L'Italia dallo Space Shuttle..

Chi, vedendo alla televisione a colori le immagini del volo dello Space-Shuttle, non ha pensato: "mi piacerebbe essere lassù, magari solo per vedere fuori dal finestrino". Non potendo realizzare questo desiderio, ci si può accontentare di "simulare" il finestrino dell'aereo spaziale con il proprio Microcomputer ...

Vogliamo, in questo articolo, descrivere la realizzazione di un programma di Computer Grafica Tridimensionale, seguendo passo passo il suo sviluppo dalla nascita dell'idea fino alla stesura ed alla esecuzione del programma finale.

L'idea, ovvero l'argomento del programma, è qualsiasi, in quanto ci interessa più seguire tutto il suo sviluppo logico che realizzare un programma per uso specifico. Quindi, trovata una metodologia, si potrà applicare ad altre idee fantasiose, come è quella del finestrino dello Space-Shuttle, oppure, molto più opportunamente, a problemi concreti di reale interesse.

L'idea che vogliamo sviluppare è quindi quella di "simulare" il finestrino dello Space-Shuttle dal quale gli astronauti in orbita vedono scorrere sotto di sé la superficie terrestre.

Il problema presenta una certa complessità in quanto si tratta di eseguire e soprattutto di tenere sotto controllo una serie di trasformazioni di coordinate spaziali e piane, ognuna delle quali è legata a numerose variabili; allora, per raggiungere l'obiettivo finale, realizzeremo dei programmi intermedi, con i quali affronteremo e risolveremo isolatamente i singoli aspetti del programma finale. Inoltre ci aiuteremo con degli "schizzi", indispensabili per mettere a fuoco le idee prima di passare all'azione.

Realizziamo il programma passo passo

Facciamo un rapido esame dei passi successivi seguiti nella realizzazione del programma. Tali passi consistono sia nell'esaminare e risolvere i problemi aritmetici e geometrici che ci si presentano sia nel realizzare i cosiddetti programmi intermedi per testare le soluzioni dei singoli problemi particolari.

1) Esame e soluzione analitica del problema della visuale.

Il problema viene enormemente semplificato ponendo tutti gli elementi sul piano XY e ponendo schermo S e orientamento dell'oggetto H da visualizzare perpendicolare all'asse X e ponendo l'osservatore sull'origine O. Con tali semplificazioni è facile trovare la formula che lega la posizione assunta dall'oggetto sullo schermo alla distanza tra osservatore ed oggetto stesso, e alla distanza, che considereremo fissa, tra osservatore e schermo D (vedi fig. 1).

