

THESE

PRESENTEE

A L'ECOLE NATIONALE SUPERIEURE

DE TECHNIQUES AVANCEES

POUR OBTENIR

LE DIPLOME DE DOCTEUR INGENIEUR

SPECIALITE : MATHEMATIQUES

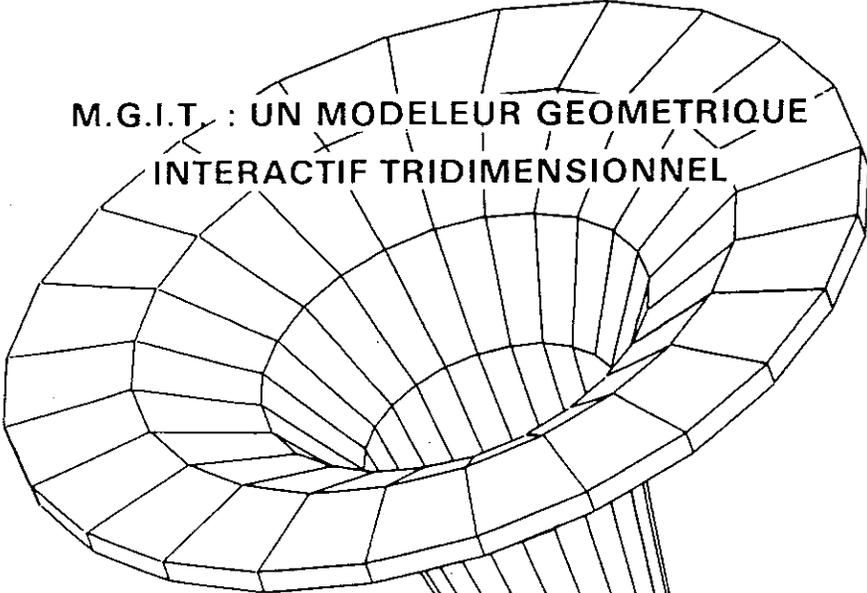
MENTION : INFORMATIQUE

PAR

Monsieur Jose Antonio GARRIDO NATAREN

Sujet de la thèse

**M.G.I.T. : UN MODELEUR GEOMETRIQUE
INTERACTIF TRIDIMENSIONNEL**



Soutenu le 29 juin 1981 devant le jury composé de :

M. ROBINET
M. BELLEVAUX
M. BRUN
M. GIRALT
M. PISTENON

Président

Examineurs

Je ne saurais entreprendre l'exposé de mon travail sans remercier tous ceux qui m'ont fait bénéficier de leurs enseignements, de leur expérience, ou simplement soutenu de leur amitié.

Je remercie Monsieur ROBINET de m'avoir fait l'honneur d'accepter la présidence du jury.

Monsieur BELLEVAUX a toujours porté une attention particulière à mes travaux.
Pour ses encouragements, sa patience, ses critiques constructives, je tiens à lui exprimer ma très vive gratitude.

Je tiens à remercier Monsieur BRUN et Monsieur GIRALT pour l'intérêt qu'ils ont porté à ce travail en acceptant de siéger à ce jury.

J'exprime ma reconnaissance à Monsieur PISTENON pour ses remarques et ses fructueux conseils.

J'aimerais bien remercier chacune des personnes avec qui j'ai eu l'occasion de travailler au sein du groupe CPAO.

Mais,

la tradition voulant que le nombre de pages consacré aux remerciements soit notablement plus restreint que le nombre de pages consacré à la thèse.

Je me vois contraint de les remercier autrement :

Merci à tous.

SOMMAIRE

	pp
I - <u>INTRODUCTION</u>	1
II - <u>MODELES GEOMETRIQUES : REPRESENTATION ET</u> <u>VISUALISATION</u>	4
2.1 - Définition	5
2.2 - Représentation spatiale	6
2.2.1 - Énumération spatiale	6
2.2.2 - Décompositions récursives	7
2.2.3 - Fonction de densité	8
2.3 - Représentation de frontière	9
2.3.1 - Représentation avec des surfaces	10
2.3.2 - Représentation polyédrique	12
2.4 - Historique de création	14
2.5 - Caractérisation visuelle d'un modèle polyédrique	16
2.5.1 - Principes de visualisation	16
2.5.2 - Visibilité des faces, arêtes et sommets	17
III - <u>CONSTRUCTION INTERACTIVE DE VOLUMES</u>	21
3.1 - Saisie d'un contour	23
3.1.1 - Entrée d'un segment	24
3.1.2 - Entrée d'un arc de cercle	25
3.1.3 - Entrée d'un cercle	25
3.1.4 - Définition d'une spline	26

p		pp
8	4.5 - Complexité de l'algorithme	68
8	4.6 - Conclusion	68
9		
0	V - <u>ALGORITHMES GEOMETRIQUES EFFICACES</u>	73
1	5.1 - Test minimax	74
1	5.2 - Comparaison segment-segment	76
2	5.3 - Comparaison point-polygone	79
5	5.4 - Tri par ventilation	84
7	5.5 - Algorithme d'intersection de segments	89
0	5.6 - Algorithme de comparaison de points et polygones	94
4	VI - <u>CONCLUSION</u>	98
6	VII - <u>BIBLIOGRAPHIE</u>	100
7		
8		
8		
8		
9		
3		
6		
9		
1		

	pp
3.2 - Fonctions disponibles	28
3.2.1 - Agrandissement-retrécissement	28
3.2.2 - Rotation	29
3.2.3 - Translation	30
3.2.4 - Changement d'échelle	31
3.2.5 - Modification des paramètres de visualisation	31
3.3 - Prismes	32
3.4 - Modèles de révolution	35
3.5 - Modèles combinés	37
3.6 - Vérification du modèle	40
 IV - <u>ALGORITHME DE VISUALISATION DE POLYEDRES AVEC</u> <u>ELIMINATION DES PARTIES CACHEES</u>	 44
4.1 - Rappel des algorithmes	46
4.2 - Algorithme	47
4.3 - Description du modèle	48
4.4 - Description des fonctions de transition	48
4.4.1 - Normalisation	48
4.4.2 - Elimination des faces, arêtes et sommets cachés	49
4.4.3 - Découpage des arêtes	53
4.4.4 - Ensemble naturel des sommets	56
4.4.5 - Ensemble naturel du premier segment d'une arête	59
4.4.6 - Parcours des arêtes	61

CHAPITRE I

INTRODUCTION

"Si j'avais appris la technique, je serais technicien. Je fabriquerais des objets compliqués. Des objets très compliqués, de plus en plus compliqués, cela simplifierait l'existence."

IONESCO

Dans un contexte de C.A.O. l'outil informatique est utilisé de façon classique pour sa puissance de calcul et ses possibilités de traitement de données.

Cependant, tous ces avantages potentiels peuvent se déprécier du fait d'une saisie de l'information par trop difficile ou d'une interprétation des résultats trop lente.

Dans le cas d'un système de modélisation tridimensionnelle dont l'objet est l'aide à la création de formes volumiques, la construction et la compréhension de cette forme est une opération suffisamment compliquée pour aboutir à une annihilation complète de l'esprit de création. De la qualité de l'interface homme-machine, de l'ergonomie du poste de travail et de la rapidité des rendus dépend le succès d'un tel système.

Parmi les actions demandant une réponse rapide notons :

- tous les calculs géométriques capables de traduire les actions de l'opérateur dans des relations topologiques et des valeurs numériques définissant le modèle.
- toutes les transformations géométriques inhérentes aux opérations topologiques (union, intersection etc..) appliquées aux modèles.
- toutes les représentations d'objets; l'utilisateur doit pouvoir juger rapidement de la qualité de l'objet construit.

Par ailleurs, il est nécessaire de préserver et de restituer dans toute leur intégrité les données et peut être même les actions de l'opérateur qui ont permis la création d'un modèle.

L'ensemble des exigences à satisfaire pose des problèmes algorithmiques et informatiques dont la résolution est non triviale.

Ainsi que nous le constatons dans certains systèmes actuellement opérationnels, les modèles définis par le concepteur sont représentés en général par des polyèdres; cette approximation est conservée dans une structure de données où elle sera traitée par des algorithmes généraux. Cette stratégie est peu conciliable avec la préservation des données utilisateur. En outre, elle demande, pour obtenir un temps de réponse acceptable du fait de la non spécification des algorithmes, une puissance de calcul non négligeable.

