

Els gràfics amb ordinador

Una de les utilitzacions dels ordinadors és la possibilitat de fer gràfics. La capacitat gràfica dels ordinadors deixa enrera, es pot dir que definitivament, aquells excel·lents artesans que fa anys

resolien els problemes de representació, tant en el disseny com en la recerca. Els ordinadors no solament podran aconseguir nivells de qualitat semblants, sinó que tenen noves capa-

citats... En aquest treball es presenten les principals direccions que poden conduir a fer representacions gràfiques, des d'algun exemple senzill fins a altres de més complicats.

En l'actualitat, és un fet que cada cop s'utilitzen més les possibilitats de sortida gràfica dels ordinadors. Podríem citar algunes de les raons d'aquesta evolució:

– El cost dels perifèrics de sortida gràfica, que ha anat disminuint en els últims anys. Així, s'ha passat, de tenir com a única possibilitat de dibuix els *plotters* molt sofisticats, a la comercialització de microordinadors "personals" amb pantalles gràfiques i possibilitat de còpia en paper del contingut de la pantalla.

– L'interès que tenen les gràfiques a l'hora de la interpretació dels resultats d'un programa. És molt més còmode interpretar una gràfica que esbrinar el comportament d'un fenomen a partir de diferents pàgines de resultats.

– La possibilitat de disposar de sistemes d'edició de dibuixos, plànols i mapes: Un cop entrada tota la informació gràfica, aquesta es pot emmagatzemar de forma estructurada en una base de dades gràfica; això permet, un altre dia, recuperar la informació (tota o només una part, i amb la possibilitat de canviar l'escala) o bé modificar-la en part amb un esforç mínim per part de l'usuari.

– La possibilitat de realitzar dissenys gràfics interactius, on l'usuari pot crear objectes de la forma que desitja tals com carrosseries de cotxes o fuselatges d'avió. Un cop creat, l'ordinador pot simular l'existència de l'objecte i donar informació sobre les seves propietats: volum, superfície, moments d'inèrcia i resistència a la fractura. La fase final consistiria en la confecció automàtica del plànol de la peça.

– El disseny automàtic de circuits electrònics (circuits impresos i circuits integrats) on l'ordinador calcula el camí que han de seguir els conductors, i un *plotter* dibuixa finalment la versió definitiva del circuit.

– Finalment, podríem citar l'interès

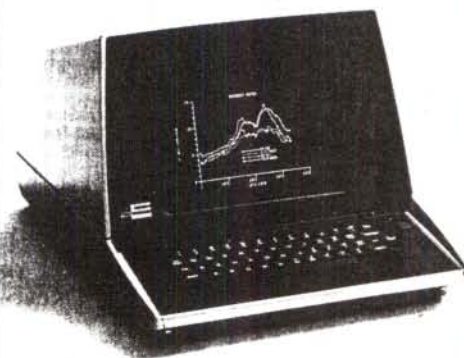
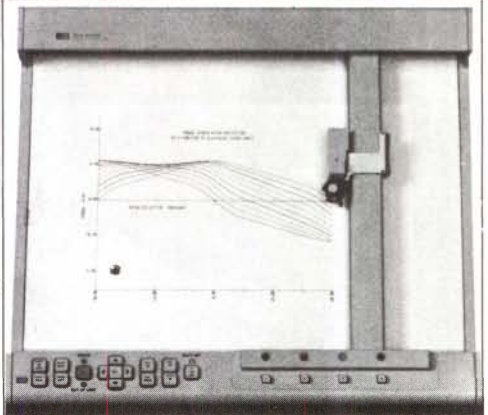


Fig. 1.- Un *plotter* de taula, i una pantalla gràfica.



dels gràfics en la presentació de resultats al públic no informàtic: histogrames i gràfiques d'estadístiques no econòmiques, possibilitat d'elements gràfics en els paquets d'ensenyament assistit per ordinador, etc. Aquest treball vol donar a conèixer alguns algorismes que fan servir els sistemes de sortida gràfica, amb els diferents passos que cal fer fins a arribar a la sortida final. Un cop feta la presentació d'alguns dels perifèrics més usuals a l'apartat 1, en els següents apartats s'analitzen un conjunt de problemes de complexitat creixent: dibuix d'una recta, dibuix d'un triangle, dibuix d'una corba coneguda, disseny de corbes i superfícies, i algorismes relacionats amb les representacions d'objectes tridimensionals.

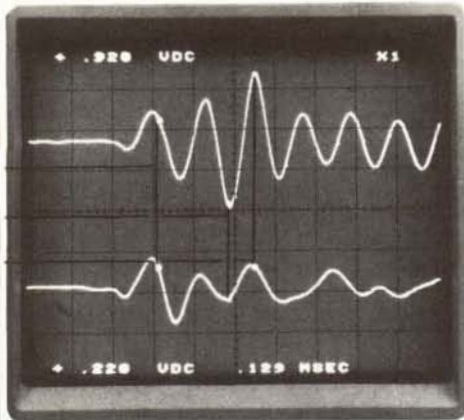
Els perifèrics de sortida gràfica

Quan una persona té necessitat d'una sortida gràfica de resultats, el més habitual és que desitgi tenir-los escrits en paper. Per aquesta raó, els perifèrics de sortida gràfica més coneguts (un perifèric de sortida és un dispositiu que permet que l'ordinador ens presenti els resultats) són els *plotters*.