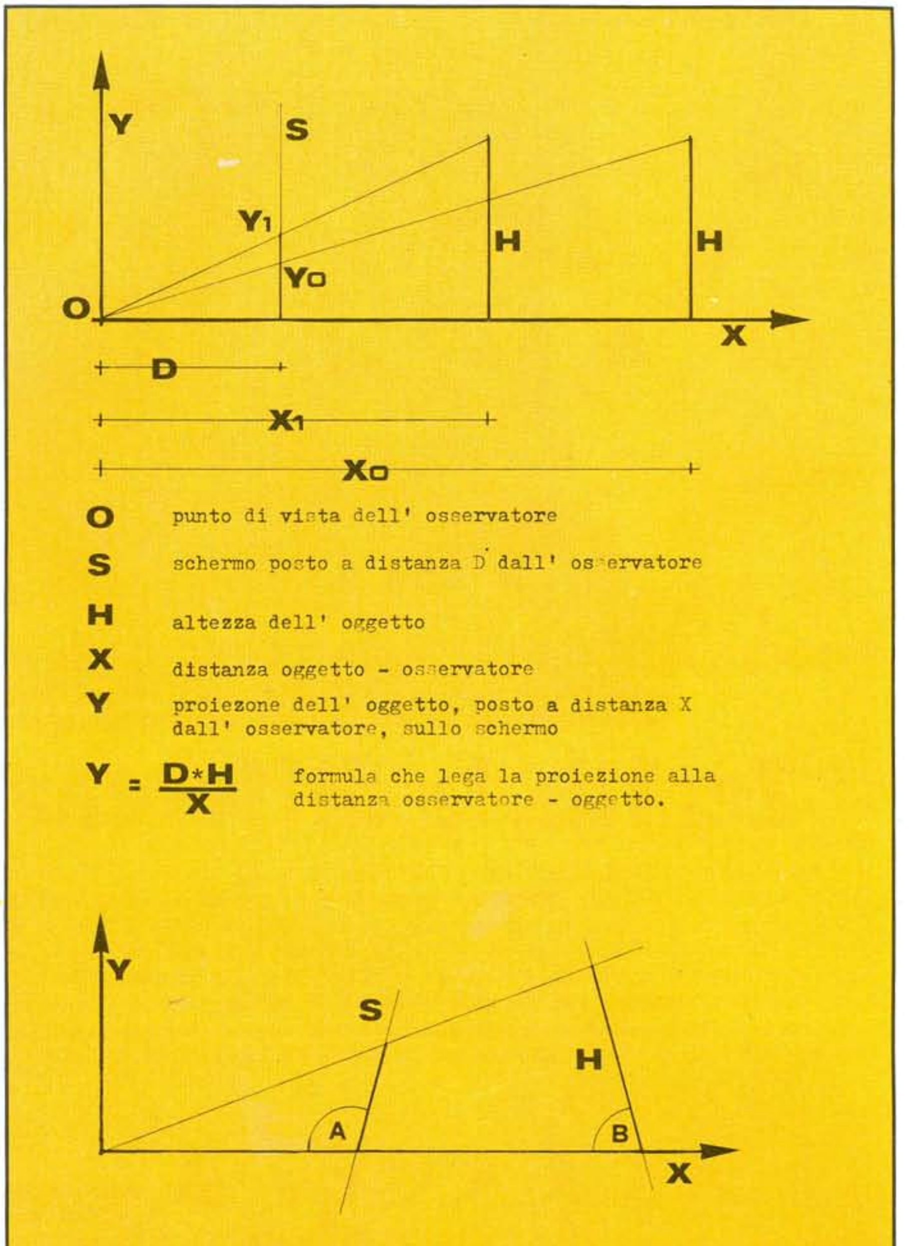


Figura 1 — Rappresentazione bidimensionale del problema della visuale.

In tutti i programmi realizzati per questo articolo sono state inserite queste semplificazioni che però non ne alterano la logica, né il risultato. Di tali semplificazioni parleremo più diffusamente in seguito.

Vediamo comunque nella fig. 1 la generalizzazione del problema della visuale. Gli elementi in più rispetto a quelli dello schizzo precedente sono i due angoli α e β. La $Y = Y(X)$ che individuava totalmente il problema semplificato diventa $Y = Y(X, \alpha\ \beta)$.

2) Realizzazione del programma "Quadrato" (figg. 2 e 3).

In tale programma viene applicata ad un caso semplice la formula precedentemente trovata. L'oggetto da visualizzare è un quadrato posto sul piano XY e immobile, l'osservatore si muove nello spazio con una traiettoria qualsiasi, portandosi dietro solidamente lo schermo.

3) Traduzione del quadrato in parallelepipedo (figg. 4, 5 e 6).

Il programma precedente è già un programma tridimensionale, in quanto il movimento dell'osservatore si svolge nello spazio, solo che l'oggetto da visualizzare è bidimensionale. Ora invece l'oggetto è tridimensionale. Anche in questo programma inseriremo numerose semplificazioni individuabili dallo schizzo di fig. 4.

4) Traduzione da coordinate cartesiane sul piano a coordinate polari nello spazio e coordinate cartesiane nello spazio.

Risolti, con i programmi precedenti, tutti i problemi di visualizzazione sullo schermo bidimensionale del solido tridimensionale (espresso in coordinate cartesiane XYZ), affrontiamo il passo finale, quello di sostituire il solido nello spazio con la superficie terrestre, della quale dobbiamo vedere una certa porzione, delimitata dall'orizzonte. Anche qui invece di vedere subito l'Italia, faremo una semplificazione per testare la correttezza delle soluzioni geometriche trovate e ci accontenteremo di vedere ancora un quadrato.

5) Ricerca delle formule di traduzione.

Per cercare le formule ricorriamo al solito schizzo bidimensionale chiarificatore del problema.

Supponiamo di voler rilevare i dati da visualizzare, ovvero i punti che individuano il profilo dello "Stivale", da una cartina geografica, specificando per ciascun punto una coppia di coordinate cartesiane.

```
]LIST
100  REM  CARICAMENTO DATI GRAFICI
110  FOR I = 1 TO 5: READ X%(I),Y%(I): NEXT
120  DATA    100,100,150,100,150,150,100,150,100,100
130  REM  INIZIALIZZAZIONE COSTANTI
140 HS = 10:HO = 20:XO = 10:YO = 50
180  REM  INIZIALIZZAZIONE MODO GRAFICO
190  HGR2 : HCOLOR= 3
200  REM  LOOP PRINCIPALE
210  FOR HO = 110 TO 5 STEP  - 2
220 YO = 120 - HO / 2
300  REM  CALCOLO COORDINATE SCHERMO
310  FOR I = 1 TO 5
320 XT%(I) = X%(I) - XO:YT%(I) = Y%(I) - YO
330 XS%(I) = HS * XT%(I) / HO + XO
340 YS%(I) = HS * YT%(I) / HO + YO
350  NEXT
500  REM  VISUALIZZAZIONE SINGOLO QUADRATO
510  FOR I = 1 TO 4
520  HPLOT XS%(I),YS%(I) TO XS%(I + 1),YS%(I + 1): NEXT

530  NEXT : END
1010  PRINT X%(I); TAB( 8)Y%(I); TAB( 20)XS%(I); TAB(
     28)YS%(I)
1020  REM  SUBROUTINE CALCOLO VALORI NUMERICI
```

Figura 2 — Listing del programma "QUADRATO"; l'osservatore O si allontana progressivamente dall'oggetto (un quadrato) posto sul piano XY.

Dobbiamo dunque prendere i punti P(XC,YC), posti sulla piantina giacente sul piano C, portarli sulla superficie sferica individuandoli prima in coordinate polari P(R, α) e poi in coordinate cartesiane P(XP,YP,ZP) e infine trasferirli sullo schermo S, P(XS,YS). Le formule trovate sono abbastanza elementari e sono quelle individuate nello schizzo di fig. 7.

