



di Francesco Petroni

L'eliminazione delle linee nascoste nel disegno dei solidi convessi

Lo spunto per questo articolo è stato tratto dal libro di R.E. Myers *Grafica per Microcomputer* (ed. Masson) nell'edizione italiana, che è quella scritta per l'Apple II. Esiste una versione per il PC IBM, ed. Addison Wesley, ma non è stata ancora tradotta in Italiano.

Nel libro viene presentato un metodo per la determinazione delle linee che risultano nascoste all'occhio dell'osservatore nella rappresentazione grafica bidimensionale di solidi tridimensionali (comunemente il problema delle linee nascoste).

Chi si occupa, sia per interesse professionale che semplicemente per hobby, di computer grafica, sa che uno dei problemi più difficili da risolvere è sicuramente quello delle linee nascoste a causa dell'estrema numerosità delle variabili che entrano in gioco, che rendono pressoché impossibile la realizzazione di una soluzione generalizzata che vada sempre bene, indipendentemente dall'oggetto da rappresentare.

Esistono però metodi semplificati che si adattano a varie categorie di «oggetti». Vi è il metodo della *Rappresentazione per Punti*, particolarmente adatto alla visualizzazione su video di curve spaziali, oppure il metodo della *Copertura* che consiste nel nascondere le linee coprendole con le porzioni di disegno che le sono logicamente davanti.

Il metodo ripreso da Myers è applicabile a solidi geometrici, è abbastanza semplice da comprendere, ma ha un campo di applicazione molto ristretto, quello dei solidi convessi.

I solidi convessi sono quelli in cui il piano su cui giace ciascuna faccia non taglia nessuna porzione del solido, che quindi rispetto a ciascuno di questi piani risulta «tutto da una parte», questo comporta il fatto che una faccia, e con essa gli spigoli che la determinano, o si vede o non si vede.

Quindi non esistono linee parzialmente nascoste, ma solo linee visibili o non visibili. Occorre quindi individuare un metodo con il quale ciascun segmento viene «giudicato» visibile. È evidente che il solido deve essere determinato tramite le sue facce, che a loro volta sono determinate tramite i segmenti che le compongono. Ogni segmento unisce due vertici. Conseguentemente va stabilito un metodo di memorizzazione dei dati

individuanti il solido che tenga conto della necessità di ragionare non solo per spigoli ma anche per facce.

Il metodo per stabilire la visibilità o meno di una faccia si basa sulla determinazione della sua normale. Ovvero se una retta perpendicolare alla faccia forma con la retta che unisce osservatore ed oggetto un angolo minore di 90 gradi significa che la faccia è «rivolta» verso l'osservatore, viceversa è rivolta dall'altra parte.

Il Test di Visibilità consisterà nel calcolo di questo angolo (uno per ogni faccia) e nel disegno dei segmenti che compongono le facce visibili.