A la figura 1 veiem un *plotter* de taula.

En ells, el paper per dibuixar es col·loca damunt d'una superfície horitzontal que disposa d'algun mecanisme per a la seva subjecció. Mitjançant un braç, que es pot desplaçar longitudinalment i un carret, que a la seva vegada es pot moure al llarg del braç, la ploma o plomes del carret poden situar-se a qualsevol posició damunt del paper. De fet, hi ha dues operacions elementals que podem fer amb el *plotter* des de l'ordinador: anar, amb la ploma aixecada, a una determinada posició de coordenades (x, y) —operació anomenada *moure*—, i anar, des de la posició on som i en línia recta, a una nova posició de coordenades predefinides amb una de les plomes baixada. Aquesta última operació, que dibuixa un segment recte sobre el paper, s'anomena *línia*. És evident que podem aconseguir qualsevol dibuix amb una seqüència adequada d'operacions línia i moure.

Encara que els *plotters* de taula són els més coneguts, n'hi ha altres tipus, com poden ser els de tambor on en lloc de moure's el braç longitudinalment és el paper el que es mou, ja que està adherit a la superfície d'un tambor cilíndric que gira en un i altre sentit, o bé els *plotters* de rodet; aquí també es mou el paper passant a un costat i a l'altre del *plotter* a causa del gir d'un rodet.



per Pere Brunet i Crosa

Pere Brunet i Crosa (Barcelona, 1948) és enginyer industrial des del 1970 per l'Escola Tècnica Superior d'Enginyers Industrials de Barcelona, on va doctorar-se el

1976. Des del 1971 és professor d'aquesta escola, primer de mecànica i després d'informàtica.



Fig. 2.- Exemple de dibuix obtingut amb un plotter o amb una pantalla de recorregut aleatori.

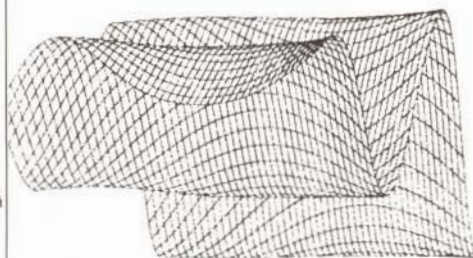
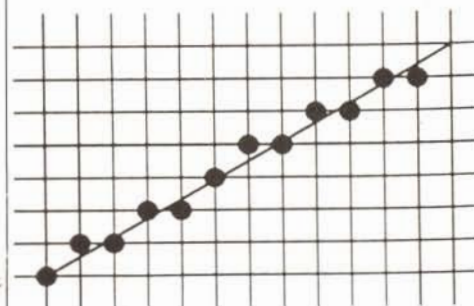


Fig. 3.- Dibuix obtingut amb una pantalla d'escombrat, i detall d'alguns dels punts que serveixen per a representar una recta inclinada.



No obstant això, i encara que els *plotters* solucionen la sortida final de resultats gràfics, són lents, i el seu ús pot ser carregós quan el que volem és fer un dibuix, veure com ha sortit, canviar-ne alguna part si no ens agrada i anar repetint aquest procés fins a arribar al gràfic final (procés que s'anomena generació *interactiva* de gràfics). Això, junt amb els fets que els ordinadors cada cop permeten més una resposta pràcticament immediata, és el que ha portat a la necessitat de les pantalles gràfiques.

Una pantalla gràfica (vegeu també la figura 1) és una pantalla que de cara a l'usuari es comportarà de manera quasi idèntica a un *plotter*: podrem "dibuixar sobre la pantalla amb ordres de moure sense dibuixar i de línia" (pintar un segment recte). Hi ha una limitació: no podem anar a qualsevol punt de la pantalla, sinó només als d'una malla la finor de la qual depèn de la qualitat de la pantalla. Per exemple, una pantalla de 1.000 per 800 punts permet direccionar punts de coordenades (x,y) tals que x i y són enters, i a més $0 \leq x \leq 999$, $0 \leq y \leq 799$. En conseqüència, el dibuix que obtindrem a la pantalla serà menys precís que el que al final dibuixarem amb el *plotter*; encara que aquest no és pas un problema en la fase de disseny interactiu.

Quant a tipus de pantalles, és important indicar, entre d'altres classificacions, la que distingeix entre les de recorregut aleatori i les d'escombrat. Les primeres es comporten de forma molt semblant als *plotters*: dibuixen trams rectes a la pantalla (això sí, entre punts amb coordenades direccionables a pantalla). Un exemple de dibuix en té a la figura 2. En canvi, les segones no permeten dibuixar segments rectes; únicament es pot dibuixar punt a punt d'entre els punts direccionables. És com si, a l'exemple anterior, tinguéssim un gran plató amb 800 files de 1.000 bombetes. Per tal de dibuixar, l'únic que podríem fer és encendre algunes d'aquestes bombetes. Aquestes pantalles han de representar els segments rectes "encenent" els punts que es troben més a prop de la recta teòrica. El resultat, com veiem a la figura 3, són representacions gràfiques on les rectes, mirades amb detall, apareixen com esglaonades. En sistemes gràfics més avançats que permeten tonalitats de gris —en el símil anterior serien bombetes mig enceses— es pot corregir força el defecte d'esglaonament dels segments rectes.

Aquesta possibilitat d'incloure nivells de gris, així com la de treballar en color, fa que, a pesar d'aquest defecte, les pantalles d'escombrat siguin molt utilitzades.