Nous allons présenter comme alternative un Modeleur Géométrique Interactif Tridimensionnel où les formes volumiques sont créées moyennant un dialogue naturel qui permet une obtention rapide des résultats. Dégagé des contraintes d'un dialogue sophistiqué, l'opérateur pourra tirer profit de toutes les possibilités du système et ainsi donner libre cours à son processus créatif.

CHAPITRE II

MODELES GEOMETRIQUES :

REPRESENTATION ET VISUALISATION

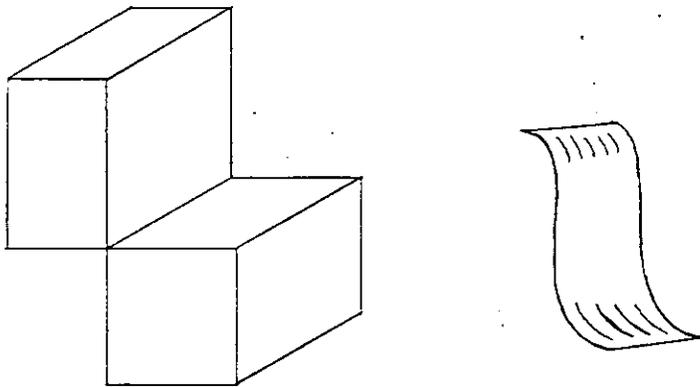
2 - REPRESENTATION

2.1 - Définition

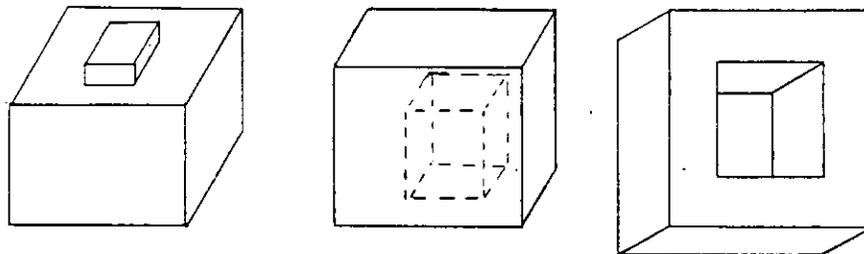
Un modèle géométrique est la représentation, dans un ordinateur, d'un objet solide.

Cette représentation est restreinte aux objets ayant les propriétés physiques suivantes :

- L'objet est en un seul morceau.
- L'objet n'a pas un volume infini et contient sa forme (fluides exclus).
- L'intérieur de l'objet est non vide et en un seul morceau.
- L'objet a une épaisseur.



Objets non représentables

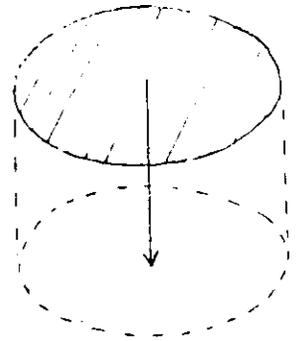
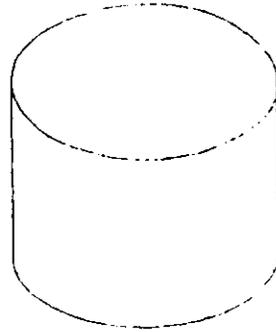
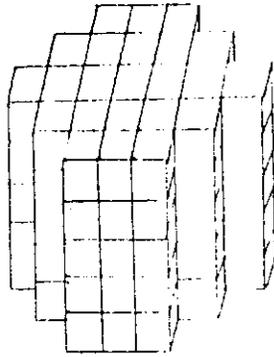


Objets représentables

Figure 2.1

Les différents types de représentations élémentaires peuvent être classés dans 3 groupes :

- 1 - Représentation spatiale
- 2 - Représentation de frontière
- 3 - Historique de création.



$M = (c1, c2, c3, c4, \dots)$

$M = (face1, face2, \dots)$

$M = (face 1, vecteur$

représentation

représentation

historique de

spatiale

de frontière

création

a

b

c

Figure 2.2

2.2 - Représentation spatiale

2.2.1 - Énumération spatiale

L'énumération spatiale est la plus simple représentation d'un modèle. On décompose \mathbb{R}^3 en volumes élémentaires disjoints. Bien que ces volumes élémentaires soient souvent des cubes, des décompositions utilisant d'autres polyèdres peuvent être envisagés : tétraèdres, parallélépipèdes etc.. Un modèle est alors représenté par l'ensemble des volumes élémentaires qu'il contient (voir fig. 2.2a).

Cette représentation offre l'avantage de pouvoir accéder directement aux volumes élémentaires (points) et il est ainsi très simple de savoir si un point appartient ou non au modèle. Ceci facilite largement la tâche des opérateurs volumiques d'union, intersection, différence et le calcul des centres de gravité, volumes etc..

Un inconvénient de cette représentation est le nombre de données nécessaires pour aboutir à une approximation précise du modèle. En effet, plus la précision requise est grande, plus la taille des volumes élémentaires doit être petite et plus le nombre de données nécessaires augmente.

En outre, cette représentation est redondante : tous les volumes élémentaires adjacents à un volume lui-même contenu dans le modèle seront aussi contenus par ce dernier sauf si le volume appartient à la frontière du modèle.

2.2.2 - Décompositions récursives

Pour représenter un modèle M par la méthode des décompositions récursives, on considère premièrement \mathbb{R}^3 comme étant un volume élémentaire (cellule cubique). S'il n'est pas homogène, le volume élémentaire est décomposé en deux, quatre ou huit nouveaux volumes élémentaires homogènes. Un volume élémentaire V_e est considéré homogène s'il satisfait à un des cas suivants :

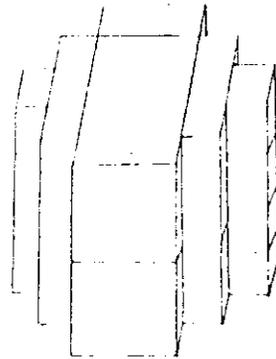
1 - $V_e \cap M = V_e$

2 - $V_e \cap M = \emptyset$

3 - $|V_e| \leq E$

En d'autres termes, le volume élémentaire est considéré homogène s'il est entièrement contenu dans le modèle, ou complètement extérieur ou si la taille du volume élémentaire est inférieure à une valeur ϵ arbitrairement fixée.

Le modèle M est ainsi représenté par l'ensemble des volumes élémentaires homogènes qui satisfont à la condition 1 ci-dessus.



Représentation du modèle de la fig. 1 par la méthode des décompositions récursives

Figure 2.3

2.2.3 - Fonction de densité

Une autre représentation spatiale d'un modèle peut être obtenue à l'aide de fonctions tridimensionnelles de densité

$$M = F (X, Y, Z)$$

Cette représentation requiert couramment une décomposition du modèle en morceaux qui peuvent être définis par une fonction de ce type mais ceci n'est pas toujours évident.

2.3 - Représentation de frontière

Un modèle possédant les propriétés décrites au § 2.1 peut être représenté par sa frontière. La surface (frontière) du modèle est définie par un ensemble de morceaux de surfaces appelés faces.

L'intersection des faces sont des lignes appelées arêtes et l'intersection des arêtes sont des points appelés sommets.

Les données définissant les faces, arêtes et sommets sont couramment appelées représentation géométrique et celles représentant les relations d'incidence appelées représentation topologique.

La représentation des frontières doit, pour être correcte, satisfaire aux conditions suivantes :

- a) La frontière définie par l'ensemble des faces doit être fermée. Ceci implique que toutes les arêtes soient adjacentes à deux et seulement deux faces.
- b) La frontière ne doit pas s'intersecter elle-même. L'intersection entre deux faces, si elle existe, ne peut avoir lieu qu'à leurs frontières (arêtes ou sommets).
- c) La frontière doit être orientable. La satisfaction des conditions a et b ci-dessus assure cette condition. La frontière sépare ainsi \mathbb{R}^3 en deux portions, la première, bornée, contenant la frontière et correspondant au modèle, l'autre correspondant au reste de l'univers.

d) La frontière doit être connexe. Ceci suppose qu'il existe toujours sur la frontière un chemin joignant deux points. Cette condition n'exclut pas l'existence de cavités à l'intérieur du modèle.