Rispetto agli esempi pubblicati sul libro citato abbiamo apportato alcune varianti per rendere il programma più «didattico». Inoltre abbiamo utilizzato la routine di calcolo della normale per «abbozzare» un discorso sull'ombreggiatura. Infatti lo stesso ragionamento fatto per il Test di Visibilità può essere esteso

```

100 REM HIDDEN LINES - THE VISUALIZZAZIONI
110 REM
120 CLS:A1=.5:A2=1:D1=20:D2=40:S1=SIN(A1):C1=COS(A1):S2=SIN(A2):C2=COS(A2)
130 REM COORDINATE SPAZIALI E VIDEO
140 DIM V(10,3),W(10,2):FOR I=1 TO 10:READ X,Y,Z:V(I,1)=X:V(I,2)=Y:V(I,3)=Z
150 X1=-X*S1+Y*C1:Y1=-X*C1*Y*S1+C2+Z*S2:Z1=-X*S2*C1-Y*S2*S1-Z*C2+D2
160 W(I,1)=D1*(X1/Z1):W(I,2)=D1*(Y1/Z1):NEXT I
170 REM
180 NS=7:DIM P(NS):FOR I=1 TO NS:READ P(I):NEXT I
190 REM VERTICI INDIVIDUANTI CIASCUNA SUPERFICIE
200 DIM S(NS,10):FOR I=1 TO NS:FOR J=1 TO P(I):READ S(I,J):NEXT J:NEXT I
210 GOSUB 440:REM VISUALIZZAZIONE COMPLETA
220 REM CALCOLO DELLA NORMALE
230 DIM N(NS,3):FOR I=1 TO NS
240 U1=V(S(I,2),1)-V(S(I,1),1):V1=V(S(I,3),1)-V(S(I,1),1)
250 U2=V(S(I,2),2)-V(S(I,1),2):V2=V(S(I,3),2)-V(S(I,1),2)
260 U3=V(S(I,2),3)-V(S(I,1),3):V3=V(S(I,3),3)-V(S(I,1),3)
270 N(I,1)=U2*V3-V2*U3:N(I,2)=U3*V1-V3*U1:N(I,3)=U1*V2-V1*U2:NEXT I
280 REM DETERMINAZIONE SPIGOLI VISIBILI
290 X2=D2*S2*C1:Y2=D2*S2*S1:Z2=D2*C2
300 DIM E(12,3):DIM SN(NS):N=1:FOR I=1 TO NS:E1=S(I,1)
310 Y3=X2-V(E1,1):Y3=Y2-V(E1,2):Z3=Z2-V(E1,3)
320 IF N(I,1)*X3+N(I,2)*Y3+N(I,3)*Z3<=0 THEN NS(I)=1:GOTO 390
330 FOR J=2 TO P(I):E2=S(I,J):K=1
340 IF E(K,1)=E2 AND E(K,2)=E1 THEN 380
350 IF K=N THEN 370
360 K=K+1:GOTO 340
370 E(N,1)=E1:E(N,2)=E2:N=N+1
380 E1=E2:E(N,3)=E(K,3)+1:NEXT J
390 NEXT I
400 GOSUB 490:REM VISUALIZZAZIONE PER SPIGOLI
410 GOSUB 540:REM VISUALIZZAZIONE PER SUPERFICI S/N
420 GOSUB 620:CLS:END:REM FINE
430 REM ROUTINE DI VISUALIZZAZIONE
440 REM VISUALIZZAZIONE COMPLETA
450 GOSUB 620:SCREEN 2:CLS:WINDOW (-10,-9)-(10,8):LINE (-10,-9)-(10,8),,B
460 FOR I=1 TO NS:FOR J=1 TO P(I)-1
470 LINE (W(S(I,J),1),W(S(I,J),2))-(W(S(I,J+1),1),W(S(I,J+1),2))
480 NEXT J:NEXT I:RETURN
490 REM VISUALIZZAZIONE PER SPIGOLI
500 GOSUB 620:SCREEN 2:CLS:WINDOW (-10,-9)-(10,8):LINE (-10,-9)-(10,8),,B
510 FOR I=1 TO 12:IF E(I,3)=0 THEN 530
520 J=E(I,1):K=E(I,2):LINE (W(J,1),W(J,2))-(W(K,1),W(K,2))
530 NEXT I:RETURN
540 REM VISUALIZZAZIONE PER SUPERFICI
550 GOSUB 620:SCREEN 2:CLS:WINDOW (-10,-9)-(10,8):LINE (-10,-9)-(10,8),,B
560 FOR I=1 TO NS
570 FOR J=1 TO P(I)-1:IF NS(I)=1 THEN 590
580 LINE (W(S(I,J),1),W(S(I,J),2))-(W(S(I,J+1),1),W(S(I,J+1),2)):GOTO 600
590 LINE (W(S(I,J),1),W(S(I,J),2))-(W(S(I,J+1),1),W(S(I,J+1),2)),,B#4444
600 NEXT J
610 NEXT I:RETURN
620 PRINT CHR$(7)
630 I$=INKEY$:IF I$="" THEN 630 ELSE RETURN
640 REM vertici
650 DATA 5,7,-5,5,7,5,5,-7,5,5,-7,-5
660 DATA -5,7,-5,-5,-7,-5,-5,-7,5,5,-7,5,0,7,8,0,-7,8
670 REM punti per ogni superficie
680 DATA 5,6,5,6,5,5,5
690 REM vertici di ogni superficie
700 DATA 1,2,3,4,1,1,5,8,9,2,1,5,6,7,8,5,4,3,10,7,6,4
710 DATA 3,2,9,10,3,7,10,9,8,7,1,4,6,5,1

```

Figura 1 - Listato programma Hidden Lines. Lo spunto è stato tratto dal libro di Myers «Grafica con il Microcomputer» ed. Masson, nel quale viene esposta la tecnica di identificazione delle linee nascoste detta «della perpendicolare».