Un problema elemental: Dibuixar un triangle

Per tal de veure amb detall totes les operacions que comporta la realització d'un dibuix, pensem en un problema que a primera vista sembla molt senzill: en una pantalla gràfica volem representar un triangle que ja hem dibuixat en paper, i els costats del qual fan 32, 37 i 44 centímetres. Per tal de poder donar les ordres de dibuix corresponents, haurérem de començar per prendre dues decisions:

— Quin tros de pantalla volem utilitzar per fer el dibuix. Encara que molt sovint es vol un dibuix que ocupi tota la pantalla, en alguns casos pot no ser així: Si n'ocupem només una quarta part, podrem veure a la mateixa pantalla tres dibuixos més a part del que ara considerem (podem pensar, per exemple, en una pantalla que ens mostra la planta, alçat, perfil i una perspectiva d'un objecte).

— Quin tros del món real volem veure al tros de pantalla ja elegit. Això és el que es diu definir una finestra en el món real. Si la finestra és de 50 per 50 centímetres, hi podem tenir tot el triangle dins; si la definim de 25 per 25 centímetres, a la pantalla ens apareixerà una figura el doble de gran —és com si ha-

Fig. 4.- Un triangle, i les seves representacions a dues pantalles gràfiques. El tros de pantalla es idèntic en els dos casos, però la finestra en el món real és diferent. En el dibuix de baix, la finestra és massa petita i cal utilitzar un algorisme de retallat. ▾

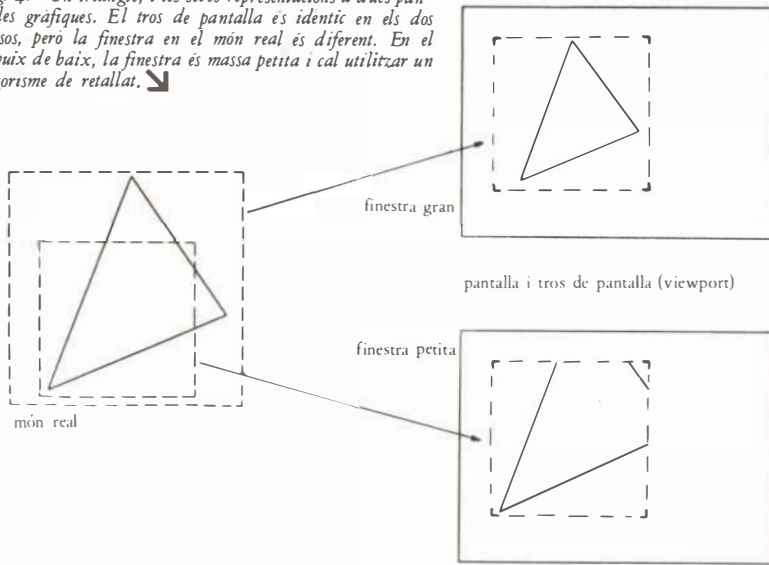
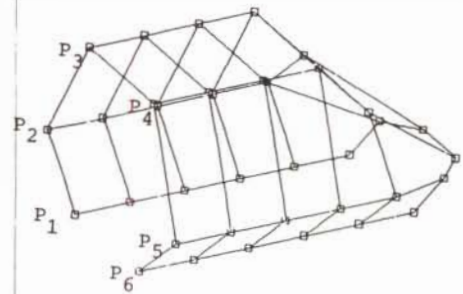


Fig. 5.- Conjunt de punts a l'espai entrats per l'usuari, units per segments rectes. Superfície obtinguda per B-splines a partir d'aquests punts. ▾



guéssim fet un zoom-, però evidentment no podrem veure tot el triangle; finalment, si la definim de 100 per 100 centímetres el triangle ens apareixerà la meitat de petit. (vegeu la figura 4).

En els sistemes gràfics no elementals, l'usuari ha d'entrar a l'ordinador les dades de la finestra i del tros de pantalla corresponent, abans de començar amb les ordres de dibuix (subministrant, per exemple, les coordenades màxima i mínima). Un cop definits ja aquests paràmetres, es podrà començar realment el dibuix del triangle: segons el que hem dit a l'apartat anterior, podríem pensar a executar una ordre "moure" per tal de posicionar-nos en un dels vèrtexs de triangle seguida de tres ordres "línia que dibuixarien cada una de les arestes. En una d'aquestes ordres línia, disposem de coordenades de dos punts -l'anterior, (x_1, y_1) i l'actual (x_2, y_2) - i pretenem que la màquina ens dibuixi el segment que els uneix. Aquest procés tampoc no és elemental, sinó que requereix tres operacions.

- en primer lloc, cal aplicar una transformació a les coordenades (X_1, Y_1) , (X_2, Y_2) del món real, per convertir-les en coordenades de pantalla; aquesta transformació depèn evidentment de la nostra definició de finestra i tros de pantalla.

- en segon lloc, cal saber si el segment és totalment interior al tros de pantalla, totalment exterior o bé té una part dins i una altra fora. En el primer cas el dibuixarem, mentre que en el segon cas no; en el tercer cas, cal esbrinar quin tros del segment és visible, i canviar les coordenades dels extrems del segment inicial a la dels extrems del tros visible. Aquesta operació de retallat o canvi de coordenades dels extrems, la realitza automàticament el mateix perifèric de sortida gràfica, mitjançant un microordinador que ha estat programat adequadament.

