 1
1300 DATA 50, 100, 100, 1

```

Figura 6 - PROGRAMMA BASE - LISTING - Nelle righe tra 10 e 100 vanno inseriti i dati base della prospettiva, e dalla riga 1000 in poi i DATA della figura da visualizzare.

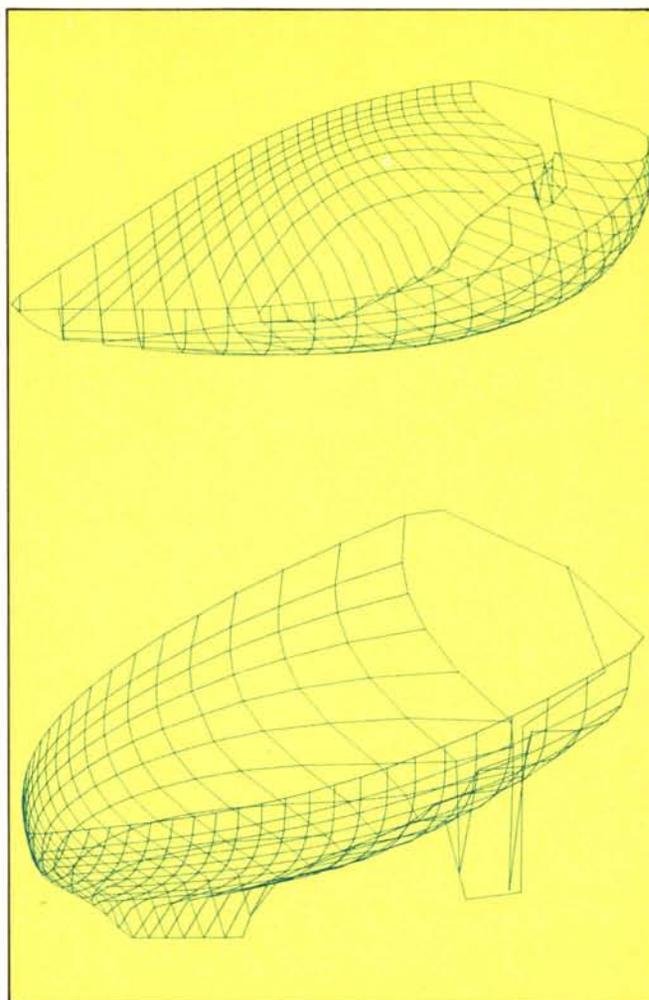


Figura 7 e 8 - PROGRAMMA BASE - OUTPUT SU PLOTTER - Due punti di vista differenti, dall'alto verso la poppa e dal basso verso la prua, della stessa barca.

12,45 metri di lunghezza per una larghezza di 4,10 metri. I dati inseriti nel programma (oltre 400) sono stati prelevati dai disegni esecutivi della barca.

È evidente la potenza che assume in questo caso il binomio calcolatore/plotter.

Con il nostro plotter Watanabe 4671, che è il più lento della famiglia (5 cm/sec.), utilizzato e descritto ampiamente in vari numeri precedenti, il tempo impiegato a disegnare lo scafo è stato di circa 8 minuti. I fratelli maggiori (40 cm/sec.) impiegherebbero molto meno a fare questo disegno!

Come immettere i dati

Quando tracciamo su un foglio di carta un quadrato possiamo farlo in diversi modi, anche se probabilmente neanche ce ne accorgiamo. O tracciando separatamente i quattro lati, oppure tracciando, senza alzare la penna dal foglio, i quattro lati in sequenza tornando al punto di partenza, o con metodi intermedi tra questi due.

Nel primo caso poiché ogni segmento è individuato da due punti abbiamo in realtà unito (a due a due) otto punti, nel secondo, invece, abbiamo unito in sequenza 5 punti, realizzando una spezzata chiusa in quanto

il 5° punto coincide con il primo.

È quindi evidente che se vogliamo realizzare un programma che disegni un quadrato o una figura anche molto complessa, dobbiamo trovare delle regole di tracciamento quanto più razionali possibile.