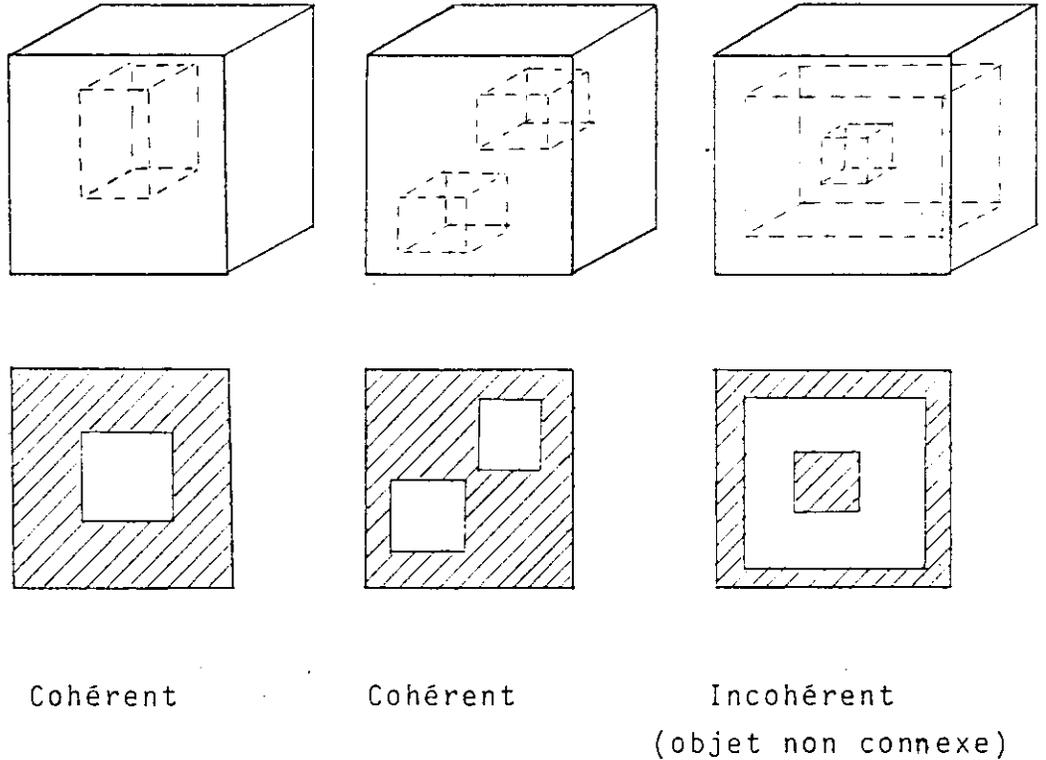


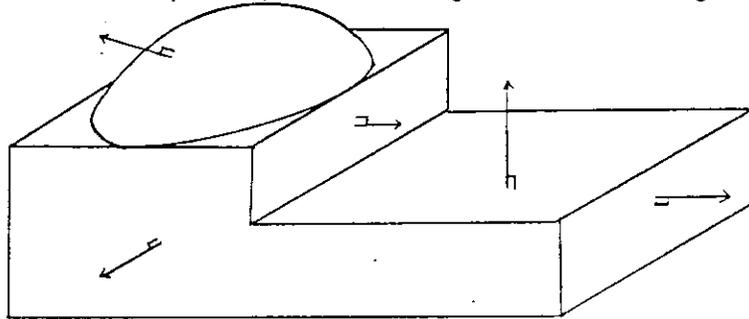
Figure 2.4

2.3.1 - Représentation avec des surfaces

La représentation des frontières offre l'avantage d'être une représentation exacte et compacte. Cependant, elle se prête mal aux tests d'appartenance pour savoir en effet si un point de l'espace appartient ou non au modèle, un algorithme est nécessaire et les calculs qui en découlent peuvent être compliqués.

Il en est de même pour les algorithmes d'union-in-

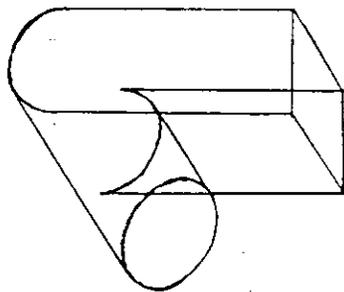
tersection-différence et les algorithmes de visualisation. La connaissance de vecteurs normaux orientés vers l'intérieur (ou vers l'extérieur) simplifie largement ces algorithmes.



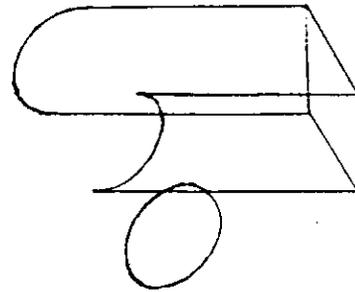
Vecteurs normaux sortants

Fig. 2.5

Par ailleurs, la détermination des arêtes dans le cas des faces tangentes n'est pas toujours évidente. Dans les cas ambigus des algorithmes capables de décomposer ces faces en d'autres faces plus élémentaires et d'ajouter des arêtes fictives sont nécessaires.



Faces courbes tangentes

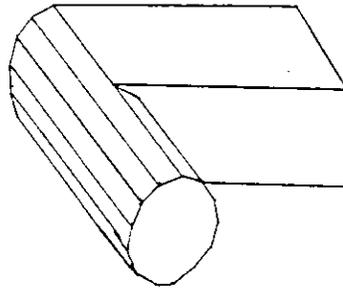


Représentation ne faisant pas intervenir des arêtes fictives

Fig. 2.6

2.3.2 - Représentation polyédrique

Les inconvénients présentés par la représentation ci-dessus peuvent être éliminés en approximant les faces par un ensemble de faces planes et les arêtes par des segments de droite, on obtient alors une représentation polyédrique.



Représentation polyédrique

Figure 2.7

La représentation topologique définit les relations d'incidence entre les différents composants du modèle : sommets, arêtes et faces. Dans le cas d'une représentation polyédrique il existe 9 relations d'incidence.

Deux de ces relations permettent de représenter entièrement la topologie du modèle. Les relations restantes, moyennant des calculs toujours compliqués, peuvent donc être obtenues à partir de celles qui sont déjà connues. La connaissance de toutes les relations fournit un accès direct aux données; mais cette représentation en plus d'être redondante demande une place mémoire importante. On retrouve ainsi le compromis PLACE MEMOIRE/VITESSE D'ACCES dans la représentation

topologique d'un modèle.