- finalment, si cal dibuixar algun tros

del segment inicial, i la representació es realitza en una pantalla d'escombrat, cal "construir" el segment decidint un a un quins són els punts de la pantalla que cal il·luminar. Vegem com es construeix una recta, per exemple, amb l'algorisme de Bresenham (suposem que $X_2 < Y_2 - Y_1$).

- a) Calcular $\Delta X, \Delta Y$ com els increments $X_2 - X_1, Y_2 - Y_1$ respectivament.
- b) Calcular e com $\Delta Y / \Delta X - 0,5$, i posar-nos en el punt de pantalla de coordenades X_1, Y_1 .
- c) Repetir els punts següents,
 - c.1) Posar un punt a la posició on som
 - c.2) Si e és positiu, pujar una unitat l'ordenada del punt on som, i disminuir e en una unitat.
 - c.3) Augmentar en una unitat l'abscissa del punt on som.
 - c.4) Incrementar e en la quantitat $\Delta Y / \Delta X$.

Fins arribar a una abscissa superior a X_2 . Es deixa al lector la comprovació que aquest algorisme realment dibuixa una recta que uneix els punts (X_1, Y_1) , (X_2, Y_2) . (S'ha suposat que les coordenades de pantalla són sempre valors enters).

El disseny i representació de corbes

A l'apartat anterior hem vist amb detall les operacions que ha de realitzar un perifèric gràfic per tal de representar un dibuix senzill. Si el que volem és representar una corba més complexa, cal distingir dos problemes diferents: o bé volem veure una corba d'equació analítica coneguda, o bé volem dissenyar una corba que tingui una forma determinada, encara que no coneguem la seva equació. En el primer cas, la corba, encara que no es veu a simple vista, queda dibuixada con un conjunt de pe-

tits trams rectes que uneixen punts consecutius d'aquesta, amb un algorisme que seria semblant a

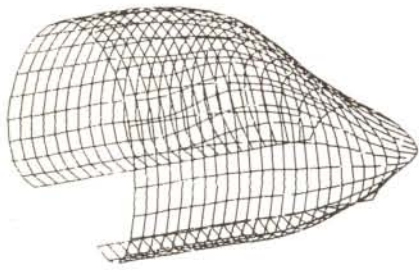
- a) fer una ordre moure el primer punt (X_1, Y_1) de la corba
- b) repetir
 - b.1) incrementar en una quantitat molt petita l'abscissa
 - b.2) Calcular la nova ordenada amb l'equació de la corba, $y = f(x)$
- c.2) Fer una ordre línia a aquest nou punt

Fins a acabar la corba.

En canvi, en el segon cas, l'usuari entra les coordenades d'alguns punts i disposa d'un algorisme que calcula una corba la forma de la qual s'acosta a la de la poligonal formada pels punts entrats. En alguns sistemes la corba no passa pels punts coneguts, i aquests només serveixen per modelar-ne la forma; en d'altres sistemes, la corba obtinguda passa per -interpolat- els punts coneguts. (Dels primers podríem citar els mètodes basats en corbes de Bézier i alguns dels basats en B-Splines; en els segons s'utilitzen splines cúbics cardinals o B-Splines). Nosaltes només parlarem dels segons.

Podem imaginar que la corba resultant constitueix el perfil d'un objecte de revolució que estem dissenyant; així, variant la posició dels punts variarem la forma de la corba i per tant la de l'objecte. En d'altres aplicacions ens interessarà dissenyar l'eix d'un conducte tubular de trajectòria complexa. Finalment, en molts casos, com es veu a la figura 5, dissenyem diferents corbes que constituiran un conjunt de perfils o llesques de la superfície resultant; en aquest cas, podrem modelitzar i després representar gràficament superfícies corbes de forma complexa: carroceries, fuselatges, els plànols de les quals no es podrien obtenir usant les tècniques clàssiques de dibuix.

Tornem a la figura 5, i vegem en detall el procés de generació de la superfície a



angles que defineixen la posició de l'observador ↓

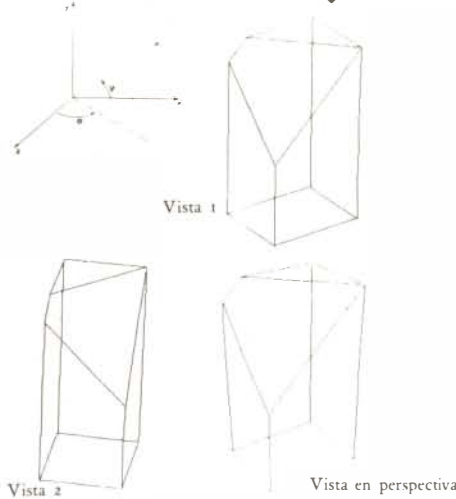


Fig. 6.— Un objecte tridimensional vist desde diferents direccions. Efecte de la perspectiva.

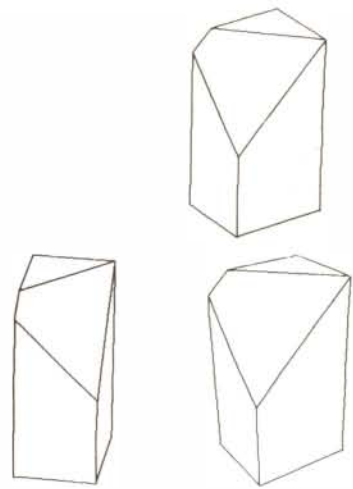


Fig. 7.— Les vistes de la figura 6, eliminant les parts amagades.

partir dels punts entrats.

