

DAO

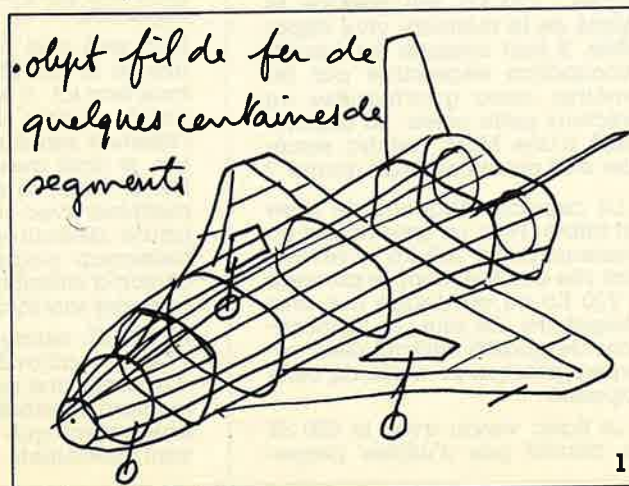


Jean-Pierre Petit a écrit une quinzaine de livres dont la série de bandes dessinées scientifiques d'Anselme Lanturiu. En mathématiques, on lui doit la découverte des équations de la surface de Boy. L'ordinateur est un outil quotidien pour ses recherches de mécanique des fluides.

LE DESSIN EN TROIS DIMENSIONS

C'est au travers d'applications spectaculaires et passionnantes comme la synthèse d'images que l'ordinateur séduira le grand public.

Avec un micro-ordinateur, on en est encore aux balbutiements prometteurs, tant sur le plan théorique que sur les problèmes de programmation. Avec cette série, Jean-Pierre Petit, maître de recherches au CNRS, fait la démonstration qu'entre ses mains expertes le micro est bien un fabuleux outil.



Rapidement, plusieurs techniques sont envisageables que nous examinerons successivement. Le dessin « fil de fer » en est la plus simple, mais on peut aussi gérer des objets constitués de facettes et éliminer les parties cachées du dessin. Là, deux techniques s'imposent : l'algorithme du peintre et l'interaction face-arête.

Dans ces deux approches, on élimine tout d'abord les faces postérieures, la première en « noircissant » les faces de la plus lointaine à la plus proche (fig. 2), la seconde, en faisant laborieusement interagir chaque arête, chaque segment, avec chaque face après un certain tri (fig. 3).

Cette seconde option conserve à chaque moment le fichier des segments progressivement amputé des parties cachées et permet donc d'attaquer *in fine* une table traçante. En revanche, l'algorithme du peintre utilise l'écran lui-même comme mémoire pour consigner le résultat du travail d'élimination. Avec une table traçante, il faudrait que celle-ci soit dotée d'une... gomme. Difficile !



SORTIES GRAPHIQUES

Le dessin sophistiqué en trois dimensions (3D) est possible sur un petit système informatique, au prix d'acrobaties d'analyse et de programmations, c'est le défi que nous relevons.

Entre l'écran et la table traçante, deux modes classiques de reproduction d'images, le second ouvre d'emblée sur la très haute résolution graphique. L'œil nu distingue difficilement des pas d'incrémentations de l'ordre du dixième de millimètre (les tables traçantes Plotter Apple, 2 500 x 1 850 points, et Watanabe possèdent cette résolution) tandis qu'un micro comme l'Apple 2 ne possède qu'une résolution d'écran de 280 x 191 points !

Les importantes résolutions d'écran sont aussi des gouffres de mémoire : de 8 Ko sur Apple 2 à 10 Mo sur système Raster en couleur. Donc n'attendons pas de pro-

grès décisif sur micro-ordinateur car la très bonne résolution d'écran multiplie au moins par dix les besoins en mémoire, et la vitesse de traitement du système doit aussi croître en proportion.

En ce qui concerne les images sur écran, une résolution de 200 x 300 points est minimale et une de 512 x 512 acceptable. Le mouvement d'une image améliore considérablement sa résolution apparente, subjective. La CAO (conception assistée par ordinateur) débute avec des résolutions de 1 024 x 1 024, mais n'est bien à son aise qu'en 1 500 x 2 000.

Alors, que faire ? Doit-on développer des logiciels capables de gérer des milliers de segments tout en sachant parfaitement qu'ils seront presque illisibles sur l'écran du micro-ordinateur (fig. 1) ? Devinez...

Nous ne nous limiterons pas arbitrairement, mais tablons d'emblée sur des ensembles de plusieurs milliers de segments. Un petit micro est capable, couplé si possible à une table traçante, de rivaliser en complexité avec les gros systèmes.