6) Prova generale del programma con un caso semplice.

Intendiamo dunque prendere un quadrato (ovvero il profilo dell'Italia), poggiarlo su una superficie sferica (ovvero sulla superficie terrestre) trovare le coordinate polari dei suoi punti (ovvero longitudine e latitudine di ciascun punto che individua il profilo) e trovare le coordinate spaziali XYZ. A questo punto, se tutto è andato bene, non occorre altro che collegare il programma di traduzione a quello di visualizzazione tridimensionale precedentemente realizzato per il parallelepipedo che aveva come dati base, appunto, le coordinate XYZ di ciascun punto da visualizzare.

7) Realizzazione del programma definitivo.

Se la prova generale è riuscita si possono sostituire con relativa rapidità i dati utilizzati nel caso semplificato con i dati più complessi che ci interessano, ad esempio i dati relativi al profilo della penisola italiana.

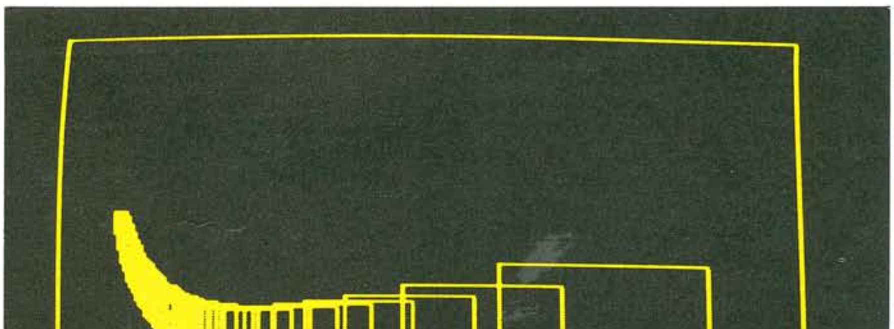


Figura 3 — Output del programma "QUADRATO"; le varie immagini sono sovrapposte per rappresentare il movimento dell'osservatore rispetto all'oggetto.

I programmi intermedi e le loro semplificazioni

Dopo aver schematicamente indicato i passi dello sviluppo della realizzazione, vedremo nei dettagli i programmi intermedi realizzati. Ma prima sarà bene fare una breve digressione sulle semplificazioni apportate a questi programmi.

Come noto i gradi di libertà di un solido nello spazio sono sei. Per chi non sapesse cosa sono i gradi di libertà: un punto su di una retta ha un solo grado di libertà, quindi per individuare un punto su di una retta è sufficiente una sola coordinata, per individuare un punto su di un piano sono sufficienti due coordinate, ovvero il punto ha due gradi di libertà. Il solido nello spazio ha sei gradi di libertà, tre relativi alle traslazioni possibili lungo gli assi XYZ e tre relativi alle rotazioni possibili di tutto il solido rispetto ai tre assi XYZ.

Il problema tridimensionale generalizzato della prospettiva deve dunque tener conto di tutti i gradi di libertà dell'oggetto da visualizzare, dello schermo su cui visualizzare e dell'osservatore che osserva l'oggetto sullo schermo.

Se realizziamo un programma di Computer Grafica Tridimensionale, oltre alla trattazione geometrica del problema, in sé, dobbiamo tener conto delle caratteristiche geometriche del sistema con il quale operiamo.

Nei sistemi Computer Grafici specializzati è possibile avere molte delle funzioni che interessano l'intero processo di trasformazione delle coordinate direttamente nel software di base, ad esempio vi sono i comandi in esecuzione diretta di scaling, traslazione e rotazione dell'immagine su video.

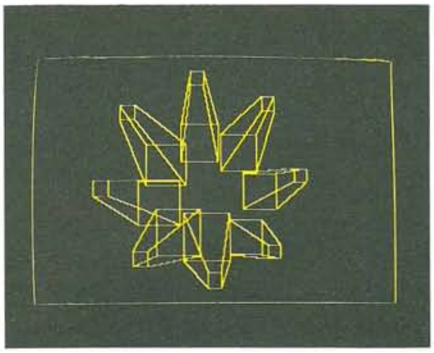


Figura 6 — Output del programma "PARALLELEPIPEDO";
l'oggetto da visualizzare è fermo, si muove l'osservatore portandosi dietro lo schermo.

Noi invece lavoriamo con un microcomputer sprovvisto di tali istruzioni (stiamo usando un Apple II, e utilizziamo il BASIC esteso Applesoft). Dobbiamo realizzare per conto nostro il software e quindi dobbiamo ricorrere a massicce semplificazioni perché altrimenti i tempi di elaborazione diventerebbero inaccettabili.

Tutti i programmi realizzati per l'articolo sono dunque semplificati e l'entità di tali semplificazioni è illustrata nel testo.

Un'altra necessità da rispettare in fase di preparazione e di debugging del programma è quella di visualizzare i risultati intermedi, ad esempio di visualizzare i valori assunti dalle varie coordinate calcolate dal programma; questo allo scopo di controllarne la correttezza passo passo ed eventualmente di individuare errori di programmazione (vedi fig. 8).