La regola più semplice, che è quella che abbiamo utilizzato nei nostri programmi, è quella di scomporre la figura, anche se tridimensionale, in una serie di spezzate, facendo in modo che passando da una spezzata all'altra, il tracciatore sollevi la penna.

Le regole che occorre seguire per individuare le spezzate sono empiriche in quanto si può, ad esempio, cercare di trovare meno spezzate possibile (e questo ottimizza i tempi di esecuzione) oppure utilizzare un sistema più razionale, ad esempio sezionando la figura con piani paralleli, per facilitare il lavoro preparatorio (e questo può aumentare anche di molto il numero delle spezzate).

In pratica i programmi presentati accettano un numero qualsiasi di punti, riconoscendo l'inizio della spezzata tramite il flag $F\%(I)=0$ che caratterizza ogni punto. Il flag posto = 1 significa invece che il punto in questione è collegato al precedente.

Per realizzare la casetta abbiamo dun-

que individuato (e si vede chiaramente dal listato di figura 6) 30 punti collegati con 4 spezzate. È chiaro che poiché da ciascun vertice della figura partono più di due spezzate, e quindi vi passano più di una spezzata, ogni vertice appare ripetuto in ogni spezzata cui appartiene.

Questo sistema è generalizzato, vale cioè per qualsiasi tipo di solido rappresentabile con spezzate.

È evidente quindi che una parte cospicua del lavoro consiste nella preparazione a tavolino dei dati base dell'elaborazione.

La sezione input del programma andrà ovviamente rivoluzionata quando fosse possibile stabilire delle regole elementari per definire i punti base del solido. Ad esempio per visualizzare (vedi la barca delle figg. 7 e 8) un oggetto mediante sezioni con piani orizzontali e con piani verticali, basterà tener conto che ciascun punto è collegato a 4 punti (due sullo stesso piano orizzontale e due sullo stesso piano verticale). Cioè in pratica, utilizzando queste regole elementari si può ottenere un notevole risparmio nella immissione (o se sono calcolabili, nel calcolo) dei dati.

Nei prossimi numeri della rivista approfondiremo questi concetti realizzando in-

```

100 DIM X%(100), Y%(100), Z%(100), F%(100): ND = 30
110 FOR I = 1 TO ND: READ X%(I), Y%(I), Z%(I), F%(I): NEXT I
120 D1 = 0: D2 = 0: D3 = 0: V1 = 0: V2 = 0: V3 = 0: GOSUB 850: TEXT = HOME
130 PRINT "ELABORAZIONE PROSPETTICA": PRINT "DI UN EDIFICIO"
140 PRINT "SIANO X, Y GLI ASSI DELLO SCHERMO"
150 PRINT "Z L'ASSE PERPENDICOLARE": PRINT "PRINT
160 PRINT "1 - FIGURA FISSA": PRINT
161 PRINT "2 - ROTAZIONE ASSE Y": PRINT
162 PRINT "3 - ROTAZIONE ASSE X": PRINT
163 PRINT "4 - TRASLAZIONE ASSE Z": PRINT
164 PRINT "5 - TRASLAZIONE ASSE Y": PRINT
165 PRINT "6 - SEQUENZA DEI QUATTRO MOVIMENTI": PRINT
166 PRINT "7 - FINE ": PRINT
170 PRINT TAB(20); "INPUT " SCEGLI "I, W
180 HGR2 ON W GOSUB 200, 300, 400, 500, 600, 700, 800
190 GOTO 120
200 REM FIGURA STATICA
210 D1 = 400: D2 = 500: V2 = 110: GOSUB 800: FOR I = 1 TO 30