La part superior de la figura representa aquests punts, units per trams rectes. Com es veu, ens queda una superfície formada per cares planes (o quasi planes) massa grans, i que no és per tant gens suau. Considerem ara els 6 punts P_1, \dots, P_6 del perfil de l'extrem. L'algorisme d'interpolació per splines cúbics genera, a partir d'ells, la corba que veiem a l'extrem de la superfície inferior. Aquesta corba compleix:

- que passa per tots els punts P_1, \dots, P_6
- que, tant pel que fa a la coordenada X com la Y, l'equació matemàtica d'aquesta corba és la d'un polinomi cúbic diferent per cada un dels cinc intervals $[P_1, P_2], \dots, [P_5, P_6]$. Aquesta propietat assegura la suavitat de les corbes dins de cada un d'aquests intervals.

— que els coeficients d'aquests polinomis són tals que es garanteix el mínim canvi en passar d'un interval al següent. En concret, es calculen de forma que, tot i que canvia el polinomi quan travessem un punt intermedi, no varia ni el pendent ni la curvatura.

Per tant la corba no és una sola funció d'equació matemàtica coneguda. En el nostre cas, està formada per cinc polinomis diferents que els enllacen de manera que no es nota el pas de l'un a l'altra. L'algorisme que acabem d'exposar és el més utilitzat en el disseny de corbes, ja que es pot demostrar que, de totes les corbes que passen pels punts donats, l'spline cúbic és el més suau. A més (sota algunes condicions), també es pot demostrar que la corba dissenyada adopta la forma que tindria una vareta flexible d'acer que obliguéssim a passar pels punts donats. Justament, aquesta curiosa propietat és la que ha donat el seu nom als splines (spline, en anglès, és una vareta flexible que fan servir els dibuixants per dibuixar corbes).

Un cop vist l'algorisme de generació del primer perfil a partir dels punts entrats,

el camí ja és molt més senzill: és immediat generar, de la mateixa manera, els altres perfils; després, només cal interpolar aquests perfils en sentit perpendicular al de construcció dels perfils per tal d'obtenir la superfície final que apareix a la part inferior de la figura 5.

Per acabar, direm que l'algorisme que acabem de veure no sols permet la construcció de corbes i superfícies; un cop obtingut un primer disseny, l'usuari pot anar modificant la posició dels punts entrats i així anar variant la forma de la corba o superfície fins a arribar a la desitjada.

Representació d'objectes tridimensionals

Un cop hem dissenyat un objecte tridimensional, si el volem representar gràficament hem d'obtenir una, o diverses projeccions bidimensionals. Tant si ens decidim per una projecció paral·lela (dièdrica, axonomètrica, cavallera, etc) com si desitgem una perspectiva del cos, el procés és el mateix. Cal projectar cada un dels vèrtexs del cos, de coordenades (X,Y,Z), a l'espai, al pla de la pantalla, i després, unir amb rectes a la pantalla els vèrtexs que estaven units a la representació tridimensional del cos. Per tal de calcular la projecció dels vèrtexs del cos, en els sistemes gràfics s'utilitzen coordenades homogènies (vegeu la ref. 3) que permeten efectuar transformacions complexes: composició de rotacions, translacions, simetries, escalats i perspectives, multiplicant les coordenades del punt de l'espai per un matriu de transformació de 4 per 4. Nosaltres veurem únicament un algorisme senzill de projecció en una direcció determinada, per tal d'il·lustrar el procediment cal seguir.

Suposem que hem definit les coordina-

des dels vèrtexs d'un objecte en un sistema de coordenades tal que l'eix X és horitzontal, l'eix y és vertical i el Z és horitzontal i mira cap a l'observador. Si pensem que la pantalla és una càmera i que comencem girant un angle θ al voltant de l'eix vertical (y) fins a col·locar-nos en l'orientació adequada, i després l'elevem un angle φ (vegeu figura 6), podem calcular, per cada vèrtex de coordenades (x,y,z),

$$\begin{aligned} x_1 &= x \cos \theta - z \sin \theta \\ y_1 &= -x \sin \theta \sin \varphi + y \cos \varphi - z \cos \theta \sin \varphi \\ z_1 &= x \sin \theta \cos \varphi + y \sin \varphi + z \cos \theta \cos \varphi \end{aligned}$$

i en aquest cas, les coordenades (x_1, y_1) són les coordenades a pantalla de la projecció axonomètrica del vèrtex després de girar la càmera els angles indicats θ, φ . Per tant, és molt senzill, un cop donada l'orientació de la càmera, calcular la posició (x_1, y_1) de tots els vèrtexs projectats a la pantalla i finalment poder dibuixar el cos. La mateixa figura 6 ens mostra diverses projeccions d'on mateix cos, variant l'orientació del punt de vista.