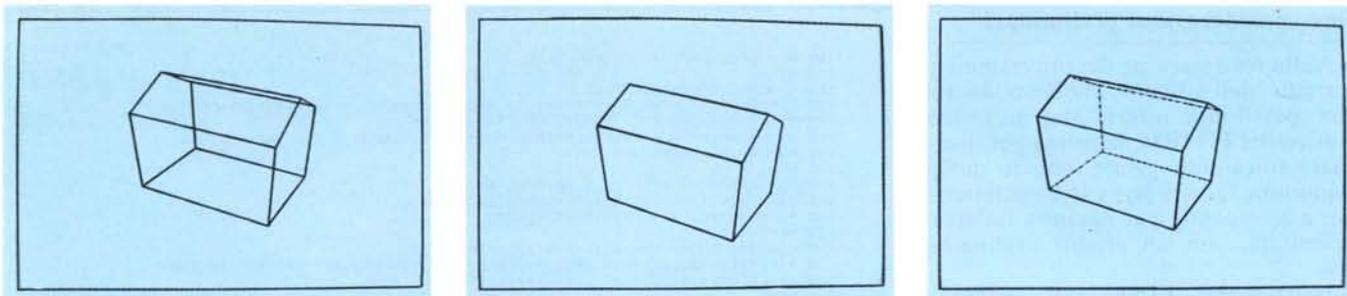


Figure 2.3-4 - Tre output del programma Hidden Lines. Rappresentano un solido convesso rispettivamente con tutte le linee, senza le linee nascoste e con le linee nascoste tratteggiate.

al calcolo dell'inclinazione di ciascuna faccia rispetto ad un ipotetico punto luminoso.

Le facce più esposte, cioè quelle la cui perpendicolare risulta meno inclinata rispetto al «raggio luminoso», saranno sempre meno illuminate man mano che l'angolo si avvicina a 90 gradi (che rappresenta l'orizzonte). Le facce che hanno una inclinazione di oltre 90 gradi sono nascoste dal solido stesso rispetto alla fonte luminosa e quindi non si vedono.

Un nuovo sistema per individuare il solido

Per poter utilizzare il Test di Visibilità basato sul calcolo della perpendicolare occorre memorizzare i dati individuanti il solido in una maniera differente da quella usata solitamente.

Nelle varie trattazioni di Computer Grafica Tridimensionale abbiamo sempre individuato il solido mediante una serie di linee spezzate nello spazio, che, unendo tra di loro alcuni ver-

tici, individuano tutti gli spigoli dai quali è costituita la figura.

Il metodo delle facce comporta sempre la necessità di memorizzare tutti i vertici, anche in un ordine qualsiasi. Va poi memorizzato un vettore di dimensione pari al numero delle facce e in cui ciascun elemento indica di quanti segmenti è costituita quella faccia.

Vi è infine una matrice a due dimensioni, la prima pari al numero delle facce e la seconda pari al massimo numero di segmenti costituenti una faccia, in cui vengono memorizzate le sequenze di vertici individuanti le facce.

In definitiva invece di utilizzare una sola matrice, se ne usano tre, con il vantaggio, però, di specificare le coordinate dei vertici con un ordine del tutto indipendente della modalità con la quale saranno visualizzati.

La determinazione delle facce è utilizzata anche per il calcolo della retta normale della faccia che può essere fatto conoscendo due segmenti giacenti su di essa.

Il meccanismo, sicuramente più complesso di quello classico, permette comunque la costruzione grafica del solido, anche non applicando la ricerca delle linee nascoste. In particolare nel primo programma presentato forniamo tre visualizzazioni. La prima del solido costruito con il metodo delle facce, con tutte le sue linee (visibili e non), la seconda rispetta il programma di Myers, che prevede la costruzione di una quarta matrice comprendente i segmenti visibili e la terza, che utilizza il Test della Perpendicolare direttamente in fase di visualizzazione delle facce.

Il secondo programma è simile al precedente solo che il Test non serve solo per determinare la visibilità o meno di una faccia ma per determinare il suo grado di «illuminazione» rispetto ad un punto luminoso, identificato per semplicità, con l'osservatore.