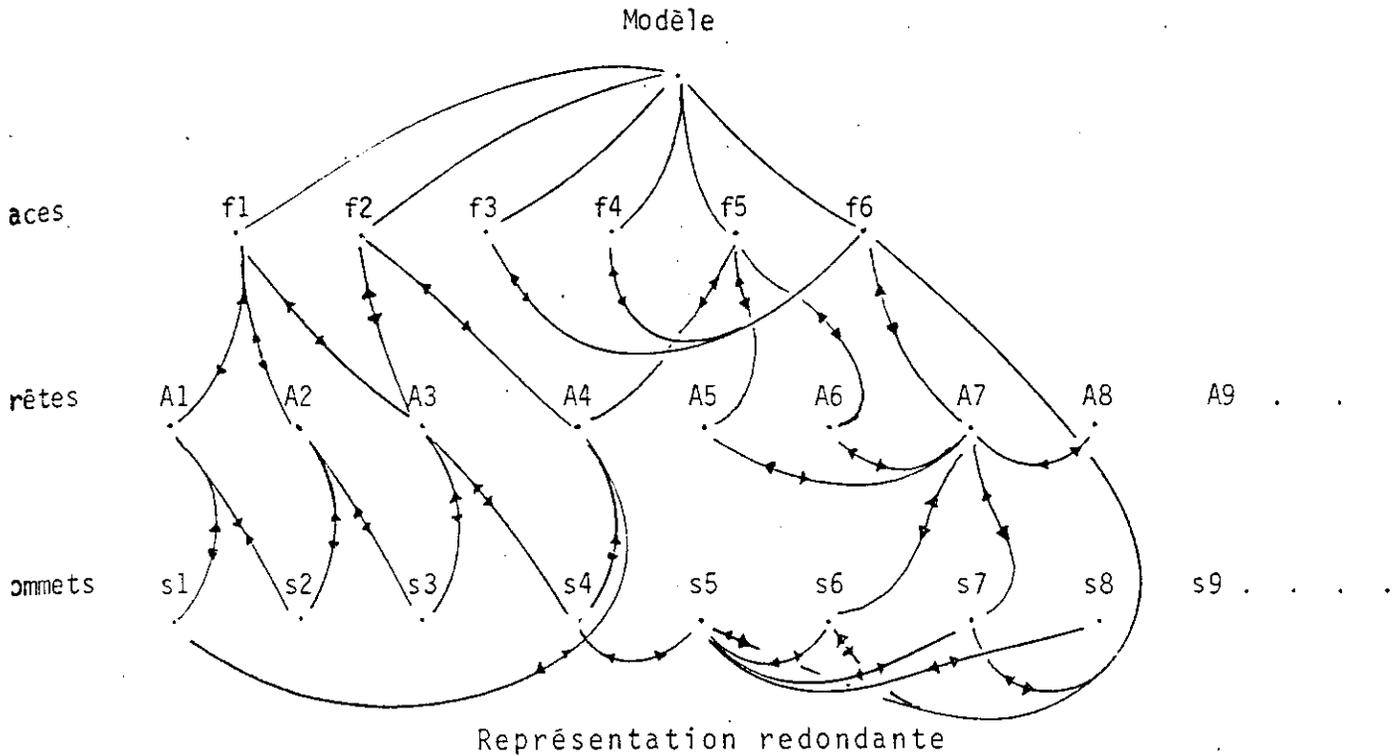


Figure 2.8

Evidemment, les critères qui déterminent le meilleur rapport place mémoire/vitesse d'accès ne sont pas les mêmes pour tous les algorithmes. Certains algorithmes privilégient la vitesse d'accès aux données aux dépens de la place mémoire tandis que pour d'autres le temps de réponse rapide n'est pas essentiel. Du fait de cette différence, un système devra conserver une représentation topologique générale qui servira à créer pour chaque algorithme des représentations intermédiaires optimales.

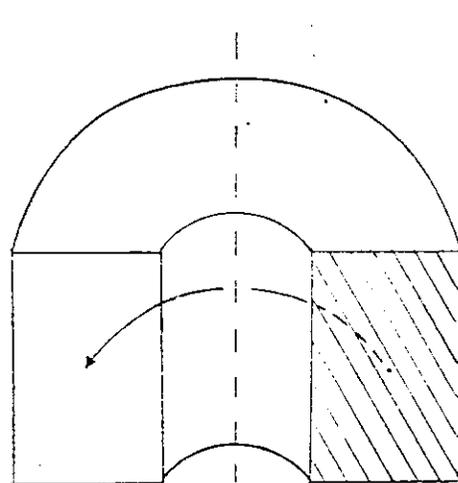
La représentation polyédrique offre l'avantage d'être d'une manipulation facile (le vecteur normal à une face est constant). Elle permet d'obtenir des algorithmes

de visualisation ayant de bonne performances. Mais de cette approximation des surfaces gauches par des faces planes ne résulte qu'un modèle approximatif insuffisant dans certaines applications (par exemple pour l'usinage).

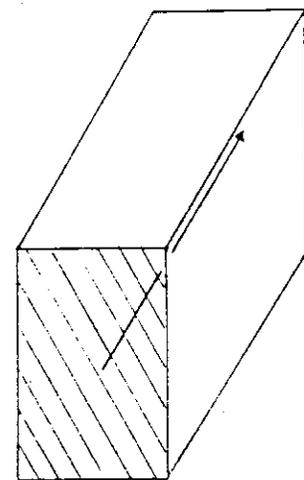
2.4 - Historique de création

Dans cette représentation, les volumes sont considérés comme des enveloppes de la trajectoire d'un contour plan.

La trajectoire peut être une courbe plane ou gauche. En particulier si cette trajectoire est un arc de cercle on obtient une section de révolution; si elle est un segment de droite on obtient une section prismée.



Déplacement suivant un
arc de cercle



Déplacement suivant une
ligne droite

Figure 2.9

Cette représentation se compose :

- d'un contour initial *cl*.

- d'une trajectoire V
- d'une fonction $K(S)$ d'abscisse curviligne le long de V , permettant de déformer le contour initial

$$M = (\mathcal{C}, V, K(s))$$

Cette représentation est difficilement utilisable par les algorithmes de visualisation, union, intersection. Par ailleurs, il est difficile de tester l'appartenance d'un point au volume engendré.

Cette représentation requiert des vérifications afin d'assurer la cohérence nécessaire à certaines applications; le déplacement du contour risque dans certains cas de balayer plusieurs fois une même portion de l'espace (modèle s'intersectant lui même).

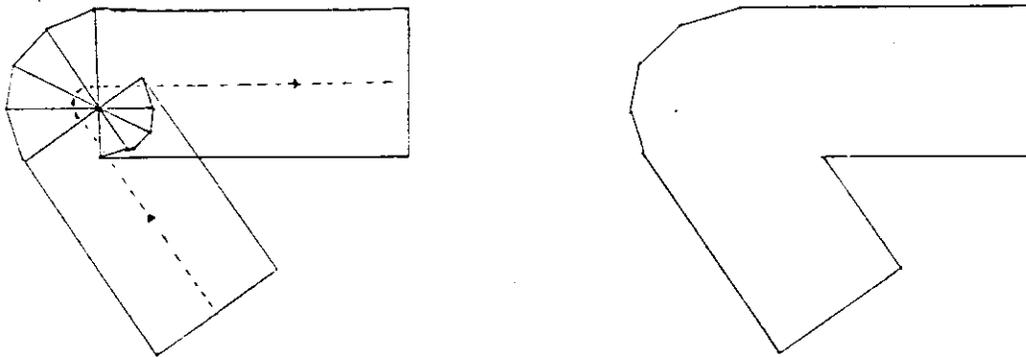


Figure 2.10

Pour l'usinage, le modèle ci-dessus est pratique car balayer une même portion de l'espace deux fois n'affecte pas les résultats, la situation se complique pour d'autres applications où il faut assurer que le centre de courbure de la trajectoire reste hors du modèle.

2.5 - Caractérisation visuelle d'un modèle polyédrique

Le modèle polyédrique présente un intérêt spécial pour la visualisation, il suffit de projeter l'ensemble des arêtes sur un plan (écran) pour obtenir l'image du modèle sous une représentation dite fil de fer. En outre, du fait le caractère plan de ses faces (vecteur normal constant pour tous les points d'une face sauf à la frontière) c'est la représentation la plus appropriée (du point de vue du temps de réponse) pour obtenir une visualisation du modèle avec élimination des parties cachées.

2.5.1 - Principes de visualisation

On considère le modèle situé dans un espace tridimensionnel (x, y, z) et le plan xy comme plan de projection.

Le modèle est visualisé avec une projection orthographique pour un observateur P_v placé à l'infini de l'axe Z , le rayon visuel V_v étant ainsi parallèle à l'axe Z et de sens contraire.

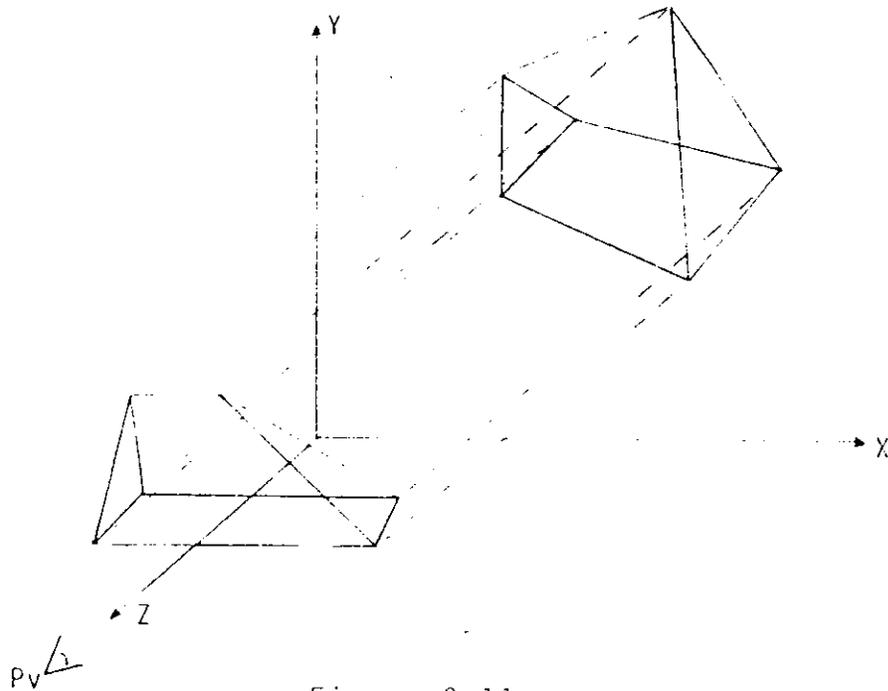


Figure 2.11

2.5.2 - Visibilité d'une face

Soit M un point intérieur à une face (non situé sur la frontière); l'intersection du rayon visuel \overline{VvM} avec la frontière du modèle peut comprendre zéro ou plusieurs segments ouverts.

