

Animazioni con il microcomputer

Si parla molto di TRON, il film "fatto con il computer", realizzato dalla WALT DISNEY PRODUCTION.

Articoli su TRON appaiono su riviste di cinema, di televisione, di informatica e di microinformatica, e in genere sono corredati da suggestive fotografie a colori.

Le immagini del film sono state create dal computer secondo la tecnica detta "Computer Imaging". Con la Computer Imaging si possono realizzare i più inverosimili e spettacolari "effetti speciali", ancora più spettacolari di quelli a cui gli ultimi film di fantascienza ci avevano abituato. Oppure si può usare la Computer Imaging per realizzare, del tutto automaticamente, un cartone animato, i cui fotogrammi siano generati dal computer. Non tutto TRON è stato realizzato con il computer, sono state infatti largamente adottate tecniche che permettono la composizione di immagini riprese dal vivo con immagini riprese da disegni, da fotografie o da video grafici. Ma l'effetto finale è molto più appariscente e spettacolare di quello, pur sempre affascinante, offerto da un video grafico a colori ad alta definizione ed alta velocità.

Noi vogliamo solamente affrontare il problema dell'animazione realizzata tramite computer, per vedere se e in quale misura il problema è risolvibile con un microcomputer. Ma prima di addentrarci nelle problematiche connesse con la creazione delle immagini, valutiamo quantitativamente il problema. L'unità di misura tipica della Computer Grafic è, come ormai tutti sanno, il pixel, singolo punto visualizzabile. Quindi la prima valutazione che va fatta è quella sul numero dei pixel formanti l'immagine video. Video Grafici professionali vanno da 512 x 512 in su. Questo numero va poi moltiplicato per una certa quantità di piani di memoria ovvero immagini contemporaneamente immagazzinate nella memoria del computer e visualizzabili individualmente o contemporaneamente secondo una certa gerarchia di sovrapposizione. Per quanto riguarda il colore, questo dipende da quanti bit sono associati ad ogni pixel. Un bit due colori, due bit quattro colori, tre bit otto colori, ecc.

È poi una caratteristica abituale dei sistemi grafici professionali quella di offrire una larga scelta di colori definibili. Questo è indispensabile per creare effetti di "sfumatura" e di "ombreggiatura" (ad esempio 16 colori utilizzabili contemporaneamente, scelti su una tavolozza di 4.096 differenti colori). Facendo i "conti in tasca" ad un

sistema grafico professionale medio, per una risoluzione 1280 x 1024, sedici colori (quindi x 4) e con otto piani di memoria (quindi x 8), necessitano 1280 x 1024 x 4 x 8 bit ovvero 5 mega byte.

Risulta da questi conti chiaro come la computer grafica professionale richieda hardware specifico e molto potente. Per quanto riguarda il software grafico di base, dovendo essere necessariamente adeguato alle capacità dell'hardware, comprende tutte le istruzioni classiche (zoom, fill, rotate, scale, ecc.) e deve essere utilizzabile direttamente con i linguaggi matematici più veloci.

Nel confrontare le potenzialità di un microcomputer dotato di funzioni grafiche, rispetto a quelle di un computer grafico professionale arriviamo alle stesse conclusioni a cui arriviamo quando confrontiamo le prestazioni di un personal con quelle di un mainframe nel campo gestionale.

```
10 HGR2
20 HCOLOR= 0: GOSUB 40: I = I + 1
30 HCOLOR= 3: GOSUB 40: GOTO 20
40 HPLLOT I, 0 TO 279 - I, 191: RETURN
```

Figura 1 - LINEAUO - listato. Il programma fa vedere un segmento che ruota attorno al centro del video. L'animazione è realizzata visualizzando e cancellando le varie immagini successive.

```
100 REM GESTIONE ERRORE
110 ONERR GOTO 210
120 REM INIZIO
130 HOME : S = 1: HGR2 : HCOLOR= 3
140 REM PAGINAZIONE
150 POKE 230, 64: CALL 62450
160 GOSUB 190: POKE 49237, 0
170 POKE 230, 32: CALL 62450
180 GOSUB 190: POKE 49236, 0: GOTO 140
190 HPLLOT K, 0 TO 279 - K, 191
200 K = K + S: RETURN
210 REM INVERSIONE DIREZIONE
220 S = - S: K = K + 1 * S: RESUME
```

Figura 2 - LINEADUE - listato. L'output è simile al precedente. L'animazione è qui realizzata con il salto da una pagina grafica all'altra. In quella non visibile il segmento viene cancellato e ridisegnato nella nuova posizione.