LE 3D SUR MICRO

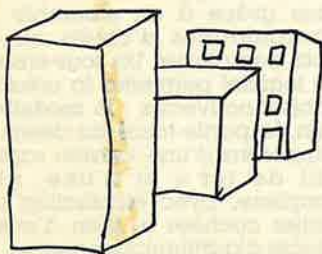
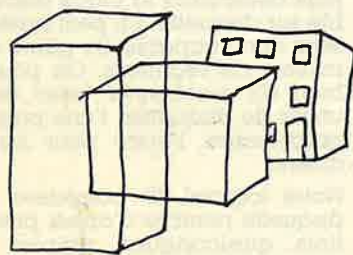
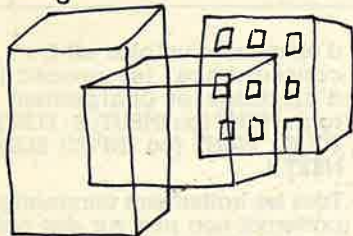
L'introduction des données se réalise la plupart du temps à l'aide d'un outil, le digitaliseur. L'utilisateur peut dessiner à main levée une plaquette avec un stylo spécial qui envoie à l'ordinateur les coordonnées X, Y de la pointe (ou du réticule). Sur les micros, comme l'Apple 2, ce digitaliseur se connecte avec une carte, et la résolution de la saisie tourne autour de 3 000 x 3 000 points.

quelques minutes avec un programme en Basic interprété. De complexes dessins, comme des vues d'ensembles architecturaux, ne demandent guère plus de trente minutes. L'image de la figure 4, avec une élimination des parties cachées utilisant le lent algorithme face-arête, a été calculée dans ces conditions en dix minutes sur Apple 2 en simple Basic. On gagnera un temps considérable avec un micro 16 bits équipé de 512 Ko de mémoire environ.

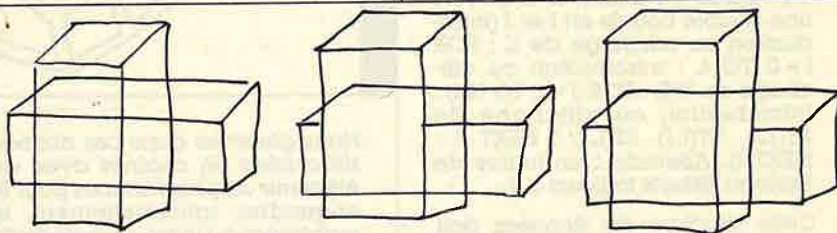
Pour commencer à construire un logiciel de dessin en trois dimensions doté de hautes performances, il faut prévoir toutes les extensions possibles du programme avant d'en débiter la réalisation.

Certaines structures de données pourraient en effet se révéler par la suite incompatibles avec des traitements nouveaux, telle l'élimination des parties cachées. Les

Algorithme du peintre

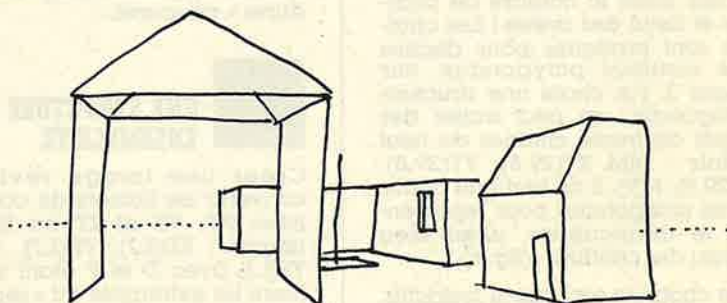


2



travail de l'algorithme face-arête

3



4

On dispose donc d'un outil sérieux d'introduction de données, mais pas encore du logiciel performant capable de gérer les outils, l'algorithme et la technique graphique, pas même dans le commerce.

Le temps de calcul n'est pas un réel problème. Sur Apple 2, machine classique à 8 bits seulement, une image « fil de fer » de trois cents segments sera calculée en

indications suivantes en tiennent largement compte.

Comment réaliser un programme de dessin « fil de fer » ? Tout d'abord, se doter d'une structure de données en mémoire d'ordinateur pour représenter un dessin dans un repère à trois dimensions X, Y et Z. La structure de base est la **chaîne**. Une chaîne se compose de quatre fichiers. Un **objet** est un

ensemble de chaînes (L+1 en l'espace), I sera l'indice de chaîne et J le pointeur sur chaîne :

L nombre de chaînes
 N(I) le nombre de points chaînés - 1
 XT(I,J)
 YT(I,J)
 ZT(I,J) sont les coordonnées des points sur la chaîne.