Dans le premier cas, le point M sera visible mais on ne peut affirmer que cette propriété reste vraie pour tous les points intérieurs à la face. Une telle face sera dite potentiellement visible ou pour abrégé visible.

Dans le second cas on note $\overline{MM'}$ le segment ayant M pour extrémité. Deux alternatives peuvent alors se présenter :

- $\overline{MM'}$ appartient au modèle, alors le point M est caché. Par continuité due à la planarité des fa-

ces et de la cohérence du modèle, cette propriété reste valable pour tous les points intérieurs à la face. Ce type de face sera appelé face arrière.

- $\overline{MM'}$ n'appartient pas au modèle, le point M est caché mais cette propriété n'est peut-être pas vraie pour tous les points de la face; nous avons donc une face potentiellement cachée ou ce qui revient au même potentiellement visible comme dans le premier cas.

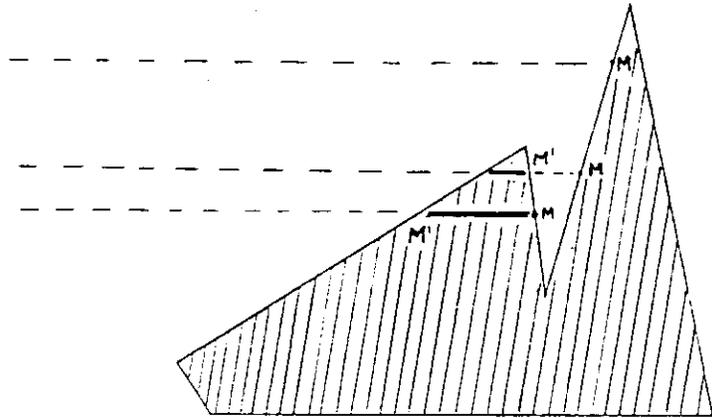


Figure 2.12

Pour déterminer les faces arrière il suffit donc de calculer le produit scalaire du vecteur de vue avec le vecteur normal V_v à la face.

Si le vecteur normal pointe vers l'extérieur de l'objet et le produit scalaire est positif la face en question est une face arrière.

Par continuité si deux faces adjacentes à une arête sont arrière, l'arête sera cachée. Pour s'en convaincre il

suffit "d'arrondir le corps au voisinage de l'arête" et de passer à la limite.

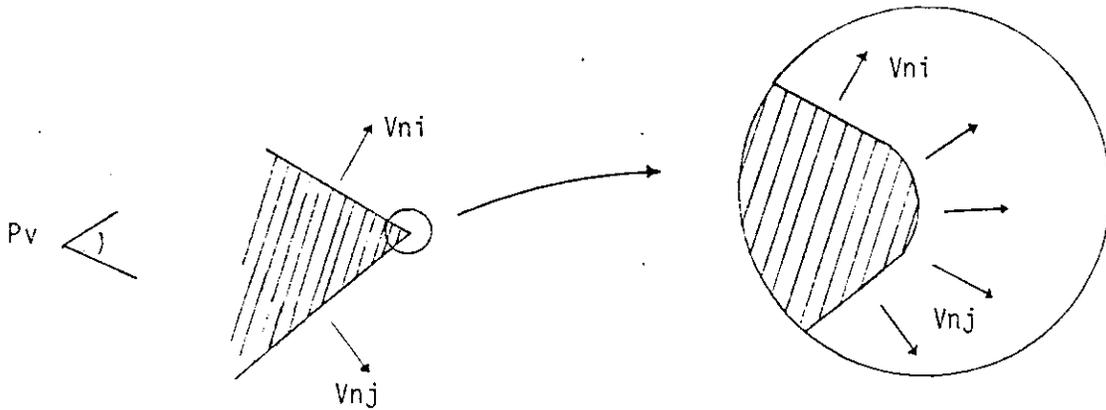


Figure 2.13

Définition

Une arête est une arête arrière si les deux faces qui lui sont incidentes sont des faces arrière.

Le même critère peut être appliqué aux sommets.

Définition

Un sommet est un sommet arrière si toutes les faces (ou toutes les arêtes) qui lui sont incidentes sont de faces (arêtes) arrière.

Si une arête appartient à une face arrière et une seule, alors pour tout point M appartenant à cette arête le rayon visuel est "tangent" au modèle.

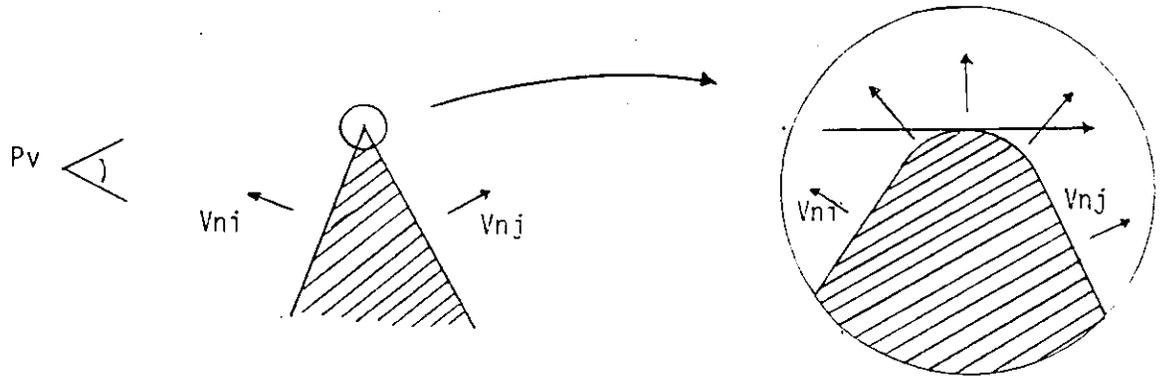


Figure 2.14

Définition

Une arête de contour est une arête ayant une et seulement une face arrière adjacente.

L'ensemble des arêtes de contour forme des polygones non planaires appelés contour apparent.

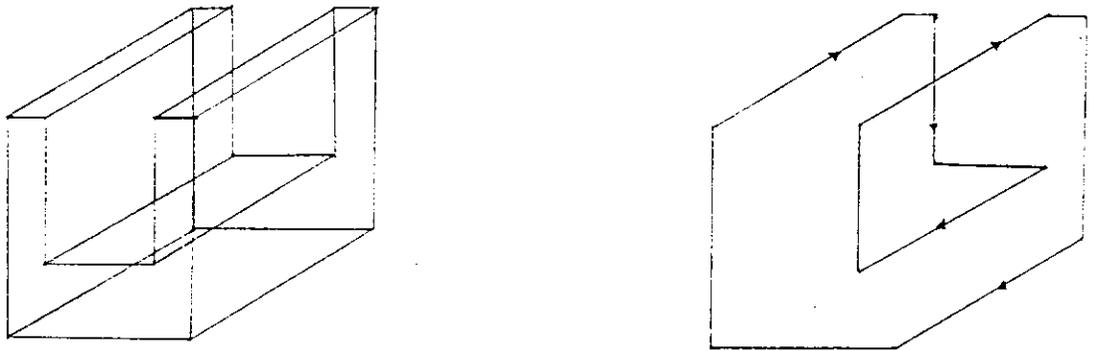


Figure 2.15

CHAPITRE III

CONSTRUCTION INTERACTIVE

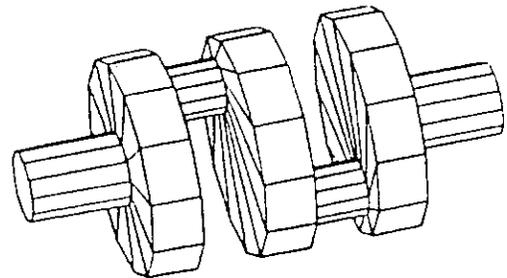
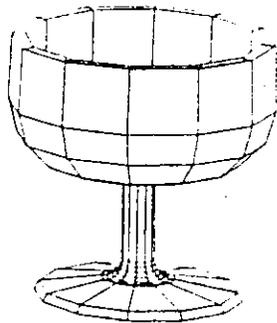
DE
VOLUMES

" L'art le plus complet est
celui de construire. "

P. VALERY

3. Le système MGIT permet dans une première phase de créer des volumes simples : prismes, corps de révolution, tuyaux etc. d'une façon interactive, de les observer sous différents angles puis de les compléter jusqu'à obtenir la forme cherchée. Une visualisation immédiate avec élimination des parties cachées permet de vérifier la forme attendue et la cohérence du modèle.