220 X1 = X%(I): Y1 = Y%(I): Z1 = Z%(I): GOSUB 900: NEXT I: RETURN
300 REM ROTAZIONE ASSE Y
310 D1 = 300: D2 = 500: V2 = 110: FOR A = 0 TO 6.28 STEP .942
320 C = COS(A): S = SIN(A): GOSUB 800: FOR I = 1 TO 30
330 X1 = X%(I) * C + Z%(I) * S: Y1 = Y%(I): Z1 = Z%(I) * C - X%(I) * S
340 GOSUB 900: NEXT I: A: RETURN
400 REM ROTAZIONE ASSE X
410 D1 = 300: D2 = 500: V2 = 100: FOR A = 0 TO 3.14 STEP .471
420 C = COS(A): S = SIN(A): GOSUB 800: FOR I = 1 TO 30
430 X1 = X%(I) * V1 + Y%(I) * C + Z%(I) * S: Z1 = Y%(I) * S - Z%(I) * C
440 GOSUB 900: NEXT I: A: RETURN
500 REM TRASLAZIONE ASSE Z
510 D1 = 300: V2 = 110: A = 1.0: C = COS(A): S = SIN(A)
520 FOR D2 = 400 TO 800 STEP 100: GOSUB 800: FOR I = 1 TO 30
530 X1 = X%(I) * C + Z%(I) * S: Y1 = Y%(I): Z1 = Z%(I) * C - X%(I) * S
540 GOSUB 900: NEXT I: D2: RETURN
600 REM TRASLAZIONE ASSE Y
610 D1 = 300: D2 = 600: V2 = 80: A = 1.0: C = COS(A): S = SIN(A)
620 FOR V3 = 0 TO 500 STEP 100: GOSUB 800: FOR I = 1 TO 30
630 X1 = X%(I) * C + Z%(I) * S: Y1 = Y%(I) - V3: Z1 = Z%(I) * C - X%(I) * S
640 GOSUB 900: NEXT I: V3: RETURN
700 REM SEQUENZIALE
710 GOSUB 200: GOSUB 850: GOSUB 300: GOSUB 850: GOSUB 400
720 GOSUB 850: GOSUB 500: GOSUB 850: GOSUB 600: GOSUB 850: RETURN
800 REM BLANKAGGIO SCHERMO
810 HCOLOR = 3: HPLLOT 0,0: CALL 62454: HCOLOR = 4: RETURN
850 FOR K = 1 TO 499: NEXT: RETURN
890 HOME: TEXT: END
900 REM DISEGNO PUNTI
920 XS = (D1 + X1) / (D2 - Z1) + 140
930 YS = V2 - (D1 + Y1) / (D2 - Z1) - V3 / 2
940 IF NOT F%(I) THEN HPLLOT XS, YS: GOTO 960
950 HPLLOT TO XS, YS
960 RETURN
1000 REM SEGUONO I DATA

```

Figura 9 - PROGRAMMA INTERATTIVO - Il programma BASE è stato implementato per ottenere sul monitor, i quattro movimenti TZ, TY, RX, RY, uno alla volta o sequenzialmente.

```

100 DIM X%(99), XS%(99), Y%(99), YS%(99), Z%(99), F%(99): ND = 30
110 FOR I = 1 TO ND: READ X%(I), Y%(I), Z%(I), F%(I): NEXT I
120 DF = CHR$(4)
150 REM SONDATURA
160 FOR I = 0 TO 3: X = I + 800: Y = 0: GOSUB 900
170 X = I + 800: Y = 1800: GOSUB 950: NEXT I
180 FOR I = 0 TO 3: Y = I + 600: X = 0: GOSUB 900
190 Y = I + 600: X = 2400: GOSUB 950: NEXT I
200 REM IDENTIFICAZIONE RICHIEDI
210 FOR J = 0 TO 2: FOR L = 0 TO 2
220 XC = J + 800 + 400: YC = L + 600 + 300
230 REM ELEMENTI RANDOM
240 A = RND(1) * 6.28: C = COS(A): S = SIN(A)
250 V2 = RND(1) * 500
260 D1 = RND(1) * 300 + 300: D2 = RND(1) * 300 + 900
270 XM = -9999: XN = 9999: YN = -9999: YN = 9999: AF = ""
300 REM FORMAZIONE E TRACCIAMENTO SCRITTA
310 AF(1) = "E "
312 AF(2) = STR$(INT(V2))
314 AF(3) = "A ="
316 AF(4) = STR$(INT(A * 180 / 3.14))
318 AF(5) = "D1="
320 AF(6) = STR$(INT(D1))
322 AF(7) = "D2="
324 AF(8) = STR$(INT(D2))
330 FOR I = 1 TO 8: AF = AF + AF(I): NEXT
340 X = J + 800 + 20: Y = L + 600 + 20: GOSUB 900: GOSUB 980
400 REM CARICAMENTO DATI ELABORATI
410 FOR I = 1 TO ND
420 X1 = X%(I) + C + Z%(I) * S
430 Y1 = Y%(I) - V2
440 Z1 = -X%(I) * S + Z%(I) * C