Chi lavora con l'Apple II quando studia la visualizzazione grafica di una qualsiasi funzione Y = Y(X), deve tener presente che le coordinate di riferimento del sistema alta risoluzione del computer non ammettono valori negativi, né valori frazionari e inoltre che l'asse Y è orientato in maniera diversa rispetto al solito. Ad esempio lo studio grafico della funzione Y = SIN(X) richiederà necessariamente una inversione dell'asse Y, una traslazione di ambedue gli assi, e il frazionamento delle coordinate. La funzione diventerà, per essere visibile sul monitor, Y = YC - SIN(X/S - XC)*SC, con YC, XC, S, SC opportune costanti.

Il programma corrispondente sarà quello di fig. 8bis.

Programma Quadrato

Il primo programma intermedio è quello che consente la rapida applicazione delle formule trovate con la fig. 1. Il listing è in fig. 2.

I dati del programma sono le coordinate del quadrato, supposto giacente ed immobile sul piano XY, la distanza HS tra lo osservatore e lo schermo S, supposta anche questa fissa. L'osservatore, posto nel punto di coordinata X0,Y0,Z0, è in movimento. Per simulare il suo movimento nello spazio basterà trovare una legge di variazione per queste tre coordinate. La legge inserita nel programma è realizzata con un loop sulla Z e corrispondentemente viene

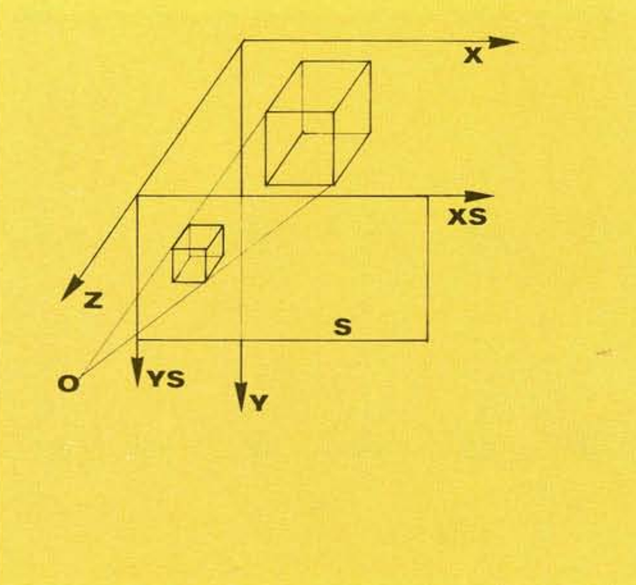


Figura 4 — Schizzo illustrativo del programma "PARALLELEPIPEDO".

```
]LIST
90  HGR2 : HCOLOR= 3
100  REM   INIZIALIZZAZIONE COSTANTI
110 HS = 10:RO = 100
120 XC = 125:YC = 90:ZC = 150
200  REM    CARICAMENTO DATI GRAFICI
210  DATA   100,100,100,100,150,100,150,150,100,150,100,100
220  DATA    100,100,130,100,150,130,150,150,130,150,100,130
230  FOR I = 1 TO 8: READ X%(I),Y%(I),Z%(I): NEXT
300  REM   LOOP PRINCIPALE DEL MOVIMENTO
310  FOR A = 6.4 TO 0 STEP  - .8
320 XO = RO *  SIN (A) + XC
330 YO = RO *  COS (A) + YC
340 ZO = ZC - A
400  REM   LOOP DI CALCOLO COORDINATE SCHERMO
410  FOR I = 1 TO 8
420 XT%(I) = X%(I) - XO
430 YT%(I) = Y%(I) - YO
440 ZT%(I) = Z%(I)
450 XS%(I) = XO + HS * XT%(I) / (ZO - ZT%(I))
460 YS%(I) = YO + HS * YT%(I) / (ZO - ZT%(I))
470  NEXT
500  REM  VISUALIZZAZIONE SINGOLA IMMAGINE
510  FOR I = 1 TO 3
520  HPLOT XS%(I),YS%(I) TO XS%(I + 1),YS%(I + 1)
530  HPLOT XS%(I + 4),YS%(I + 4) TO XS%(I + 5),YS%(I + 5)
540  NEXT
550  FOR I = 1 TO 4
560  HPLOT XS%(I),YS%(I) TO XS%(I + 4),YS%(I + 4)
570  NEXT
580  HPLOT XS%(1),YS%(1) TO XS%(4),YS%(4)
590  HPLOT XS%(5),YS%(5) TO XS%(8),YS%(8)
595  GOSUB 1000: REM  RIGA DA ELIMINARE
600  NEXT A: END
1000  REM  STAMPA VALORI COORDINATE
1010  PRINT : PRINT
1020  PRINT "N    XT    YT    ZT     XS    YS       "
1030  FOR I = 1 TO 8
1040  PRINT I; TAB( 6)XT%(I); TAB( 12)YT%(I); TAB( 18)ZT%(I);
1050  PRINT  TAB( 26)XS%(I); TAB( 32)YS%(I)
1060  NEXT I: RETURN
```

Figura 5 — Listing del programma "PARALLELEPIPEDO";
le grandezze in gioco, trattandosi di oggetti tridimensionali, cominciano ad essere parecchie.

fatta variare linearmente la Y0. L'osservatore compie quindi una traiettoria rettilinea nello spazio, si avvicinerà progressivamente al piano del quadrato, variando contemporaneamente la posizione sulla Y (Fig. 3).