Les fonctions de mise à l'échelle, rotation et translation offrent à l'opérateur la possibilité de manipuler l'objet pendant toute la phase de construction.



Différents types de modèles créés par MGIT

Figure 3.1

Ces volumes sont définis à partir d'un contour plan non forcément fermé décrit comme une suite de segments de droite, d'arcs de cercle et de splines jointifs.

Les données du contour peuvent être saisies à l'aide d'une tablette graphique.

Une visualisation immédiate des éléments saisis permet d'apprécier l'avancement de la phase de création.

Par la suite on utilisera un trièdre de référence (ox, oy, oz), défini de la façon suivante : (ox, oy) est le plan de l'écran avec oy vertical, ox horizontal et l'axe oz perpendiculaire à l'écran dirigé vers l'observateur (voir figure ci-dessous).

Le plan de la tablette est toujours correspondant au plan de l'écran.

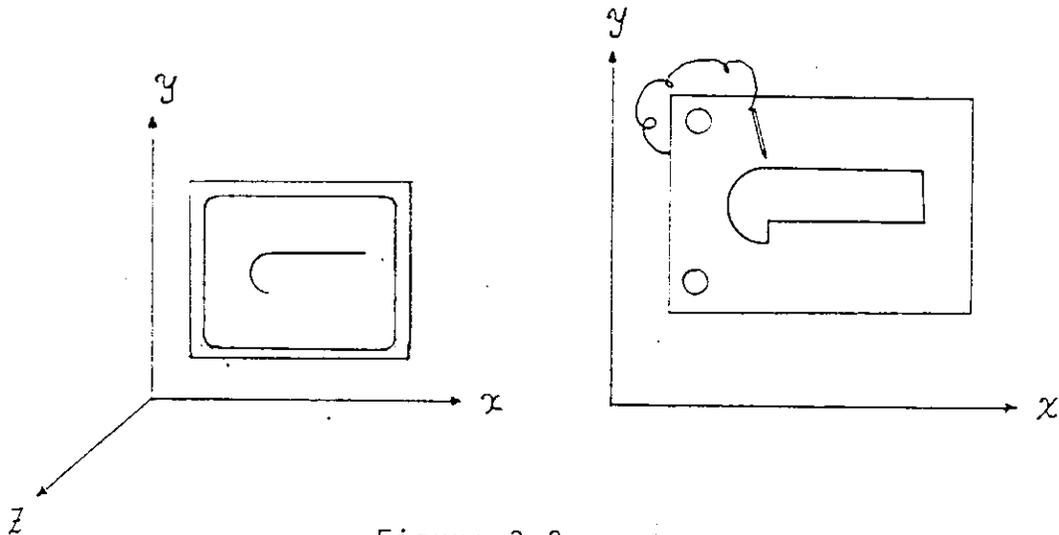


Figure 3.2

3.1 - Saisie d'un contour

L'identification des différents éléments du contour (segment, arc de cercle, spline) se fait d'une façon implicite. L'opérateur n'ayant pas besoin d'un menu pour les définir concentre toute son attention sur la forme à entrer.

Le type d'élément se déduit du nombre de points rentrés N et du laps de temps Δt , appelé temps-limite, écoulé après la saisie d'un point.

Type = F (N, Δt)

Sachant qu'un segment, arc de cercle ou spline est entièrement déterminé par deux, trois ou plus de trois points distincts respectivement, le système obtient une information numérique qui lui permettra d'identifier le type d'élément à un instant donné. Le temps-limite fixe le délai de réception d'un nouveau point, il permet au système de savoir quand il doit déclencher la reconnaissance de l'élément qui vient d'être saisi.

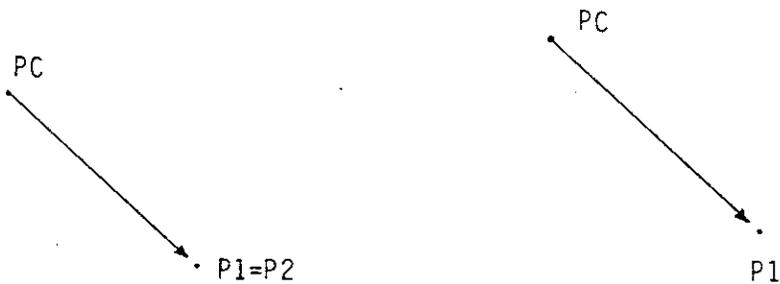
Une demande de reconnaissance peut être donnée au système en entrant un point confondu avec le dernier point de l'élément.

Le dernier point de chaque élément qui devient le premier point de l'élément suivant est appelé point courant. Les points qui servent à définir les parties intermédiaires des éléments de deuxième degré : cercle et plines, seront appelés points intermédiaires. Le point courant du premier élément est le premier point entré au début de la saisie du contour.

3.1.1. - Entrée d'un segment

Un segment de droite est identifié par une des deux éventualités suivantes :

- a) On entre deux points intermédiaires P1 et P2 confondus.
- b) On entre un point intermédiaire P1 et le système lance la reconnaissance lorsque le temps-limite est écoulé.

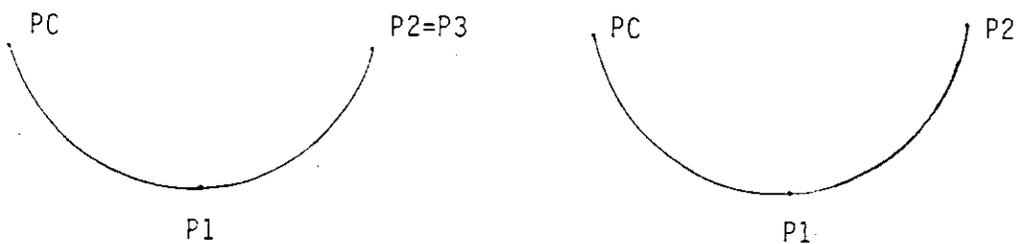


Identification d'un segment

3.1.2 - Entrée d'un arc de cercle

Un arc de cercle est identifié si une des deux éventualités suivantes se présente :

- a) On entre trois points intermédiaires P1, P2 et P3, P2 et P3 étant confondus.
- b) On entre deux points intermédiaires P1 et P2 distincts et la reconnaissance est faite lorsque le temps-limite est écoulé.

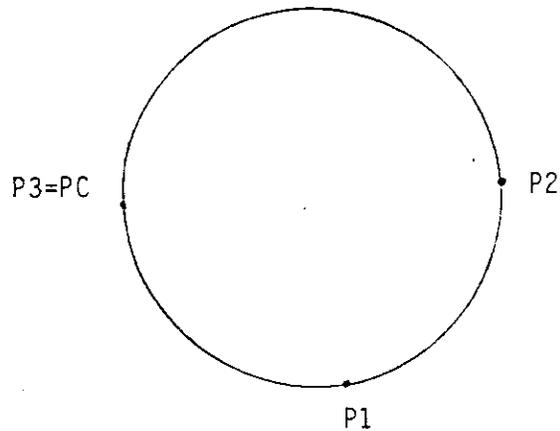


Identification d'un arc de cercle

3.1.3 - Entrée d'un cercle

Le cercle constitue en lui-même un contour. Il ne pourra être identifié que si c'est le premier élément du con-

tour que l'on entre et si le système recupère trois points intermédiaires P1, P2 et P3 distincts, dont P3 confondu avec le point courant.