Il terzo programma riprende il discorso sui Solidi Geometrici pubblicato su MC. n. 14, per implementarlo con la routine di ricerca delle linee nascoste che qui si applica perfettamente.

```

100 REM hidden lines e ombreggiatura
110 REM
120 REM inizializzazioni
130 D1=20:D2=40:NS=7
140 DIM V(10,3),W(10,2):FOR I=1 TO 10:READ V(I,1),V(I,2),V(I,3):NEXT I
150 DIM P(NS):FOR I=1 TO NS:READ P(I):NEXT I
160 DIM S(NS,10):FOR I=1 TO NS:FOR J=1 TO P(I):READ S(I,J):NEXT J:NEXT I
170 DIM N(NS,3),NS(NS)
180 DIM Cx(6,2):FOR I=1 TO 6:READ Cx(I,1),Cx(I,2):NEXT I
190 REM
200 REM loop di movimento
210 FOR A=1 TO 3 STEP .2:A1=A:A2=2-A
220 S1=SIN(A1):C1=COS(A1):S2=SIN(A2):C2=COS(A2)
230 X2=D2*S2*C1:Y2=D2*S2*S1:Z2=D2*C2
240 REM
250 REM calcolo coordinate schermo
260 FOR I=1 TO 10:X=V(I,1):Y=V(I,2):Z=V(I,3)
270 Y1=-X*S1+Y*C1:Y1=-X*C1+Y*S1:Z1=-X*S2+C1-Y*S2*S1-Z*C2+D2
280 W(I,1)=D1*(X1/Z1):W(I,2)=D1*(Y1/Z1):NEXT I
290 REM
300 REM determinazione retta normale
310 FOR I=1 TO NS
320 U1=V(S(I,2),1)-V(S(I,1),1):V1=V(S(I,3),1)-V(S(I,1),1)
330 U2=V(S(I,2),2)-V(S(I,1),2):V2=V(S(I,3),2)-V(S(I,1),2)
340 U3=V(S(I,2),3)-V(S(I,1),3):V3=V(S(I,3),3)-V(S(I,1),3)
350 N(I,1)=U2*V3-V2*U3:N(I,2)=U3*V1-V3*U1:N(I,3)=U1*V2-V1*U2:NEXT I
360 REM
370 REM loop di visualizzazione
380 FOR I=1 TO NS:NS(I)=0:L=S(I,1)
390 NS(I)=N(I,1)*(X2-V(L,1))+N(I,2)*(Y2-V(L,2))+N(I,3)*(Z2-V(L,3))
400 NEXT I:GOSUB 420:NEXT A:END
410 REM
420 REM visualizzazione segmenti
430 SCREEN 2:CLS:WINDOW (-9,-6)-(9,6):,B
440 FOR I=1 TO NS:FOR J=1 TO P(I)-1:IF NS(I)<=0 THEN 460
450 LINE (W(S(I,J),1),W(S(I,J),2))-(W(S(I,J+1),1),W(S(I,J+1),2)):GOTO 470
460 LINE (W(S(I,J),1),W(S(I,J),2))-(W(S(I,J+1),1),W(S(I,J+1),2)),,H1111
470 NEXT J:NEXT I
480 REM
490 REM visualizzazione campiture
500 FOR I=1 TO NS:IF NS(I)<0 THEN 550
510 XX=(W(S(I,1),1)+W(S(I,3),1))/2:YY=(W(S(I,1),2)+W(S(I,3),2))/2
520 CC=INT(NS(I)/1000+1)
530 C1=CX(CC,1):C2=CX(CC,2):PAINT (XX,YY),CHR*(C1)+CHR*(C2)
540 PAINT (XX,YY),CHR*(C1)+CHR*(C2)
550 NEXT I:RETURN
560 REM
570 REM vertici
580 DATA 5,7,-5,5,7,5,5,-7,5,5,-7,-5
590 DATA -5,7,-5,-5,-7,-5,-5,-7,5,-5,7,5,0,7,8,0,-7,8
600 REM punti per ogni superficie
610 DATA 5,6,5,6,5,5,5
620 REM vertici di ogni superficie
630 DATA 1,2,3,4,1,1,5,8,9,2,1,5,6,7,8,5,4,3,10,7,6,4
640 DATA 3,2,9,10,3,7,10,9,8,7,1,4,6,5,1
650 REM coppie colori
660 DATA 0,17,0,51,68,17,204,51,255,221,255,255

```

Figura 5 - Listato del programma Ombre. Il metodo della perpendicolare viene modificato per individuare l'inclinazione di ciascuna faccia del solido rispetto ad un punto luminoso.

Due considerazioni preliminari

Nella realizzazione dei programmi a corredo dell'articolo abbiamo usato due possibilità offerte dal software grafico del PC IBM, la prima per disegnare linee tratteggiate, con le quali comunque far «vedere» le linee nascoste, e la seconda per campire un'area delimitata, con un effetto «retinatura».

L'istruzione `LINE (x0, y0)-(x1, y1),,H` accetta un parametro che definisce un tratteggio. Ad esempio se il valore del parametro `H` è 4444 in esadecimale, la linea viene disegnata con il corrispondente valore in bit/pixel. Cioè il valore in binario è 1000100010001000 e al valore 1 corrisponde un pixel acceso al valore 0 un pixel spento.