Tot això eren projeccions paral·leles. Si el que volem és una perspectiva on l'observador és a una distància de l'origen en l'orientació θ, φ , cal prendre com a coordenades de pantalla corresponents al vèrtex de coordenades (x,y,z) —vegeu altra cop la referència 3—,

$$\left(\frac{x_1}{1 - \frac{z_1}{d}}, \frac{y_1}{1 - \frac{z_1}{d}} \right)$$

En aquest cas, les coordenades del punt depenen de la profunditat z_1 d'aquest, que abans no feiem servir. Es veu molt clar, per exemple, que si el punt és molt lluny de l'observador (z_1 negatiu i molt gran), el denominador és molt elevat, i les dimensions x_1, y_1 es veuen molt petites a la pantalla. La figura 6 ens presenta, junt amb les projeccions pa-

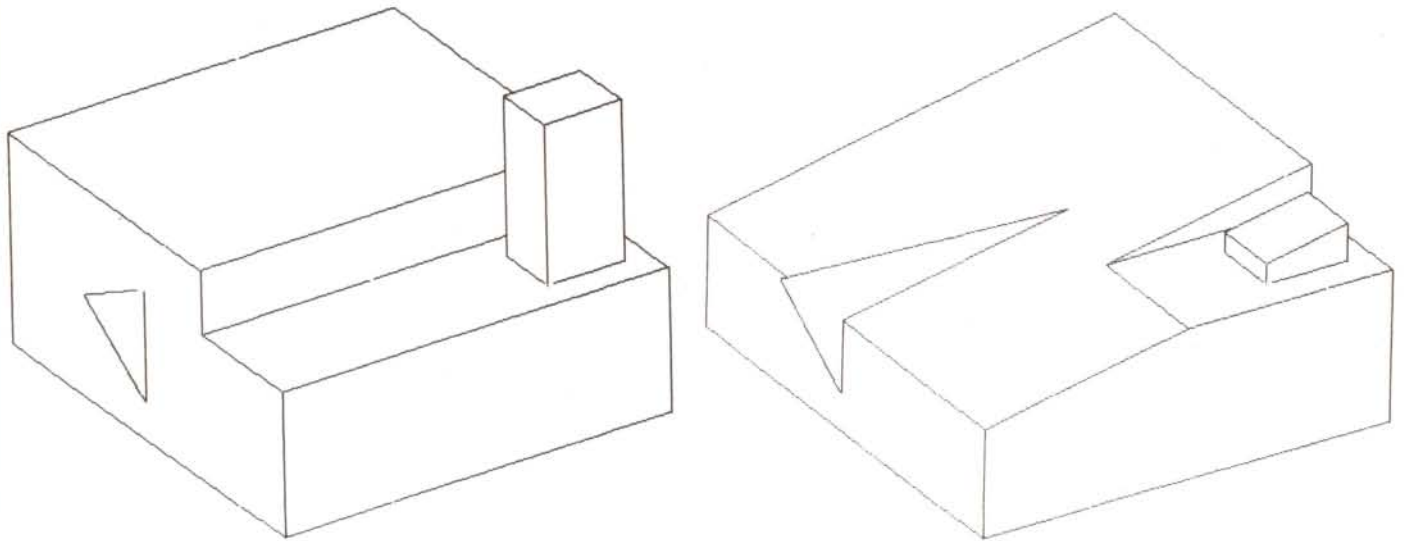


Fig. 9.- Un objecte, i el mateix objecte seccionat.

ral·leles, una perspectiva de l'objecte. El resum d'aquest apartat podria ser que les transformacions a dos dimensions impliquen realitzar, per a tots els vèrtexs del dibuix, un conjunt d'operacions molt senzill i sistemàtic.

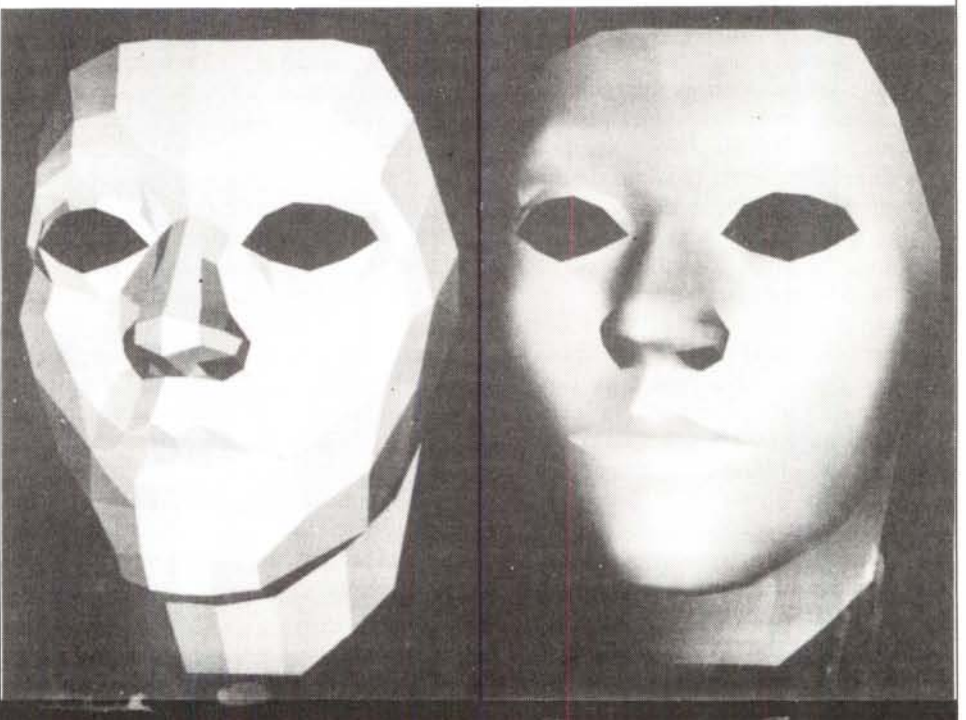
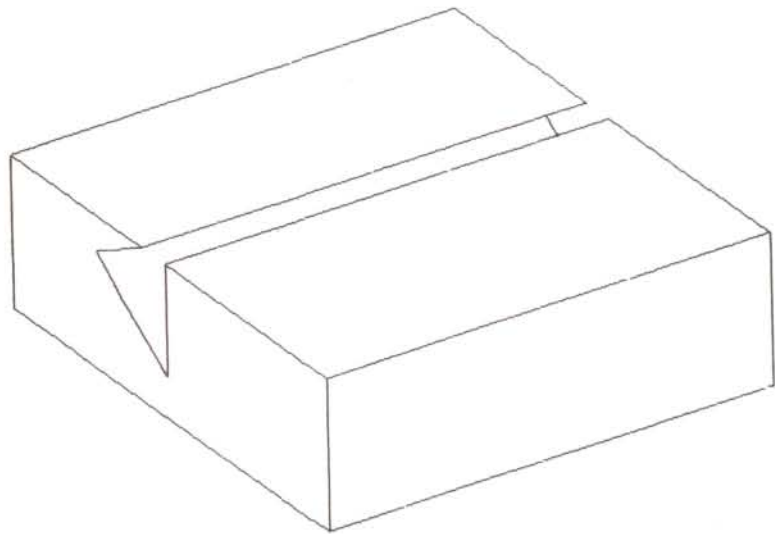
Realisme a les representacions tridimensionals