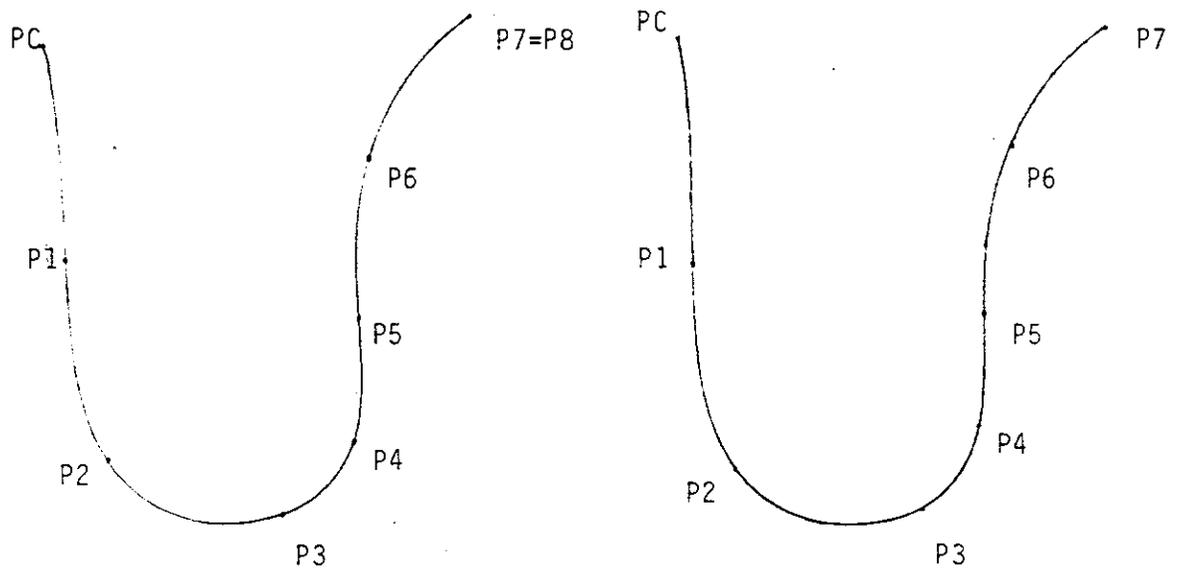


Identification d'un cercle

3.1.4 - Définition d'une spline

De même que dans les cas précédents, une spline sera caractérisée par :

- un ensemble de $n+1$ points $P_1, P_2, \dots, P_n, P_{n+1}$
 P_n et P_{n+1} étant confondus avec $n > 3$
- ou par un ensemble de n points distincts P_1, P_2, \dots, P_n avec une absence d'entrée pendant le temps-limite

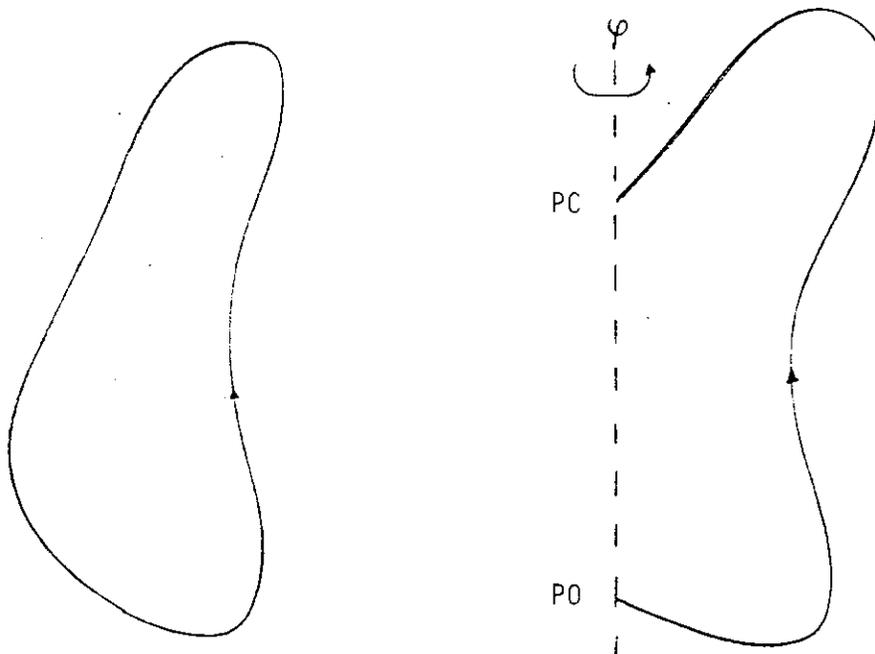


Identification d'une spline

a

La saisie de données du contour sera terminée soit par une demande de création d'un modèle de révolution, soit par retour au point initial (contour fermé).

....
ite



3.2 - Fonctions disponibles

Au cours de la conception, les options suivantes sont accessibles :

- Agrandissement-retrécissement
- Rotation
- Déplacement
- Modification des paramètres
- Redéfinition de l'horizontale du plan
- Redéfinition du point courant Pc
- Redéfinition de l'échelle du plan support

3.2.1 - L'agrandissement-retrécissement permet à tout instant de changer l'échelle et de visualiser les éléments déjà saisis; on pourra ainsi soit les voir tous, soit privilégier ceux qui semblent les plus intéressants.

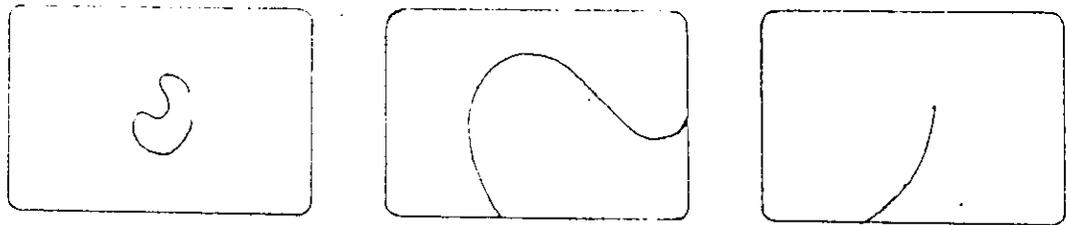


Figure 3.3

Par la suite, on appellera centre du contour le centre du rectangle d'encombement ayant ses côtés parallèles aux axes x et y des éléments saisis.

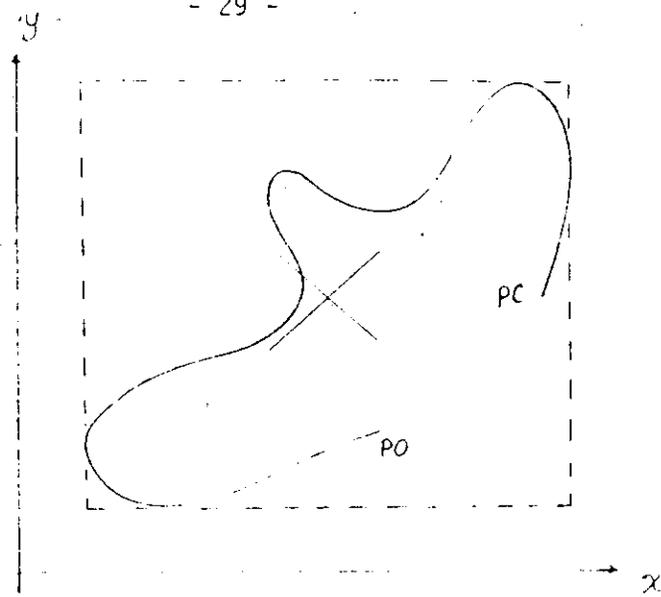


Figure 3.4

L'agrandissement-retrécissement est obtenu par homothétie, le centre d'homothétie étant le centre du contour et le rapport d'homothétie un paramètre qui peut être mis au jour au cours de la création (voir paragraphe 3.2.4).

3.2.2 - Une rotation permet de positionner le contour bidimensionnel dans n'importe quelle position spatiale. Cette opération sera nécessaire pour générer des objets tridimensionnels.

L'axe de rotation est défini passant par le centre du contour et parallèle soit aux axes ox ou oy du trièdre de référence.