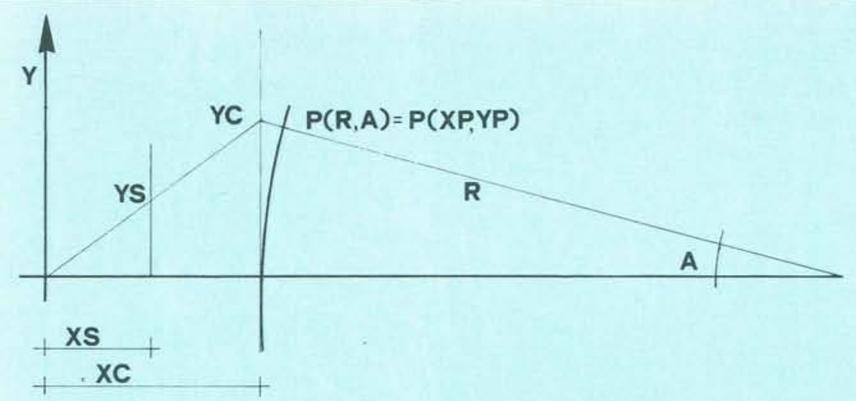
Il programma già così semplificato, permette infinite varianti, poiché infinite sono le leggi che posssono legare le X0,Y0,Z0 tra di loro. Ad esempio si può simulare una traiettoria circolare, una traiettoria parabolica, una traiettoria in cui la legge di variazione sull'asse Z sia la legge di gravità, ecc.

In questo come nei successivi programmi è stata lasciata la subroutine di stampa (righe 1000 e successive) per la visualizzazione dei valori intermedi assunti dalle coordinate calcolate dal programma. La subroutine, che va bypassata nel programma finale, va richiamata con un GOSUB, interno al loop principale, in fase di debug.

Programma Parallelepipedo

Il secondo programma realizzato è sostanzialmente simile al precedente, solo che l'oggetto da visualizzare è tridimensionale e quindi è definito da una serie di punti individuati nello spazio da terne di valori XYZ. Quindi la proiezione dei singoli punti sullo schermo S deve tener conto di questo ulteriore dato.

La trasformazione, con le semplificazioni che ci siamo dati, da bidimensionale in tridimensionale è facilissima in quanto la nuova coordinata Z interviene solamente dove c'era la distanza tra osservatore e oggetto, unico elemento che anche nel caso



XS   distanza osservatore - schermo
XC   distanza osservatore - superficie sferica
P(XC,YC)   punto sulla piantina
P(R,A)   punto sulla superficie sferica in coord.polari
P(XP,YP)   punto sulla superficie sferica in coord.cartes.
P(XS,YS)   punto sullo schermo

$$\frac{XS}{YS} = \frac{XP}{YP} \quad ; \quad YS = \frac{XS*YP}{XP}$$

ma

$$XP = XC + R(1 - COS(A))$$

$$YP = R*SEN(A)$$

allora

$$YS = \frac{XS(XC + R(1 - COS(A)))}{R*SEN(A)}$$

Figura 7 — Schizzo illustrativo del programma "SPACE-SHUTTLE"; la trasformazione in più, rispetto al programma precedente, è da dati polari (longitudine e latitudine) sulla superficie sferica a dati spaziali cartesiani.

```
]
N  - PP -    - PA -    - PR        -     - PS -
   PIANO     ANGOLO    SPAZIO            SCHERMO
1  100 100   19  19    58  58  58        30  30
2  120 100   26  19    61  77  58        28  28
3  140 100   32  19    66  94  58        26  26
4  160 100   38  19    72  111 58        24  24
5  160 120   38  26    73  111 77        31  31
6  160 140   38  32    76  111 94        37  37
7  160 160   38  38    80  111 111       41  41
8  140 160   32  38    76  94  111       43  43
9  120 160   26  38    73  77  111       45  45
10 100 160   19  38    72  58  111       46  46
11 100 140   19  32    66  58  94        42  42
12 100 120   19  26    61  58  77        37  37


]LIST
10  HGR2 :A = 95:B = 15:C = 140:D = 50
20  FOR X = 0 TO 279
30 Y = A -  SIN (X / B - C) * D
40  HPLOT X,Y TO X,95: NEXT
```