1	20	20	59.4642751	3.48461394
2	56	36	20.3960781	3.33898556
3	68	36	8.94427191	3.60523761
4	136	20	63.2455532	-321.750554
5	144	24	69.8569968	-231090667
6	140	32	64.498062	-124354995
7	76	56	16	1.570795
8	68	56	17.8885438	2.03444128
9	60	52	20	2.49808889
10	12	28	65.1152824	3.32693795
11	12	24	65.96969	3.38656867
12	20	20	59.4642751	3.48461394

Figura 4 - DATIROTAZ - output alfanumerico. Il primo numero è il progressivo del dato, poi sono stampate le due coordinate cartesiane, e poi Raggio ed Angolo, che sono le coordinate polari.

Cioè il micro può fare tutto quello che fa il mini o il grande elaboratore solo che lo può fare in scala ridotta e con tempi di elaborazione molto più lunghi.

I problemi e gli algoritmi per risolverli sono sostanzialmente gli stessi e quindi se li studiamo e li risolviamo sui piccoli computer possiamo anche capire come si affrontano e si risolvono sui grandi.

La prima questione fondamentale è sapere in quale modo va presentato il prodotto finito. Se va presentato su pellicola cinematografica, ci si avvicina alle tecniche di realizzazione dei cartoni animati, solo che le singole immagini sono realizzate dal computer. Questa tecnica presenta le difficoltà del trasferimento dell'immagine dal video alla pellicola. Per compiere questa operazione nel migliore dei modi si usano apparecchiature prodotte da case specializzate in Computer Grafica e molto costose. Sono macchine che si collegano da una parte all'uscita video del computer e dall'altra all'ottica di una macchina fotografica o cinematografica. Tramite un altro collegamento tipico delle periferiche (ad esempio un seriale RS 232), è possibile ovviamente comandare da computer il funzionamento della macchina.

In questo modo il tempo di creazione delle singole immagini è del tutto indipendente dal tempo di visualizzazione in fase di proiezione. Cioè se per creare l'immagine il computer impiega 30 secondi, dovremo scattare un fotogramma ogni 30 secondi. Quindi per realizzare un minuto di film (1440 fotogrammi proiettati a 24 fotogrammi al secondo), nel nostro caso occorrebbero 12 ore di elaborazione.

Da questi conti si capisce come sia indispensabile avere un computer veloce, e soprattutto avere la ripresa gestita in toto dal computer. Se si possiede un monitor grafico di buona qualità (oppure se si vuol fare un film in bianco e nero o a due colori) i risultati ottenibili riprendendo direttamente con la cinepresa il video sono soddisfacenti e alla portata di ogni cineamatore.

Si perde un po' di definizione ai bordi, a causa della curvatura del video, riscontrabile di più se questa è composta principalmente da linee rette. Occorre una cinepresa dotata di scatto singolo e occorre soprattutto fare delle prove per ottimizzare la distanza di ripresa, i filtri da utilizzare per ammorbidire la luminosità del pixel, tipo di pellicola, tempi e diaframmi di esposizione. Se le immagini vanno presentate direttamente sul video del computer in fase di elaborazione, si devono superare due difficoltà. Non si può fare animazione, cioè immagini in movimento, con il movimento che appaia continuo e non a scatti, se non realizzando almeno 20 immagini al secondo. Inoltre, l'immagine deve essere visualizzata già formata, in quanto, anche se è molto semplice, la sua realizzazione non è mai istantanea.