Per quanto riguarda la campitura, che si realizza con l'istruzione `PAINT (x0, y0),C`, il valore `C` può essere una somma di caratteri stringa, con i quali viene costruita una trama di pixel accesi e spenti, corrispondenti ai valori esadecimali passati nel parametro `C`.

Il primo programma (Hidden Lines)

Il primo programma produce tre visualizzazioni di un solido convesso. La caratteristica comune ai tre output consiste nel modo di memorizzare i dati, per facce. La differenziazione sta nel fatto che dapprima non viene eseguita la routine `Hidden Lines`, per verificare l'efficacia del metodo di memorizzazione dei dati, poi la routine viene eseguita nel modo descritto nel libro di Myers (memorizzando in un vettore i segmenti visibili) e la terza in un modo semplificato che evita di dover ricorrere a questo vettore.

Il programma è scomponibile in più parti. Alcune sono le routine iniziali di caricamento dati, alcune sono le subroutine comuni alle tre fasi, e in mezzo ci sono le tre routine principali, ciascuna per ogni visualizzazione.

La prima routine (riga 120) si occupa della definizione dei parametri di visualizzazione e cioè delle coordinate geometriche (angoli e distanze) individuanti il punto di osservazione. Il procedimento di restituzione tridimensionale è sempre lo stesso più volte utilizzato e descritto in questa rubrica.

La seconda routine (righe 140-160) definisce le due matrici `V`, in cui sono immagazzinate le coordinate tridimensionali dei vertici individuanti l'oggetto da visualizzare, e `W` in cui sono memorizzati i risultati (in termini di coordinate video) del procedimento di restituzione tridi-bidi.

È evidente che il calcolo delle coordinate video è del tutto indipendente dai risultati del test di individuazione

```

100 REM ICOSAEDRO - HIDDEN LINES
110 REM
120 D1=20:D2=40:READ NP,NS
130 DIM V(NP,3),W(NP,2):FOR I=1 TO NP:READ V(I,1),V(I,2),V(I,3):NEXT I
140 DIM P(NS):FOR I=1 TO NS:READ P(I):NEXT I
150 DIM S(NS,4):FOR I=1 TO NS:FOR J=1 TO P(I):READ S(I,J):NEXT J:NEXT I
160 DIM NS(NS),N(NS,3)
170 REM
180 FOR A=1 TO 3 STEP .2:A1=A:A2=2-A
190 S1=SIN(A1):C1=COS(A1):S2=SIN(A2):C2=COS(A2)
200 X2=D2*S2*C1:Y2=D2*S2*S1:Z2=D2*C2
210 REM
220 FOR I=1 TO NP:X=V(I,1):Y=V(I,2):Z=V(I,3)
230 X1=-X*S1+Y*C1:Y1=-X*C1*Y2-Z*S2:Z1=-X*S2*C1-Y*S2*S1-Z*C2+D2
240 W(I,1)=D1*(X1/Z1):W(I,2)=D1*(Y1/Z1):NEXT I
250 REM
260 FOR I=1 TO NS
270 U1=V(S(I,2),1)-V(S(I,1),1):V1=V(S(I,3),1)-V(S(I,1),1)
280 U2=V(S(I,2),2)-V(S(I,1),2):V2=V(S(I,3),2)-V(S(I,1),2)
290 U3=V(S(I,2),3)-V(S(I,1),3):V3=V(S(I,3),3)-V(S(I,1),3)
300 N(I,1)=U2*V3-V2*U3:N(I,2)=U3*V1-V3*U1:N(I,3)=U1*V2-V1*U2:NEXT I
310 REM
320 FOR I=1 TO NS:NS(I)=0:L=S(I,1)
330 IF N(I,1)*(X2-V(L,1))+N(I,2)*(Y2-V(L,2))+N(I,3)*(Z2-V(L,3))<=0 THEN NS(I)=1
340 NEXT I:GOSUB 400:NEXT A:END
400 REM
410 SCREEN 2:CLS:WINDOW (-4,-3)-(4,3):LINE (-4,-3)-(4,3),,B
420 FOR I=1 TO NS:FOR J=1 TO P(I)-1:IF NS(I)=1 THEN 440
430 LINE (W(S(I,J),1),W(S(I,J),2))-(W(S(I,J+1),1),W(S(I,J+1),2)):GOTO 450
440 LINE (W(S(I,J),1),W(S(I,J),2))-(W(S(I,J+1),1),W(S(I,J+1),2)),,H4444
450 NEXT J:NEXT I:RETURN
500 REM
510 REM data icosaedro
520 REM numero vertici,facce
530 DATA 12,20
540 REM coordinate vertici
550 DATA 0,0,5,3,58,0,2,1,11,3,4,2,-2,9,2,1,2
560 DATA -2,9,-2,1,2,1,11,-3,4,2,2,9,2,1,-2,-1,11,3,4,-2
570 DATA -4,0,-2,-1,11,-3,4,-2,2,9,-2,1,-2,0,0,-5
580 REM numero vertici per faccia
590 DATA 4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4
600 REM vertici per faccia
610 DATA 1,2,3,1,1,3,4,1,1,4,5,1,1,5,6,1,1,6,2,1
620 DATA 2,7,3,2,3,8,4,3,4,9,5,4,5,10,6,5,6,11,2,6
630 DATA 7,8,3,7,8,9,4,8,9,10,5,9,10,11,6,10,11,7,2,11
640 DATA 12,8,7,12,12,9,8,12,12,10,9,12,12,11,10,12,12,7,11,12