Figura 8 — Esempio di output di dati di controllo; se c'è qualche errore logico nei programmi, la visualizzazione dei risultati intermedi aiuta a scoprirlo.

```
]LIST
100  REM  INIZIALIZZAZIONE COSTANTI
110  DIM PP(99,2),PA(99,2),PR(99,3),PS(99,2)
120 R = 300:HV = 50:HS = 40:N = 12
130 P1 = 40:P2 = 40:XS = 50:YS = 0
140 B$ = "                    "
200  REM  RICHIAMO DATI
210  GOSUB 500
220  REM  CALCOLO PASSAGGI INTERMEDI
230  FOR HS = 30 TO 100 STEP 10
240  GOSUB 700: GOSUB 800: GOSUB 900
300  REM  VISUALIZZAZIONE
305  GOSUB 1000: REM  RIGA DA ELIMINARE
310  HGR2 : HCOLOR= 3
320  FOR I = 1 TO N
330  HPLOT PS(I,1),PS(I,2) TO PS(I + 1,1),PS(I + 1,2)
340  NEXT I: NEXT HS
350  END
500  REM   CARICAMENTO COORDINATE SUL PIANO
510  FOR I = 1 TO N + 1: READ PP(I,1): READ PP(I,2): NEXT I
520  DATA 100,100,120,100,140,100,160,100,160,120,160,140,160
530  DATA 160,140,160,120,160,100,160,100,140,100,120,100,100
540  RETURN
700  REM   TRADUZIONE IN COORDINATE POLARI
710  FOR I = 1 TO N + 1
720 PA(I,1) =  ATN ((PP(I,1) - P1) / R)
730 PA(I,2) =  ATN ((PP(I,2) - P2) / R)
740  NEXT I: RETURN
800  REM   TRADUZIONE IN COORDINATE SPAZIALI
810  FOR I = 1 TO N + 1
820 PR(I,1) = HV + R *  SQR ((1 -  COS (PA(I,1))) ^ 2 + ((1 -
     COS (PA(I,2))) ^ 2))
830 PR(I,2) = R *  SIN (PA(I,1))
840 PR(I,3) = R *  SIN (PA(I,2))
850  NEXT I: RETURN
900  REM   TRADUZIONE IN COORDINATE SCHERMO
910  FOR I = 1 TO N + 1
920 PS(I,1) = PR(I,2) * HS / PR(I,1) + XS
930 PS(I,2) = PR(I,3) * HS / PR(I,1) + YS
940  IF PS(I,1) < 0 THEN PS(I,1) = 0
950  IF PS(I,1) > 279 THEN PS(I,1) = 279
960  IF PS(I,2) < 0 THEN PS(I,2) = 0
970  IF PS(I,2) > 191 THEN PS(I,2) = 191
980  NEXT I: RETURN
1000  REM   VISUALIZZAZIONE VALORI PASSAGGI INTERMEDI
1010  PRINT "N  -  PP -  - PA -  -  PR     -     - PS -   PIANO
      ANGOLO  SPAZIO        SCHERMO"
1020  FOR I = 1 TO N
1030  PRINT  STR$ (I); TAB( 4)(PP(I,1)); TAB( 8)(PP(I,2)); TAB(
     13) INT (PA(I,1) * 100); TAB( 17) INT (PA(I,2) * 100);
1040  PRINT  TAB( 21) INT (PR(I,1)); TAB( 25) INT (PR(I,2)); TAB(
     29) INT (PR(I,3)); TAB( 34) INT (PS(I,2)); TAB( 38) INT (
     PS(I,2))
1050  NEXT I
```

Figura 9 — Listing del programma "VELA"; organizzato e provato il programma sul solito quadrato, si tratta solo di scegliere un soggetto fotogenico.

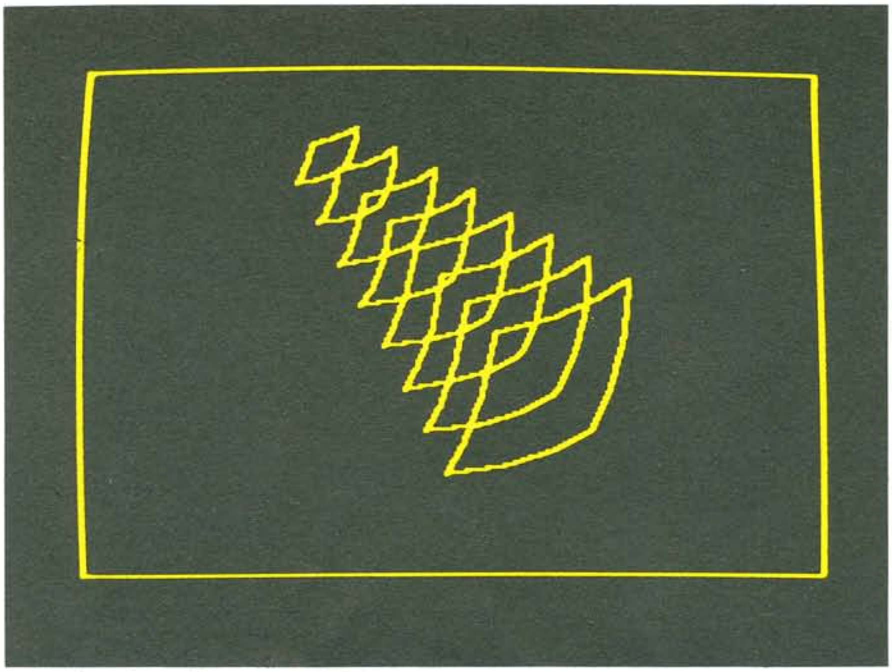


Figura 10 — Output del programma "VELA"; il quadrato, poggiato su una superficie sferica, non può più essere individuato solo con i quattro vertici.

precedente era interessato da una coordinata sull'asse Z.