La seconda difficoltà è superabile solo quando si dispone di almeno due pagine grafiche indipendenti l'una dalle altre, sia


```

100 REM CARICAMENTO COORD. CARTESIANE
110 NI = 12: DIM X%(NI), Y%(NI), R(NI), A(NI): P = 3.14159
120 FOR I = 1 TO NI: READ X%(I), Y%(I): NEXT I
150 REM VISUALIZZAZIONE DATI CARTESIANI
160 HGR2: HCOLOR= 3
170 FOR I = 1 TO NI - 1
180 HPLLOT X%(I), Y%(I) TO X%(I + 1), Y%(I + 1)
190 NEXT I
200 REM CALCOLO DELLE COORDINATE POLARI
210 X% = 76: Y% = 40
220 FOR I = 1 TO NI
230 DX = X%(I) - X%(I-1): DY = Y%(I) - Y%(I-1)
240 R(I) = SQR (DX ^ 2 + DY ^ 2)
250 IF DX = 0 THEN A(I) = P / 2: GOTO 280
260 A(I) = ATN (DY / DX)
270 IF DX < 0 THEN A(I) = A(I) + P
280 NEXT I
300 REM VISUALIZZAZIONE
310 HGR2: HCOLOR= 3
320 HPLLOT X% - 2, Y% TO X% + 2, Y%: HPLLOT X%, Y% - 2 TO X%, Y% + 2
330 FOR I = 1 TO NI - 1
340 X1% = X% + R(I) * COS (A(I)) + .5
350 Y1% = Y% + R(I) * SIN (A(I)) + .5
360 X2% = X% + R(I + 1) * COS (A(I + 1)) + .5
370 Y2% = Y% + R(I + 1) * SIN (A(I + 1)) + .5
380 HPLLOT X1%, Y1% TO X2%, Y2%
390 NEXT I
400 REM STAMPA DEI VALORI
410 FOR K = 1 TO 999: NEXT
420 HOME: TEXT: FOR I = 1 TO NI: PRINT I; TAB( 5)X%(I);
430 PRINT TAB( 10)Y%(I); TAB( 16)R(I); TAB( 29)A(I)
440 NEXT I: END
900 DATA 20, 20, 56, 36, 68, 36, 136, 20, 144, 24
910 DATA 140, 32, 76, 56, 68, 56, 60, 52, 12, 28, 12, 24, 20, 20

```

Figura 3
DATIROTAZ
listato. È un
programma che
permette di tradurre
e controllare
graficamente dati
grafici da coordinate
cartesiane in
coordinate polari
data l'origine degli
assi.

come gestione sia come visualizzazione. In tal caso si visualizza una pagina mentre si forma l'immagine successiva sull'altra e così via. La prima difficoltà dipende dalla velocità della macchina in larga misura e in misura minore dall'abilità del programmatore nel velocizzare l'elaborazione.

Escludiamo la soluzione "slideshow" (che consiste nel creare prima tutte le immagini, nell'immagazzinarle nella memoria di massa, nel caricarle e visualizzarle

una per una), in quanto richiede memorie di massa enormi e tempi di elaborazione legati al tempo di accesso al disco. Ad esempio la pagina grafica dell'Apple II è grande 8K, su un dischetto D.O.S. 3.3, entrano circa 15 slide, e il caricamento di una di queste immagini richiede 3 o 4 secondi.

Ci siamo dilungati un po' nel trattare il tema generale, ed ora arriveremo alle questioni che riguardano più da vicino il pro-

grammatore al quale è affidato il compito di realizzare un programma che crea la successione delle immagini. Infatti il programma deve formare una serie di immagini successive collegate logicamente tra di loro e ciò incide profondamente nella struttura del programma stesso.

Abbiamo realizzato alcuni programmi dimostrativi le cui uscite, trattandosi di figure in movimento, non sono pubblicabili; le abbiamo trasformate in uscite su plotter, per dare più compiutamente il senso del movimento. I primi due listati riguardano il tracciamento di un segmento che ruota attorno al centro dello schermo.

Il programma LINEAUNO (listato in fig. 1), crea la animazione disegnando la linea, cancellandola subito dopo per poi disegnarla di nuovo nella nuova posizione. L'effetto, pur trattandosi di disegno elementare, per il quale la istruzione del disegno è unica (riga 50), non è soddisfacente. Il movimento appare a scatti e se pure sia quasi istantanea si nota la formazione dell'immagine.

L'altro programma LINEADUE realizza lo stesso movimento utilizzando la tecnica del passaggio fra le due pagine grafiche dello Apple II. Il risultato è migliore per il fatto che il passaggio fra le due pagine è istantaneo e quindi è istantanea anche la visualizzazione dell'immagine già formata e già posta correttamente nella nuova posizione (listato in fig. 2).