```

Figura 8 - Listato del programma `Icosaedro`. Le varie tecniche presentate si adattano ai solidi convessi e quindi anche ai «classici» ottaedro e icosaedro già «presentati» nel n. 14 di *MCmicrocomputer*.

delle linee nascoste. Compito di tale test sarà quello di determinare quali dati, e viene fatto direttamente sui dati video, appartengono a segmenti visibili e quali no.

La successiva routine (riga 180) carica in un vettore il numero di vertici (e quindi di segmenti) individuanti ciascuna faccia. Questo serve per mettere in relazione i vari vertici tra di loro. E di questo si occupa la successiva routine (riga 210) che immagazzina in una matrice di dimensioni pari al numero delle facce la sequenza di segmenti individuanti una faccia.

Per esempio un vertice di un cubo appartiene a tre facce. Il vertice superiore di una piramide a base quadrata appartiene a quattro facce, mentre un vertice della base appartiene solo a tre facce.

In un solido generico è possibile che il numero dei vertici di ciascuna faccia sia estremamente variabile. In tal caso la seconda dimensione della matrice dovrà essere pari al numero massimo dei vertici in una singola faccia.

Per determinare la normale (routine di riga 230) si utilizzano elementi di calcolo vettoriale, che tralasciamo. Diremo solo che il programma utilizza i due segmenti corrispondenti ai primi tre vertici di ogni faccia per individuare il piano cui questa appartiene. Come noto un piano è determinato tramite tre punti, o, ed è lo stesso, tramite due rette che si intersecano.

A questo punto occorre verificare se le normali alle singole facce formino o meno un angolo minore di 90 gradi con la direzione punto di vista ogget-

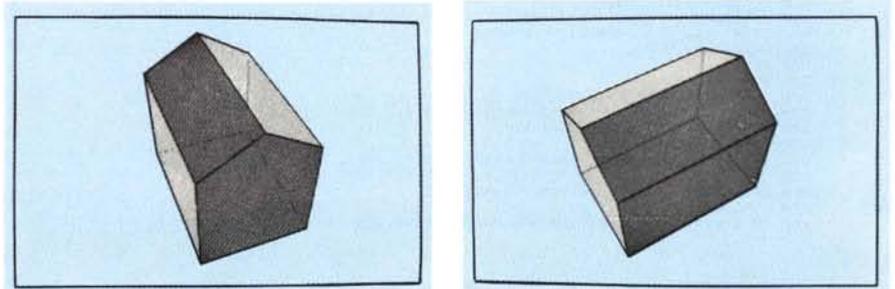


Figure 6, 7 - Due output del programma `Ombre`. L'effetto di chiaroscuro può essere realizzato solamente tramite una «retinatura» di pixel, quindi è tanto più efficace quanto più numerosi sono i pixel a disposizione.

to. La versione Myers del programma produce un ulteriore vettore dove sono immagazzinati i numeri corrispondenti ai progressivi dei soli segmenti visibili. Questo procedimento ha il vantaggio che un segmento viene tracciato una sola volta anche se appartiene a due facce ambedue visibili. Lo svantaggio sta nel fatto che la routine, oltre al fatto che comporta l'uso di un'altra matrice, è complessa.

La soluzione più semplice consiste nell'utilizzare un vettore di flag, legati non ai segmenti ma alle facce, che indichino se la faccia è visibile o meno. Il vantaggio di una maggiore semplicità è compensato dallo svantaggio di dover disegnare alcuni elementi del disegno due volte.