Il programma di visualizzazione deve poi conoscere come sono collegati tra di loro gli otto vertici del parallelepipedo.

Per quanto riguarda lo spostamento dell'osservatore nello spazio, qui è stata scelta una traiettoria elicoidale cilindrica nello spazio, facilissima da realizzare e da comprendere. C'è un loop in A, la XØ e la YØ variano con il seno ed il coseno di A e la ZØ varia linearmente con A. Il listing è in fig. 5. L'output di fig. 6 fa vedere le varie immagini sovrapposte, in realtà il programma rappresenta come in una animazione l'avvicinamento progressivo sulla traiettoria elicoidale, dell'osservatore all'oggetto. L'animazione si realizza semplicemente cancellando l'immagine precedente all'apparire della successiva.

C'è però da dire che si può realizzare, con un programma in BASIC, una animazione solo quando l'oggetto da far muovere è semplice, altrimenti la lunghezza dell'elaborazione, necessaria per visualizzare l'oggetto nelle varie posizioni rende il movimento stesso non continuo ma a scatti.

Programma Vela

Nel terzo ed ultimo programma realizzato per l'articolo vengono utilizzate le formule trovate nello schizzo di fig. 7, per la traduzione dalle coordinate cartesiane sul piano in coordinate polari nello spazio e poi in coordinate cartesiane nello spazio.

Dapprima vengono caricati i dati del quadrato da visualizzare (righe 500 e seguenti). Il quadrato è individuato da dodici punti (quattro per lato), in quanto dovendolo depositare sulla sfera, perderà il suo aspetto "quadrangolare" per assumere quello di una vela.

Successivamente le coordinate del quadrato sono tradotte in coordinate polari spaziali, ovvero longitudine e latitudine PA(I,1) e PA(I,2) e poi in coordinate spaziali cartesiane PR(I,1), PR(I,2), PR(I,3), tramite le formule geometriche, e che utilizzano un po' la trigonometria, trovate con lo schizzo di fig. 7.

Infine riutilizzeremo le formule del programma "parallelepipedo", per passare da coordinate cartesiane spaziali a coordinate-schermo. Questa routine deve essere dotata anche di controllo di compatibilità tra i valori trovati e valori accettati dall'Apple II, per i grafici ad alta risoluzione.

Il listing è in fig. 9 e l'output in fig. 10.

Per passare dal quadrato a qualcosa di più complesso basta sostituire le righe di caricamento dati con altre istruzioni di caricamento: READ DATA, lettura di files con dati grafici caricati con altri programmi specifici, routine di caricamento da digitizer, se lo abbiamo a disposizione. Il profilo dell'Italia da noi utilizzato è composto da circa 150 punti. (Vedi figg. 11 e 12).