Con l'istruzione di riga 130 vengono ini-

```

100 HOME: VTAB (22): PRINT "ATTENDERE"
110 NI = 11: DIM R(NI), A(NI), C%(60, 2), P%(60, NI, 2)
120 REM TRIAETTORIA
130 FOR I = 0 TO 60
140 C%(I, 1) = I * 30 + 100
150 C%(I, 2) = (I - 30) ^ 2 + 100: NEXT I
160 FOR L = 0 TO NI: READ R(L), A(L): NEXT L
170 REM PRIMA FASE
180 GOSUB 310: FOR I = 0 TO 60: FOR L = 0 TO NI
190 P%(I, L, 1) = C%(I, 1) + R(L) * COS (A(L)) + I / 4)
200 P%(I, L, 2) = C%(I, 2) + R(L) * SIN (A(L)) + I / 4)
210 NEXT L: GOSUB 320: NEXT I
220 REM SECONDA FASE
230 GOSUB 310: FOR I = 0 TO 60: GOSUB 320: NEXT I
240 REM TERZA FASE
250 GOSUB 310: I = 0
260 POKE 230, 64: CALL 62450
270 GOSUB 300: POKE 49237, 0
280 POKE 230, 32: CALL 62450
290 GOSUB 300: POKE 49236, 0: GOTO 260
300 I = I + 1: GOSUB 320: RETURN
310 HGR2: HCOLOR= 3: RETURN
320 REM ROUTINE DISEGNO
330 FOR L = 0 TO NI - 1
340 X1% = P%(I, L, 1): Y1% = P%(I, L, 2)
350 X2% = P%(I, L + 1, 1): Y2% = P%(I, L + 1, 2)
360 HPLLOT X1% / 7, Y1% / 7 TO X2% / 7, Y2% / 7
370 NEXT L: RETURN
380 DATA 59, 4, 3, 485
390 DATA 20, 4, 3, 339
400 DATA 8, 9, 3, 605
410 DATA 63, 2, -, 322
420 DATA 69, 8, -, 231
430 DATA 64, 5, -, 124
440 DATA 16, 1, 571
450 DATA 17, 9, 2, 034
460 DATA 20, 2, 499
470 DATA 65, 1, 3, 327
480 DATA 66, 3, 386
490 DATA 59, 5, 3, 485

```

Figura 5 - PROGRAMMA BOOMERANG - listato. Viene visualizzato sul monitor grafico Apple II il moto di un boomerang. Prima tutte le posizioni intermedie, poi le due animazioni nelle due modalità presentate.

```

100 N = 20: DIM X%(N), Y%(N), Z%(N), A%(N), B%(N)
110 FOR I = 1 TO N: READ X%(I), Y%(I), Z%(I): NEXT I
120 REM CARICAMENTO DATI
130 DA = 140, 5: DB = 95, 5: HCOLOR= 3
140 REM PAGINAZIONE
150 K = - 1: HGR2: HCOLOR= 3
160 POKE 230, 64: CALL 62450
170 GOSUB 200: POKE 49237, 0
180 POKE 230, 32: CALL 62450
190 GOSUB 200: POKE 49236, 0: GOTO 160
200 REM DISEGNO
210 K = K + 1
220 A = K * 1: B = K * 2: D1 = 500 + K * 10: D2 = 1000
230 CA = COS (A): CB = COS (B): SA = SIN (A): SB = SIN (B)
240 FOR I = 1 TO N
250 X1 = X%(I) * CA + Z%(I) * SA
260 Y1 = Y%(I)
270 Z1 = - X%(I) * SA + Z%(I) * CA
280 X2 = X1
290 Y2 = Y1 * CB + Z1 * SB
300 Z2 = - Y1 * SB + Z1 * CB
310 AX%(I) = (D1 * X2) / (D2 - Z2) + DA
320 BX%(I) = (D1 * Y2) / (D2 - Z2) + DB: NEXT I
330 FOR I = 1 TO 13
340 HPLLOT AX%(I), BX%(I) TO AX%(I + 1), BX%(I + 1)
350 NEXT I
360 FOR I = 15 TO N - 1
370 HPLLOT AX%(I), BX%(I) TO AX%(I + 1), BX%(I + 1)
380 NEXT I: RETURN
390 DATA -100, 75, -50, 100, 75, -50
400 DATA 100, 75, 50, -100, 75, 50
410 DATA -100, 75, -50, -100, -25, -50
420 DATA 100, -25, -50, 100, -25, 50
430 DATA -100, -25, 50, -100, -25, -50
440 DATA -50, -75, 0, 50, -75, 0
450 DATA 100, -25, -50, 100, 75, -50
460 DATA -100, 75, 50, -100, -25, 50
470 DATA -50, -75, 0, 50, -75, 0
480 DATA 100, -25, 50, 100, 75, 50

```

Figura 7 - PROGRAMMA TRIDIMENSIONALE - listato. Lo stesso procedimento di animazione è stato applicato ad un oggetto tridimensionale che ruota attorno a due assi cartesiani e trasla lungo uno di essi.