Completata la parte «calcolo» vi sono in sequenza le due istruzioni che ri-

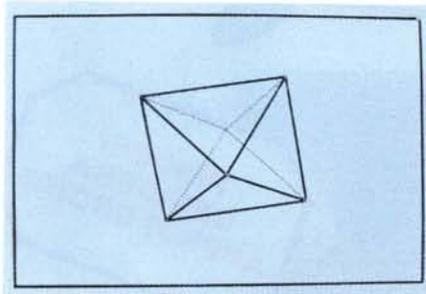


Figure 9, 10 - Due output del programma Ottaedro. Purtroppo i solidi regolari presentano numerose simmetrie per cui le varie viste si assomigliano tutte.

chiamano le due visualizzazioni, quella per spigoli e quella per superfici (righe 400-410).

Le routine di visualizzazione vera e propria sono:

- la visualizzazione completa, senza eccezioni (riga 440), che viene eseguita per mezzo di due loop nestati. L'esterno fa scorrere le facce, l'interno fa scorrere i vertici di ogni singola faccia.

- la visualizzazione per spigoli, che preleva i dati direttamente dalla matrice dei segmenti visibili (riga 490).

- la visualizzazione per superfici, che a seconda che il flag di visibilità sia 0 o 1, traccia i segmenti di ogni faccia tratteggiati o continui (riga 540).

I risultati «esteriori» del programma si vedono nelle figure 2, 3, 4.

Il secondo programma (Ombre)

In realtà non si tratta di ombre ma di «chiaroscuro». Il chiaroscuro è una tecnica di disegno che serve per dare una certa tridimensionalità ad un soggetto che altrimenti risulterebbe piatto. Ad esempio nel disegno di una sfera è difficile far capire che il soggetto è una sfera e non un cerchio.

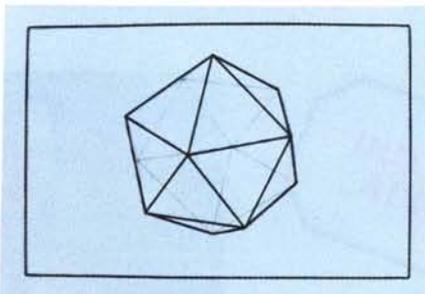
La tecnica del chiaroscuro consiste

nel riprodurre le differenti luminosità che assumono le varie porzioni del soggetto rispetto ad una fonte luminosa che lo illumina.

Le zone più illuminate, che cioè risulteranno più chiare nel disegno, sono quelle più esposte alla luce. Questa condizione corrisponde ad una precisa legge fisica che è quindi facilmente «calcolabile».

La sorgente produce un flusso luminoso che investe il soggetto. La quantità di luce che raggiunge la superficie del soggetto dipende dalla posizione che questa assume nei confronti della sorgente, ovvero tanto più è inclinata tanto meno luce «intercetta», in quanto una stessa quantità di luce si distribuisce su un'area maggiore.

La misura di questa quantità è data da una semplice legge trigonometrica



legata all'angolo formato tra il raggio luminoso e la normale alla superficie. Se l'angolo è zero e cioè il raggio è perfettamente perpendicolare alla superficie, l'illuminazione è massima, via via che l'angolo si avvicina ai 90 gradi diminuisce, per azzerarsi per 90 o più gradi.

Se il solido è curvo il calcolo vale lo stesso solo che si considerano areole infinitesimali in quanto la condizione varia da «chiaro» a «scuro» senza soluzione di continuità.

Il discorso teorico è molto semplice, purtroppo la «resa» in termini grafici, a livello microcomputer, è insoddisfacente. Vediamo perché.

Per ottenere risultati apprezzabili occorre avere una scala di grigi (per grafica monocromatica) o una scala di colori (per grafica a colori) molto estesa. Questa non è una caratteristica corrente nei micro di questa generazione.

In grafica a colori sono necessarie almeno 16 tonalità intermedie tra il colore di massima illuminazione e quello di minima illuminazione. Questa significa che i colori disponibili debbono essere almeno 256 e indirizzabili separatamente. Per ottenere tali prestazioni occorrono video memory da 256 Kb in su.

Per quanto riguarda il monocroma-

tico esistono due possibilità, che il software grafico usato disponga di livelli di grigio, e questo è la stessa problematica del colore, anche se «serve meno memoria», o che esista solo il pixel acceso e spento.