450 XS%(I) = (D1 + X1) / (D2 - Z1)
460 YS%(I) = (D1 + Y1) / (D2 - Z1)
500 REM CALCOLO MAX/MIN
510 IF XS%(I) < XN THEN XN = XS%(I)
520 IF YS%(I) < YN THEN YN = YS%(I)
530 IF XS%(I) > XM THEN XM = XS%(I)
540 IF YS%(I) > YN THEN YN = YS%(I)
550 NEXT I
560 DX = XM - XN: DY = YN - YN
570 SX = 600 / DX: SY = 400 / DY
580 SC = SX: IF SX > SY THEN SC = SY
600 REM FORMAZIONE DATI E DISEGNO
610 FOR I = 1 TO ND
620 X = INT((XS%(I) - XN - DX / 2) * SC + XC)
630 Y = INT((YS%(I) - YN - DY / 2) * SC + YC)
640 IF NOT F%(I) THEN GOSUB 900
650 GOSUB 980: NEXT I: L: J: END
900 REM PLOTTER MOVE
910 PRINT D;"PR#1": PRINT "M,X", "Y": PRINT D;"PR#0"
920 RETURN
950 REM PLOTTER DRAW
960 PRINT D;"PR#1": PRINT "D,X", "Y": PRINT D;"PR#0"
970 RETURN
980 REM PLOTTER PRINT
990 PRINT D;"PR#1": PRINT "P,X": PRINT D;"PR#0": RETURN
1000 REM SEGUONO I DATA

```

Figura 10 - PROGRAMMA RANDOM - LISTING - Il programma realizza 9 uscite su plotter, in 9 riquadri, tre righe su tre colonne, realizzati con due loop uno dentro l'altro.

teressanti implementazioni del programma BASE.

Di quanto abbiamo semplificato il problema

Per individuare la posizione di un punto su una retta X, occorre una sola informazione, ovvero il valore della coordinata X, in questo caso si dice che il punto ha sulla retta un grado di libertà. Per individuare la posizione di un punto su un piano XY occorrono due coordinate, per individuare un punto nello spazio occorrono tre coordinate.

Per individuare la posizione di un solido nello spazio occorrono 6 coordinate, tre per individuare la posizione di un punto (ad esempio il baricentro) del solido e altre tre per specificare l'orientazione del solido rispetto ai tre assi. Il solido nello spazio ha 6 gradi di libertà.

Analogamente si dimostra che per individuare un piano nello spazio occorrono tre coefficienti e quindi il piano ha tre gradi di libertà.

In sostanza il problema della rappresen-

tazione prospettica da un punto di osservazione di un oggetto su di un piano ha in totale ben 12 gradi di libertà (tre per il punto di osservazione, tre per il piano, sei per l'oggetto). Questo vuol dire che esistono infinite alla dodicesima prospettive di un oggetto.

Un programma generalizzato deve essere in grado di mostrarle tutte.

Il programma da noi realizzato semplifica radicalmente il problema riducendo i gradi di libertà, ovvero i parametri da immettere per scegliere la prospettiva, a 5.

Sono, come abbiamo visto, D1, D2, Y2, A1, A2.

Modificando questi parametri si possono realizzare tutte le possibili posizioni reali relative tra osservatore, quadro e oggetto.

Il programma interattivo

Come implementazione del programma BASE ne abbiamo realizzati altri due: il programma INTERATTIVO che simula il movimento dell'oggetto sul monitor dell'APPLE II, e il programma RANDOM,

che calcola in modo random le grandezze A, D1, D2, Y2 e visualizza sul plotter 9 differenti prospettive della nostra casetta.