Si l'únic que es fa en un cos tridimensional és projectar-lo en dues dimensions, com hem fet a l'apartat anterior, s'obtenen les imatges de la figura 6, ja comentades. Ara bé, el dibuix és molt més clar si únicament es representa la part visible del cos i s'eliminen les parts tapades, com ja es fa habitualment en els dibuixos clàssics. La millora és evident si es compara la figura 6 amb la figura 7. Els sistemes gràfics que permeten el treball en tres dimensions contenen algorismes d'eliminació automàtica de les parts amagades per tal d'obtenir representacions realistes dels objectes.

A la bibliografia es pot trobar un nombre considerable d'algorismes per resoldre aquest problema. Aquí n'exposarem a grans trets un d'ells, basat en el sistema d'ordenació de Newell i Sancha:

a) En un primer pas, del conjunt de totes les cares de l'objecte o objectes a representar, s'eliminen totes les que miren enrera, ja que segur que no es podran veure. Aquesta decisió es pren segons el valor del producte escalar del vector normal cap enfora de la cara amb el vector que mira cap a l'observador.

b) Si només teníem un objecte i era convex, l'apartat a) soluciona totalment l'eliminació de parts ocultes. En cas contrari, s'ordenen totes les cares que queden de manera que, al final, cada cara només tapi (totalment o parcialment) les que té darrera seu a la llista de cares.



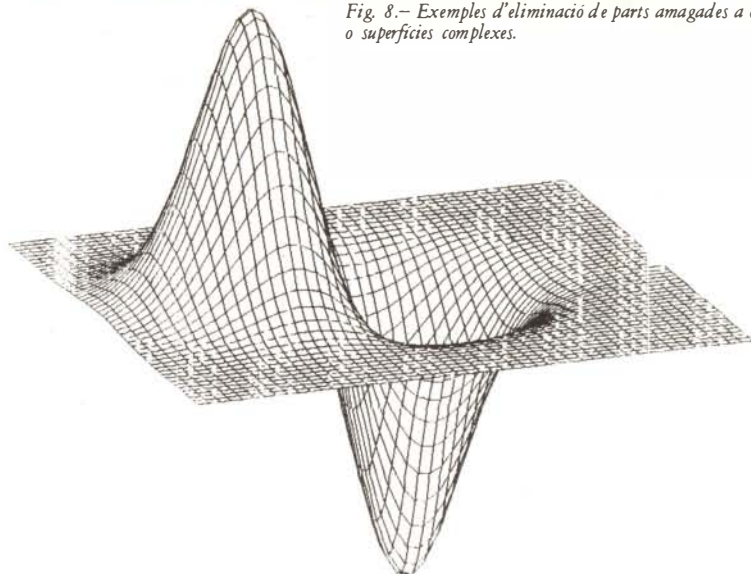
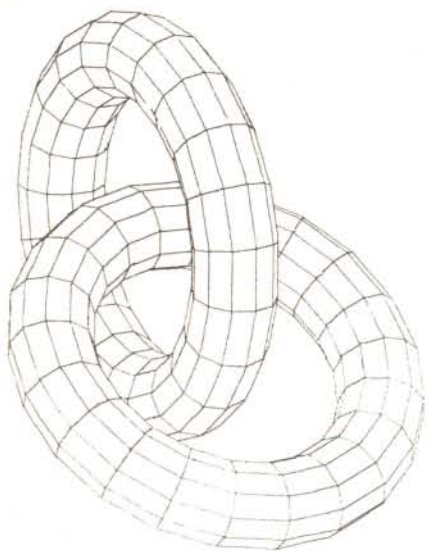
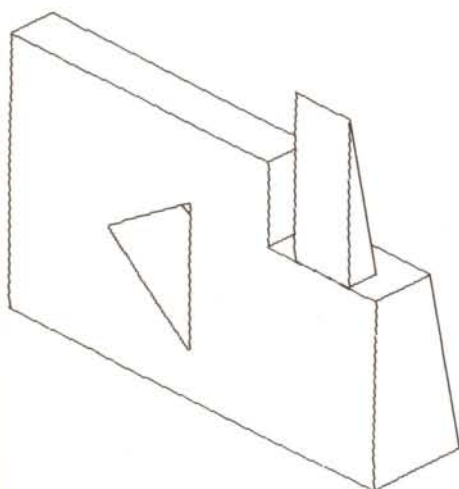