Francesco Petroni

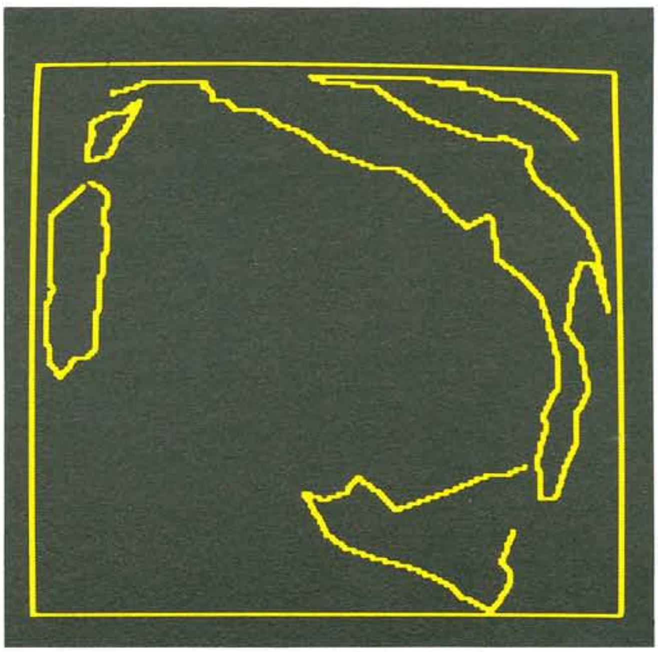
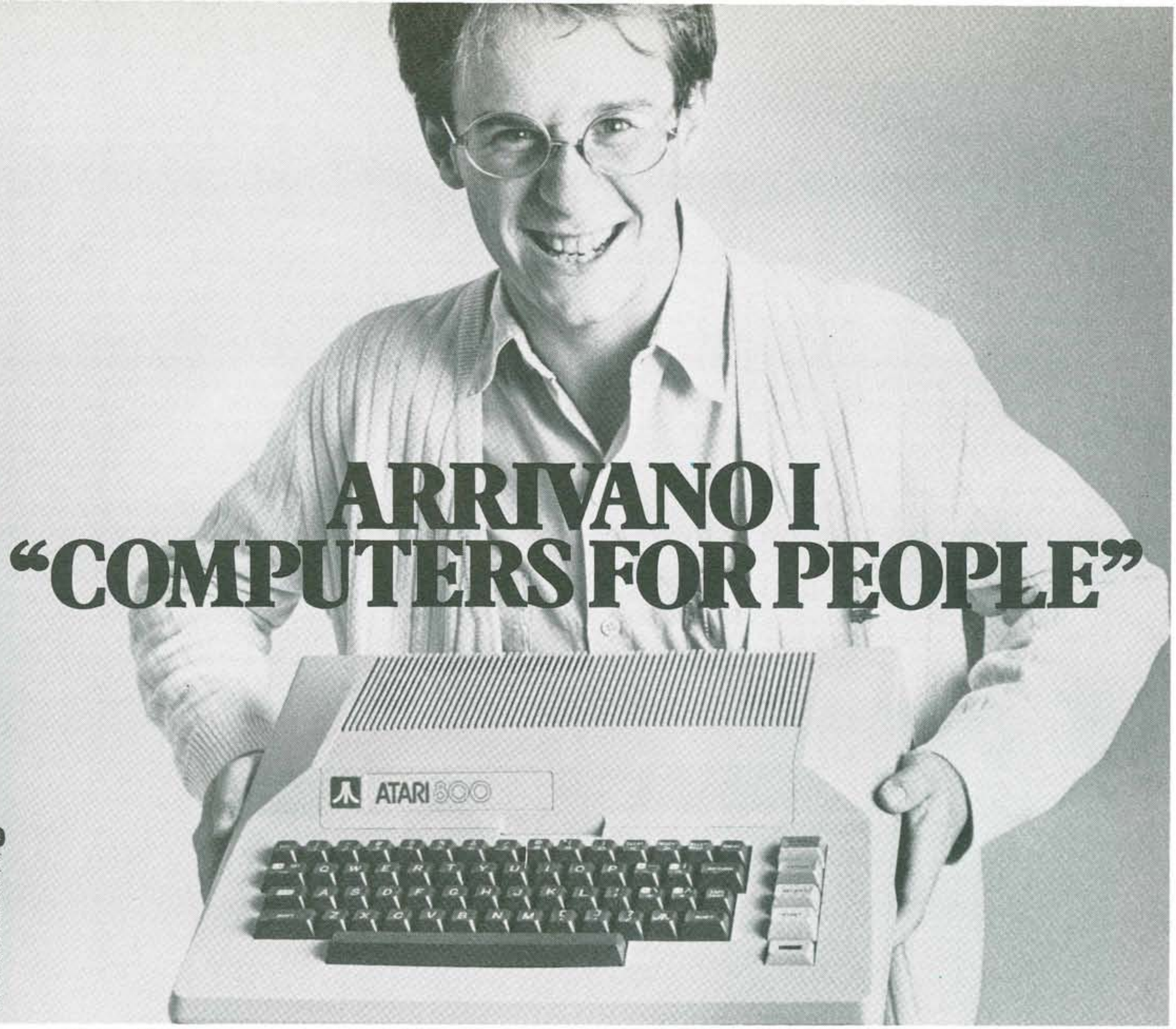


Figura 11 — Output del programma "SPACE-SHUTTLE";

Figura 12 — Il soggetto prescelto è il profilo dell'Italia, visto da due punti differenti. Purtroppo l'elaborazione di tutte le trasformazioni è troppo lenta per raggiungere l'effetto "animazione".



81 XAD 1

A Warner Communications Company

Non più grande di una macchina da scrivere, non più costoso di un hi-fi, Atari è il risultato più avanzato della tecnologia informatica americana.

Collegate Atari al televisore di casa ed è tutto: Atari è già pronto a funzionare: facilmente, docilmente, velocemente.

Atari sa fare per voi (che siete un ingegnere, un medico, un negoziante, un artigiano...) tante cose: archivi, schedari, agenda personale, gestione di magazzino, fatturazione e bolle, ecc.

Un discorso a parte, poi, merita la scuola: Atari è un aiuto prezioso sia per gli studenti che per gli insegnanti, in ogni ordine e tipo di scuola.

Atari è già entrato come moderno sistema didattico nelle aule d'America e di altri paesi: Atari è lo strumento migliore per preparare i giovani a quella "civiltà del computer" che certamente li aspetta.

Atari può essere usato anche per tutti i tipi di video-games, dal basket agli scacchi. Insomma Atari scrive, disegna grafici, disegna figure, suona e compone musica, calcola, prevede, ricorda, consiglia soluzioni. E tante prestazioni ancora che scoprirete usandolo.

E se le vostre esigenze aumentano, aumenta anche lui: può essere affiancato da più accessori (stampanti, unità-memoria esterne, accoppiatore acustico, telelink e tanti altri).

Potete scegliere il vostro Atari nei due modelli base 400 ed 800. Telefonateci e saremo lieti d'invitarvi a vedere come un Atari è facile da usare, capace, rapido, agile e perchè no, affascinante.

ATARI®
Computers for people.

DISTRIBUTORE ESCLUSIVO PER L'ITALIA
ADVEICO
CONSUMER DIVISION

ADVEICO S.r.l. - 20124 Milano - Via A. Tadino, 22, Tel. 02/2043281 - 43016 S. Pancrazio, Parma - Via Emilia Ovest, 129, Tel. 0521/998841