Il programma INTERATTIVO (listato in figura 9), dopo aver caricato i dati riferiti all'oggetto da visualizzare, posti nelle righe 1000 e seguenti (per testare il programma copiate quelli del programma BASE), offre una scelta di cinque possibilità:

- FIGURA FISSA (righe 200-220) - dove le quattro grandezze assumono valori predeterminati e quindi l'immagine risultante è fissa;
- ROTAZIONE ATTORNO ASSE Y (righe 300-340) - nella quale variano le coordinate X e Z dell'oggetto che ruota attorno all'asse Y, in un loop in cui varia l'angolo A. Per rendere più lenta la rotazione si può diminuire il valore dello step del loop;
- ROTAZIONE ATTORNO ASSE X (righe 400-440) - analoga alla precedente, solo che variano le coordinate Y e Z;
- TRASLAZIONE LUNGO ASSE Z (righe 500-540) - fissati D1, Y2 e l'angolo A, il loop riguarda D2 che viene

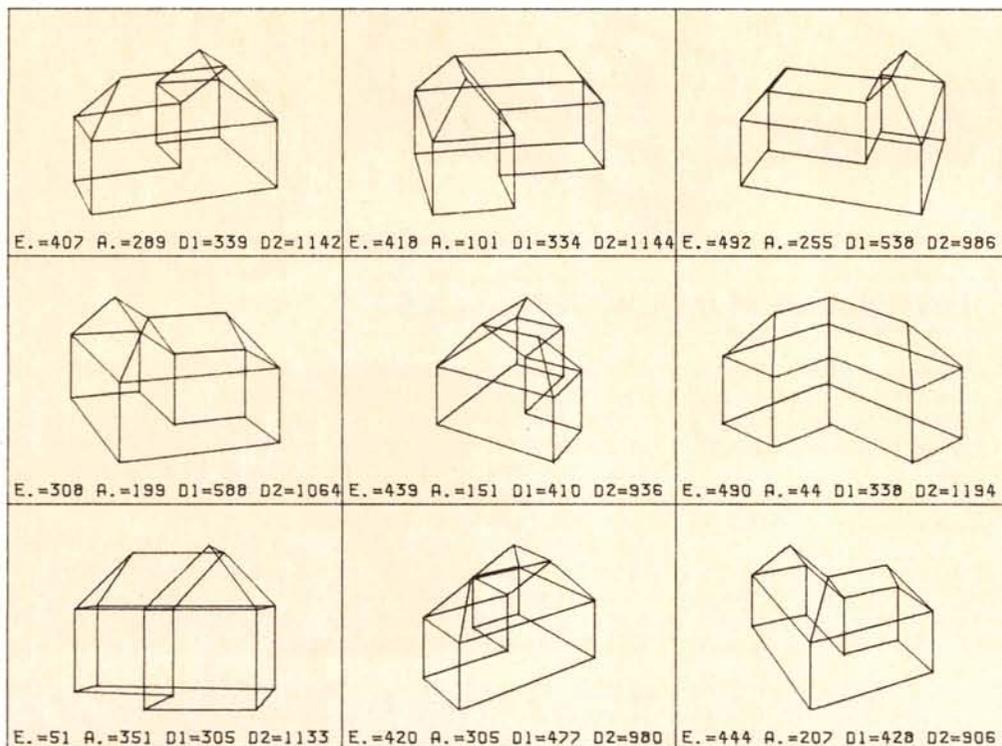


Figura 11 - PROGRAMMA RANDOM - OUTPUT - Punti di vista differenti, specificati con una stringa riportata in basso, della nostra casetta.

incrementato, producendo l'allontanamento dell'oggetto;

- TRASLAZIONE LUNGO ASSE Y (righe 600-640) - incrementando o decrementando le coordinate Y, si ottiene un innalzamento o un abbassamento dell'osservatore rispetto al piano orizzontale su cui giace l'oggetto.