Quest'ultimo caso che è il più diffuso peggiora la situazione in quanto occorre «fabbricarsi» i livelli di grigio intermedi miscelando pixel accesi con pixel spenti. Questo è abbastanza facile da fare ma comporta uno scadimento nella definizione del disegno. Ad esempio si possono comporre quadratini di 9 pixel per 9 pixel. Le possibilità in termini di pixel accesi/spenti sono 10, cui corrispondono teoricamente 10 livelli di grigio.

In realtà esistono due difficoltà. La prima è che non è detto che le dieci combinazioni costituiscano una scala di grigi, che cioè «spegnendo» un pixel si abbia un grigio del 10 per cento più vicino al nero. La seconda è che la combinazione di questi quadratini mostra una trama che allontana la sensazione di grigio. Cioè se occorre comporre un quadratino con 3 pixel accesi e 6 spenti il risultato sarà sempre una specie di tratteggio, fitto quanto si vuole, ma che anche con definizioni di oltre 100.000 pixel, si nota.

Per avere un'idea dei risultati ottenibili mostriamo due uscite su scheda grafica Hercules (720 per 348 pixel) per IBM (figg. 6, 7).

Il programma (listato in fig. 5) è simile al precedente, e quindi non lo descriviamo in dettaglio. È stata utilizzata solo la terza metodologia e sono state aggiunte due routine.

La prima routine serve per dare movimento alla figura e consiste in un loop in cui vengono fatti variare i parametri di visualizzazione (riga 200).

La seconda routine è quella che realizza le campiture (riga 490). Le campiture sono realizzate con l'istruzione PAINT (X, Y),CS, che accetta nella stringa CS una composizione di caratteri che vengono tradotti in pixel accesi e spenti in funzione del valore in binario del carattere.

Con tale funzione sono stati utilizzate 8 coppie di caratteri, che permettono una modestissima scala dei grigi.

Ultimo programma (Icosaedro)

Le tecniche illustrate valgono solo per la rappresentazione dei solidi convessi. Possono essere quindi applicate con successo ai solidi regolari (CUBO, OTTAEDRO, ICOSAEDRO, DODECAEDRO).

Forniamo il programma per l'icosaedro (listato in fig. 8) ricordando che il tema dei solidi regolari è stato trattato nel n. 14 di MCmicrocomputer (figg. 9, 10).

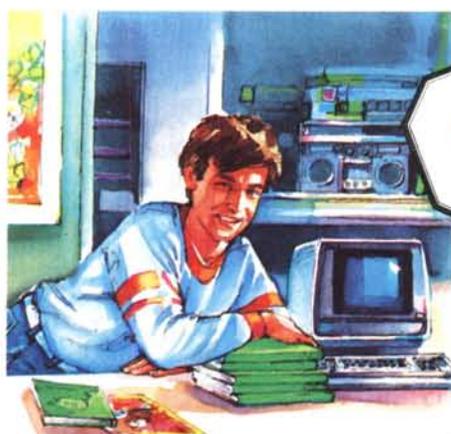
**COSTRUISCI LA PRIMA BIBLIOTECA COMPLETA
DI INFORMATICA PERSONALE**

QUADERNI JACKSON

DI PERSONAL COMPUTER



Volume per volume, settimana per settimana, tutti gli argomenti dell'informatica personale: da cosa significa "vivere col computer" al BASIC; dal COBOL al PASCAL e ai più diffusi linguaggi e sistemi operativi (MS-DOS, CP/M, MSX ...) Dall'hardware al software, dalla programmazione dei personal computer all'utilizzo dei più affermati pacchetti software: Multiplan, Visicalc, Lotus 1-2-3, Symphony dBase II, Wordstar ...



**PER CHI
INIZIA**

la sua avventura nel mondo dei computer e ha bisogno di una guida sicura per costruire le basi della sua conoscenza ed esperienza.



**PER CHI
DESIDERA
APPROFONDIRE**

per ragioni di studio, hobby, lavoro, precise tematiche di software, di hardware e di programmazione.

**PER CHI
VUOLE
POSSEDERE**

per esigenze professionali o culturali, un'organica Biblioteca di testi di base, scritti da esperti qualificati, coerente con un disegno editoriale di ampio respiro.



QUADERNI JACKSON

DI PERSONAL COMPUTER

Una Biblioteca di agili monografie, rivolta ad un vastissimo pubblico: appassionati, professionisti, studenti e tecnici. 30 Volumi pratici redatti con uno stile chiaro, competente, accurato ed essenziale.

**IN EDICOLA
I PRIMI 3 LIBRI
A SOLE
L. 9.800**



**GRUPPO EDITORIALE
JACKSON**

DIVISIONE LIBRI