Fig. 8.- Exemples d'eliminació de parts amagades a objectes o superfícies complexes.



c) Es dibuixa la primera cara de la llista, ja que és segur que no és tapada per cap altra.

d) Cada cara C_k de la llista s'estudia en relació amb cada una de les cares anteriors, C_i amb $i = 1 \dots k-1$, (segons el que hem dit, les cares C_1, \dots, C_{k-1} són les úniques que poden tapar la C_k). Si alguna de les C_i tapa parcialment la C_k , es suprimeix la cara C_k de la llista i es deixa en lloc seu el tros que la C_i deixa veure. Si després d'haver analitzat totes les cares C_1, \dots, C_{k-1} , encara queda algun tros vist de C_k , es dibuixa (el que es dibuixa és el polígon del contorn del que queda de C_k). Sobre l'algorisme anterior, cal fer dues observacions: En primer lloc, cal tenir guardat l'objecte o objectes a representar amb estructura de cares. En altres paraules, no podem entendre un cos com un conjunt d'arestes (model que s'anomena de "filferros"), sinó que hem de saber quantes cares té el nostre cos i quines arestes té cada cara al seu contorn (model polièdric en la notació habitual). Això implica, per exemple, que la per-

sona que crea el cos hagi d'entrar cares i no un dibuix amb línies, simplement. En segon lloc, hem de dir que és un algorisme millorable en el sentit que podria ser més ràpid. En efecte, és molt fàcil comprovar que el temps de càlcul creix amb el quadrat de nombre de cares, i pot conduir a temps massa elevats quan els objectes són més complexos. Ara bé, cal dir que els algorismes més eficients són també força més complexos d'estructura.

A la figura 8 es tenen dos exemples d'eliminació de parts amagades en entorns geomètrics força complexos. La figura 9 mostra que un algorisme d'eliminació de parts ocultes junt amb un sistema que permeti interseccions i diferències de cossos, pot resoldre molts dels problemes clàssics del dibuix d'objectes: obtenir vistes de seccions del cos a base d'interseccionar-los amb el semiespai definit pel pla de tall.

Fins ara hem vist possibilitats de dibuix d'objectes tridimensionals, amb eliminació o no de les parts amagades. No obstant això, algunes vegades no es desitja un dibuix (on només es marquen les arestes que delimiten cada cara), sinó una imatge de l'objecte on cada cara tingui una tonalitat de gris o un color dependent de la seva orientació i de la posició d'un hipotètic focus de llum que il·lumina l'objecte. En aquest cas, es comença per calcular la intensitat de cada cara, que variarà entre zero (blanc) i la unitat (color negre, cas de les cares amb normal dirigida en contra de la direcció d'il·luminació). Després, un dels algorismes de representació de la imatge (encara que no és el més eficient) és "l'algorisme del pintor":

a) El primer pas és idèntic al de l'algorisme de dibuix amb eliminació de parts ocultes: s'eliminen les cares que mirin enrera.

b) El segon pas consisteix també en una

ordenació de les cares que queden de manera que cada cara tapi només les que té al seu darrera.

c) Finalment, es dibuixen totes les cares començant per la darrera de la llista i acabant per la primera (dibuixar una cara vol dir incorporar a la pantalla d'escombrat el polígon del contorn amb tot l'interior pintat amb la tonalitat de gris o color corresponent a la cara). Evidentment les cares més a prop de l'observador es dibuixaran més tard, pintant amb el seu color "damunt" de les parts de les cares més llunyanes que eren tapades per ella.

La figura 10 presenta una cara dibuixada amb aquest mètode. A la mateixa figura s'observa la millora en la qualitat de la imatge quan no s'assigna a tota la cara la mateixa tonalitat sinó que es realitza una ponderació amb els nivells de gris de les cares contigües. El resultat és que abans de la ponderació s'observa perfectament la divisió en plans de la superfície, i en canvi després no. No obstant això, l'estructura polièdrica s'observa encara en les vores de la cara.

Pere Brunet i Crosa

Materials de lectura

- W.M. Newman, R.F. Sproul: *Principles of Interactive Computer Graphics*. Mac Graw Hill, 1981.
 J.D. Foley, A. Van Dam: *Fundamentals of Interactive Computer Graphics*. Addison Wesley, 1982.
 D.F. Rogers, J.A. Adams: *Mathematical Elements for Computer Graphics*. Mac Graw Hill, 1976.
 I.D. Faux, M.J. Pratt: *Computational Geometry for Design and Manufacture*.
 M. Garcia Hoffman, X. Pueyo, E. Garcia, P. Brunet, J. Amat, F. Ferrer, J.L. Colomer, A. Jinich, F. Casasayas: Conjunt d'articles sobre Sistemes Gràfics del volum 8, núms. 47-48 de "Novattica", any 1982.
 P. Brunet, D. Ayela, I Navazo: *An Interactive Algorithm for the Generation of B Spline Surfaces*. Procs. of the ICS-83 Erlangen, Nurnberg, 1983.