Il programma con le grandezze già inserite non ha bisogno di correzioni di formato. Sono quindi eliminate, con vantaggio per i tempi di esecuzione, tutte le routine di

ricerca di max e min e di scaling.

La subroutine di riga 900, richiamata da tutti i sottoprogrammi, è quella con la quale vengono calcolati, partendo dai punti nello spazio, i punti sullo schermo e con la quale vengono disegnati.

Programma random

Il programma ha una uscita su plotter Watanabe. Le istruzioni output su plotter sono quelle inserite nelle righe 900-990 e

sono note a chi ha letto gli ultimi numeri della rivista.

Anche nel listato di questo programma (fig. 10) i dati base dell'oggetto non sono stati inseriti per brevità; potrete, per testare il programma, utilizzare quelli del programma BASE.

Viene dapprima disegnata una squadratura e quadrettatura del foglio (righe 150-190), con la quale vengono realizzate 9 finestre (tre righe per tre colonne) nelle quali inserire le 9 prospettive dello stesso oggetto.

La elevazione E, l'angolo A (rotazione attorno all'asse Y), e le distanze D1 (tra osservatore e schermo) e D2 (tra osservatore e oggetto) di ciascuna vista sono determinati in modo casuale.

Determinate tali grandezze vengono tradotte in una stringa e riportate nel singolo riquadro tramite una subroutine di PRINT.

I loop principali di identificazione righe e colonne sono in J ed in L (riga 210).

Per ogni riquadro quindi, identificato da un valore di J ed uno di L, vengono dapprima determinati gli elementi random, poi viene composta e tracciata la scritta di riferimento, poi vengono calcolate le coordinate schermo, con il loop in I, con identificazione dei valori max e min delle XS e YS. Poi ancora per ogni vista e quindi in ogni riquadro il programma esegue lo scaling e una centratura del disegno nel suo interno e infine la visualizzazione.

Il programma, listato in figura 10 e il cui output è in figura 11, produce, se eseguito più volte, viste sempre differenti, questo naturalmente dipende dal fatto che le grandezze sono determinate in modo random.

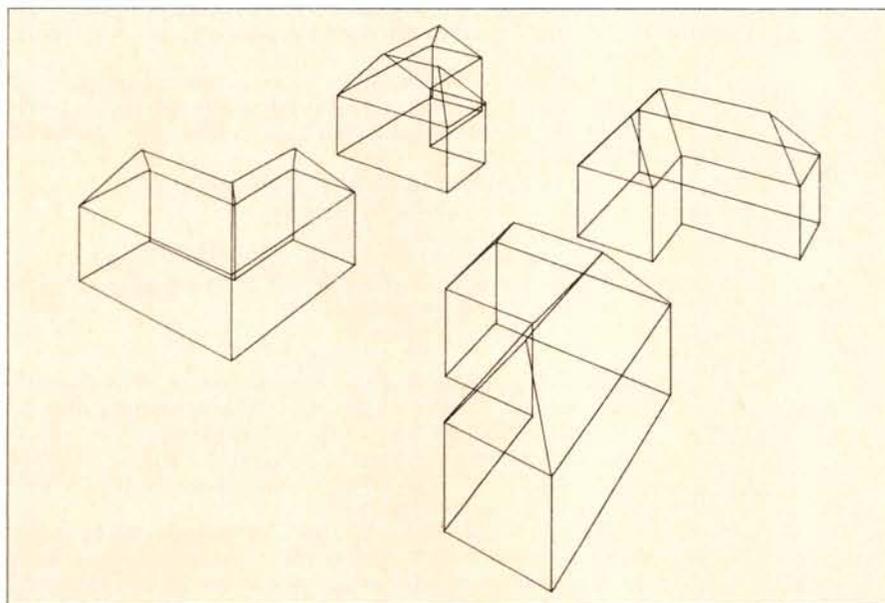


Figura 12 - PROGRAMMA BASE - OUTPUT MULTIPLIO SU PLOTTER - Altra implementazione elementare è quella di utilizzare gli stessi dati per disegnare più volte lo stesso oggetto, aumentando ad esempio semplicemente una sola coordinata.