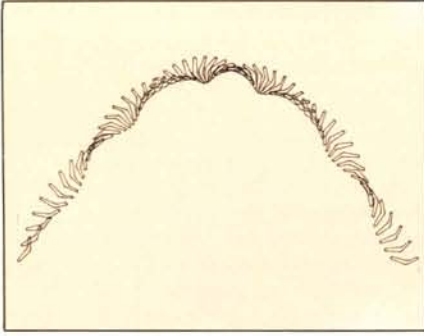


Figura 6 - PROGRAMMA BOOMERANG - output su plotter. Non potendo pubblicare una animazione, abbiamo realizzato una uscita su plotter delle varie posizioni assunte dall'oggetto in movimento.

zializzati il Flag S, il colore e viene pulita la pagina grafica HGR2.

L'andamento del programma è ciclico ed è regolato dal contatore K. Le fasi successive sono la riga 150, cancellazione dell'immagine nella prima pagina; la riga 160, formazione della nuova nella posizione successiva e sua visualizzazione; la 170 cancellazione dell'immagine nella pagina nascosta; la riga 180, formazione nella posizione successiva e nella pagina nascosta (che a questo punto è la prima) e sua visualizzazione. Poi si ricomincia da capo (riga 140) dopo aver incrementato opportunamente il contatore K.

Come si può notare nel nostro caso la riga 190 rappresenta l'intera routine di plottaggio. Questo perché si tratta di un disegno molto semplice. Ma anche per un disegno complesso (e lo vedremo) la struttura del programma non cambia, la funzione di plot anche se molto estesa può essere inserita tutta in una routine, richiamata volta per volta dal gioco delle pagine.

L'unica difficoltà sta nel trattare con attenzione il gioco del contatore K che deve far assumere all'oggetto le varie posizioni giuste a seconda che venga disegnato o cancellato. La funzione ONERR GOTO gestisce il valore del flag S e dell'incremento K, permettendo al segmento di "ritornare indietro" ogni volta che supera i valori canonici 0 e 279.

Boomerang

Come applicazione della teoria svolta e come implementazione del programma precedentemente illustrato abbiamo realizzato altri due programmi di animazione. Uno che mostra il moto bidimensionale di un boomerang, immaginato su un piano, ed un altro che mostra un solido che si muove su una traiettoria nello spazio.

Prima di descrivere i due nuovi programmi ne illustriamo uno di utilità, che ci è servito per trattare i dati grafici del boomerang. È un programma di conversione che serve per facilitare l'uso di programmi grafici in cui ci siano funzioni di rotazione. Traduce i dati cartesiani di un disegno tridimensionale in dati polari rispetto ad un riferimento di cui sia data l'origine, e di cui l'orientamento dell'asse principale sia

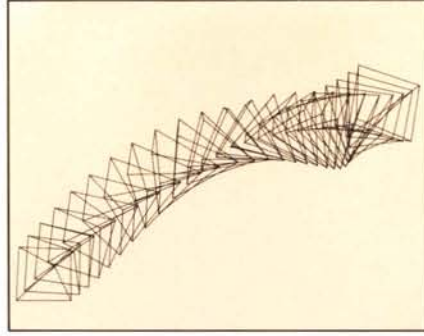


Figura 8 - PROGRAMMA TRIDIMENSIONALE - output su plotter. Anche in questo caso il movimento del solido è stato riportato su carta, modificando l'uscita da video a plotter.

quello dell'asse X. Il programma ha anche due uscite grafiche di controllo per i dati in coordinate cartesiane (righe 150-190) e in coordinate polari (righe 300-390). Per un uso pratico del programma tali routine possono essere omesse.

In sostanza il programma va usato immettendo in riga 110 il numero delle coppie di coordinate. Nei DATA vanno immessi i loro valori, che vengono caricati nei vettori X%, Y% di riga 120. Vanno poi immesse le coordinate dell'origine (riga 210) e a questo punto il programma calcola R(I), raggio e A(I) angolo per ogni punto operando le opportune correzioni dell'arcotangente con le solite modalità (righe 220-280). Alla fine viene stampata (vedi fig. 4) la tabellina di traduzione.