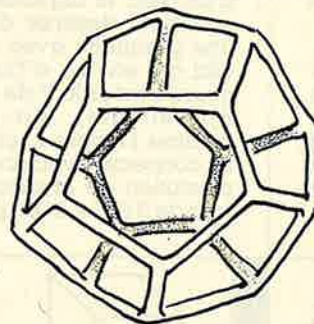
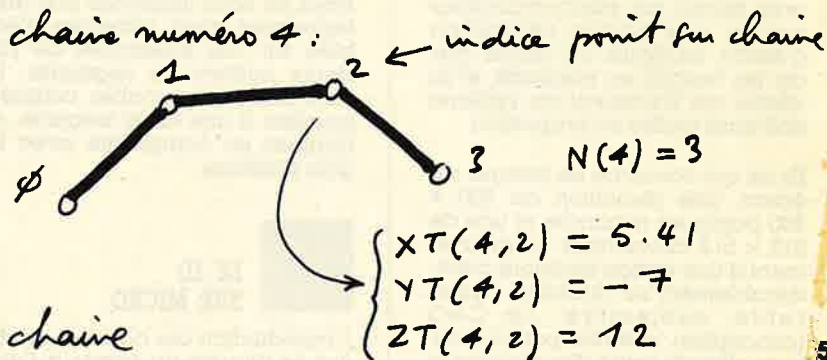
La figure 5 donne un exemple de chaîne graphique, composant une partie d'un dessin, codée dans une telle structure de données. Cette chaîne est la quatrième du dessin (I = 4), elle comporte trois segments (N(I) = 3), et donc aussi quatre points. Chacun d'eux a ses coordonnées conservées dans les tableaux XT, YT et ZT. Le point numéro 2 (indice J=2) correspond donc aux valeurs de XT(4,2), YT(4,2), ZT(4,2).

Sur le plan de la programmation, les coordonnées des points seront introduites ou affichées à travers une double boucle en I et J (introduction ou affichage de L : FOR I = 0 TO L : introduction ou affichage de N(I) : FOR J = 0 TO N(I) : introduction ou affichage de XT(I,J), YT(I,J), ZT(I,J) : NEXT J : NEXT I). Attention : un indice de tableau débute toujours à 0.

Cette structure de données doit être associée à un ordre DIM de réservation de place en mémoire pour les tableaux, mais il faut choisir entre le nombre de chaînes et celui des arêtes ! Les chaînes sont pratiques pour décrire des contours polygonaux. Sur Apple 2, j'ai choisi une structure octogonale : on peut traiter des objets de trente chaînes de neuf points : DIM XT(29,8), YT(29,8), ZT(29,8), N(9). Il ne faut que douze faces octogonales pour représenter le dodécaèdre, objet bien connu des citadins... (fig. 6).

Ces choix ne sont pas si restrictifs. Un cercle, par exemple, sera acceptablement représenté à l'aide de trente-six segments, un tous les dix degrés. L'objet **cercle** sera donc selon les cas constitué de quatre chaînes de huit segments plus un de quatre segments, ou bien de sept chaînes de cinq segments plus une d'un seul segment.

On ne gagne donc pas systématiquement en choisissant des chaînes composées de nombreux segments, au contraire, on gaspille une précieuse place mémoire.



• Dodécaèdre



Nous gérerons donc ces données structurées en chaînes avec des éléments de programmes pour les engendrer rationnellement, les manipuler à l'infini, user de toutes les transformations géométriques (translation, rotation, homothétie, affinité, symétries, etc.), de « soudures », coupures...



UNE STRUCTURE ENSEMBLISTE

Créer une image revient à convertir les fichiers de coordonnées XT, YT et ZT en fichiers image : XD(I,J), YD(I,J), XF(I,J), YF(I,J) avec D et F étant simplement les extrémités du « segment-image ».

Ces fichiers, stockés sur disquette (à ne pas confondre avec une image d'écran) formeront une image précalculée, très commode pour tracer le dessin ultérieurement.

Notre structure de données est partie de la chaîne de points pour aboutir à l'objet, ensemble de chaînes. Définissons maintenant le **bloc** comme un ensemble d'objets où chaque élément ELS(K) est une suite de caractères. Le nombre

d'éléments d'un bloc est E + 1. En conséquence, les procédures d'affichage et chargement seront : PRINT (ou INPUT) E : FOR K=0 TO E : PRINT (ou INPUT) ELS(K) : NEXT K.

Tous les traitements informatiques porteront non plus sur des objets, mais sur les blocs entiers de E + 1 éléments. On pourra en gérer une trentaine sur Apple 2 : la seule limite demeurant la place disponible sur disquette. Un petit système sera donc capable de gérer des milliers de segments. On pourra bien sûr fonctionner avec deux unités de disquettes, l'une pour le programme, l'autre pour les fichiers.

Notre logiciel 3D possédera sur disquette nombre d'objets prédéfinis, quelconques, simples ou complexes. On pourra les positionner n'importe où dans l'espace grâce à un ensemble de manipulations d'objets ou de blocs complexes. Un sous-ensemble logiciel permettra la création d'objets nouveaux : le **modèleur**. Enfin, la partie tracé du dessin se composera d'une version rapide « fil de fer » et d'une, plus complexe, avec élimination des parties cachées et bien d'autres astuces algorithmiques encore. ■