L'utilità di questo programma sta nel fatto che per i programmi grafici di rotazione è indispensabile lavorare in coordinate polari, mentre l'immissione dei dati base è sempre più facile in coordinate cartesiane.

Il nostro Boomerang (listato in fig. 5) è quindi formato da una spezzata chiusa di 12 punti, ovvero da 11 segmenti. Le coordinate polari rispetto al baricentro sono caricate nei vettori R(I), A(I) in riga 160.

Il moto sul piano è un moto composto dal moto parabolico del baricentro e dalla contemporanea rotazione dell'oggetto attorno ad esso.

Il moto del baricentro viene precalcolato, vengono cioè individuate 61 sue posizioni differenti lungo la parabola, le cui coordinate sono caricate nella matrice C% (I, L). Cioè nella posizione I-esima il baricentro del boomerang assume le coordinate C%(I, 1), C%(I, 2) rispettivamente sulla X e sulla Y.

Il programma si divide in tre parti.

Nella prima vengono calcolate tutte le 12 posizioni delle coordinate dei punti del boomerang in ognuna delle 61 posizioni del baricentro. Il calcolo (righe 170-210) comporta l'esecuzione di due loop (I da 0 a 60 e L da 0 a NI) e produce il caricamento della matrice P%(60, NI, 2) con i risultati del calcolo della composizione dei due moti. Il moto di rotazione attorno al baricentro tra due passi successivi comporta un incremento dell'angolo di circa 15 gradi.

Durante il calcolo vengono visualizzate

le varie immagini. La routine di plot, che vale per tutte le varie fasi del programma, è nelle righe da 320 a 370.

Dopo questa fase ce n'è un'altra che visualizza in rapida successione le immagini, ormai immagazzinate in una matrice, semplicemente accendendo volta per volta la pagina grafica (righe 220-230).

Infine c'è la terza fase, le cui istruzioni, se notate, sono identiche a quelle utilizzate nel programma LINEADUE (righe 240-290). La routine di plot, come abbiamo detto, è sempre la stessa, ed è nelle righe da 320 a 370. Quest'ultima fase finisce con un errore, in quanto non ha, come si può notare una fine naturale. In realtà potevamo usare un loop per l'indice I in modo da considerare correttamente gli incrementi necessari alla paginazione. Oppure avremmo potuto mettere un IF I=60 nella routine di plottaggio, con il risultato di rallentare il programma ma con l'inutile vantaggio di avere una fine naturale nello stesso momento in cui si ha la fine accidentale. L'uscita su plotter è in figura 6.

Tridimensionale

Se il procedimento di paginazione funziona per il bidimensionale, certamente funzionerà anche per il tridimensionale. L'unica difficoltà, al solito, è il gioco del contatore che deve garantire ad un tempo il corretto utilizzo della paginazione e a cui deve essere legata la routine che genera il movimento. Nel programma dimostrativo di figura 7 guardate a quante cose serve il K. Nella riga 210 per la paginazione, nella riga 220 per definire la nuova posizione del solido ruotato rispetto all'asse Y (angolo A), rispetto all'asse X (angolo B) e la traslazione rispetto all'asse Z. Programmi di grafica tridimensionale li abbiamo pubblicati più volte, riassumiamone i passaggi.

Le coordinate dei vertici del solido da visualizzare sono immagazzinate nei vettori X%, Y%, Z% (riga 110). Ogni nuova posizione viene definita dai parametri A, B angoli di rotazione attorno a Y e X, e D1, D2 distanza lungo l'asse Z tra osservatore e origine e schermo e tra osservatore ed origine del riferimento (righe 220-230).

Vengono poi calcolate le posizioni intermedie X1, Y1, Z1 e X2, Y2, Z2 e con queste le posizioni sullo schermo bidimensionale di ogni singolo punto (righe 240-300).

Le coordinate trovate sono caricate nei vettori A%, B% (righe 310-320) che sono poi utilizzati per la routine di visualizzazione (righe 330-380).

Utilizzando i dati pubblicati appare una casetta, formata da due spezzate rispettivamente di 12 e di 5 segmenti. Da qui i due loop per la visualizzazione.

Il disegno su plotter (fig. 8) lo abbiamo leggermente modificato. Si tratta infatti di una piramide a base quadrata, invece che di una casetta e di una traiettoria nello spazio, in quanto la sovrapposizione delle varie posizioni avrebbe generato un disegno incomprensibile.