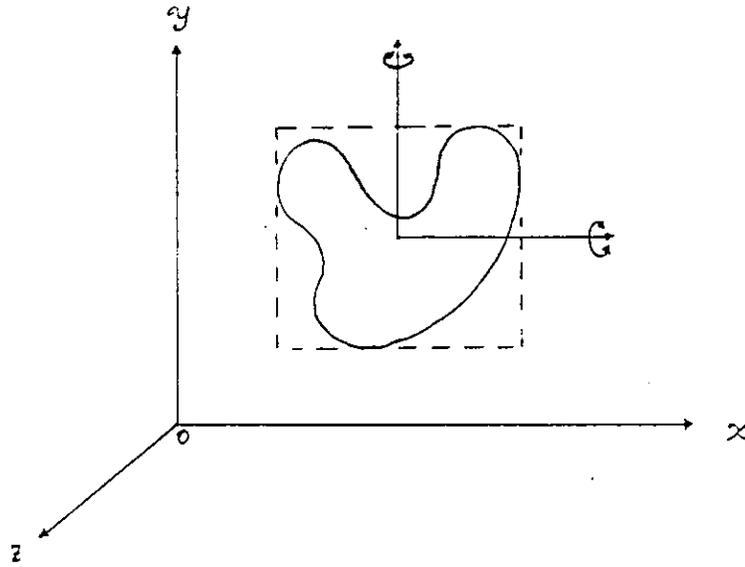


Figure 3.5

Le système assure la planarité du contour en empêchant toute entrée d'éléments tant que la position n'est pas correcte.

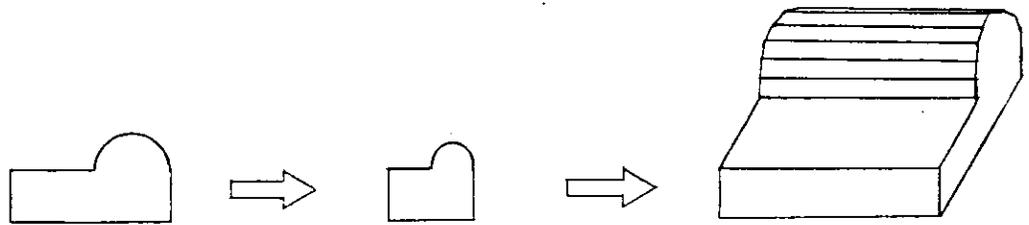


Figure 3.6

3.2.3 - Dans le cas de la saisie de données, à petite échelle par rapport à celle de l'écran, un déplacement de l'image du contour est nécessaire afin d'en visualiser l'ensemble; un retrécissement peut éventuellement être nécessaire pour obtenir une image plein écran.

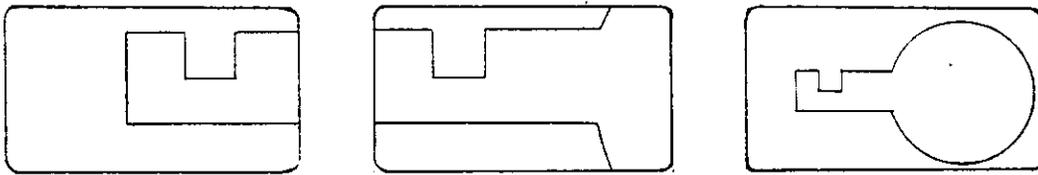


Figure 3.7

3.2.4 - La redéfinition dynamique du point courant, de l'horizontale et de l'échelle offre la possibilité de saisir des données réparties sur des plans différents et à des échelles différentes.

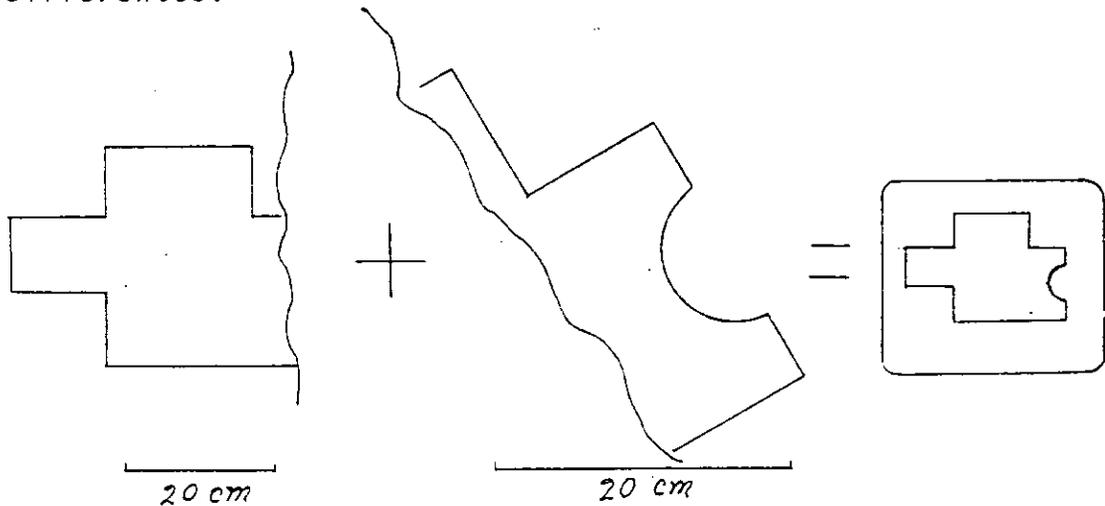
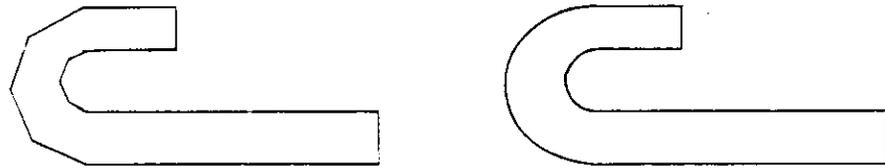


Figure 3.8

3.2.5 - Une modification des paramètres de visualisation, au cours de la saisie des données, donne au système une grande souplesse, il sera ainsi possible de mettre à jour le degré de résolution d'un arc de cercle (nombre moyen de segment de droite par cercle), le temps-limite, la vitesse de rotation, le pas de translation dans des unités écran et le facteur de l'agrandissement-retrécissement.

Deux descriptions du contour créé sont générées; la

première contient une approximation du contour sous forme polygonale qui servira à générer un modèle polyédrique, la seconde contient la description géométrique exacte du contour (c'est-à-dire le contour comme une suite orientée de segments d'arcs de cercle et de splines). C'est cette description qui sera conservée pour applications autres que la visualisation.



Deux représentations du contour de base

Figure 3.9

Trois types de modèles peuvent être générés à partir du contour entré : prismes, modèles de révolution et combinés (déplacements successifs du contour de base).

3.3 - Prismes

Un contour fermé, non croisé, peut être utilisé pour créer un modèle prismé.

Le modèle prismé est construit avec pour base inférieure le contour initial et pour base supérieure un contour homothétique au précédent.

L'homothétie utilisée est déterminée par un vecteur ψ saisi à partir de la tablette et un rapport k donné au clavier alphanumérique.

Le vecteur ψ permet de déterminer un point homologue

au point origine du contour.

Ayant admis que le plan de la tablette est toujours le plan de saisie, une rotation du contour est nécessaire de telle sorte que le plan du vecteur φ corresponde à celui de l'écran.

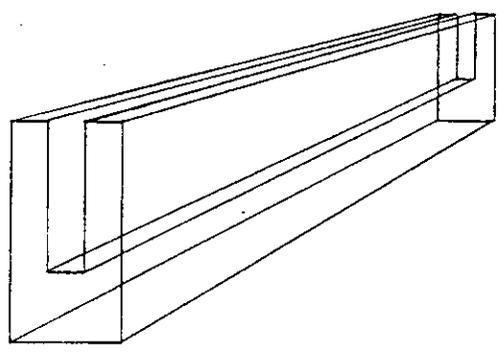
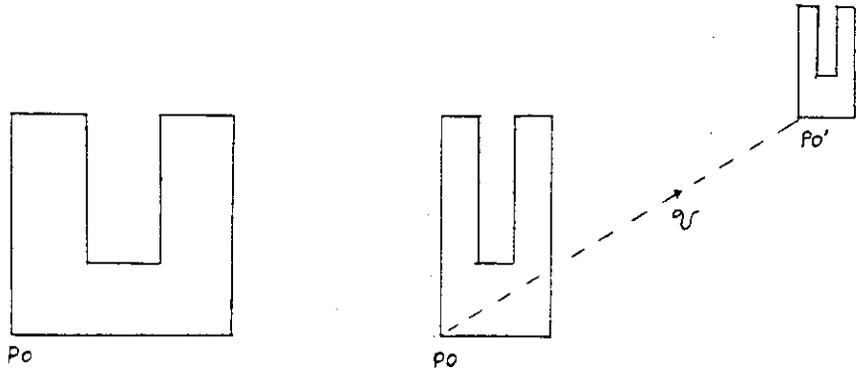
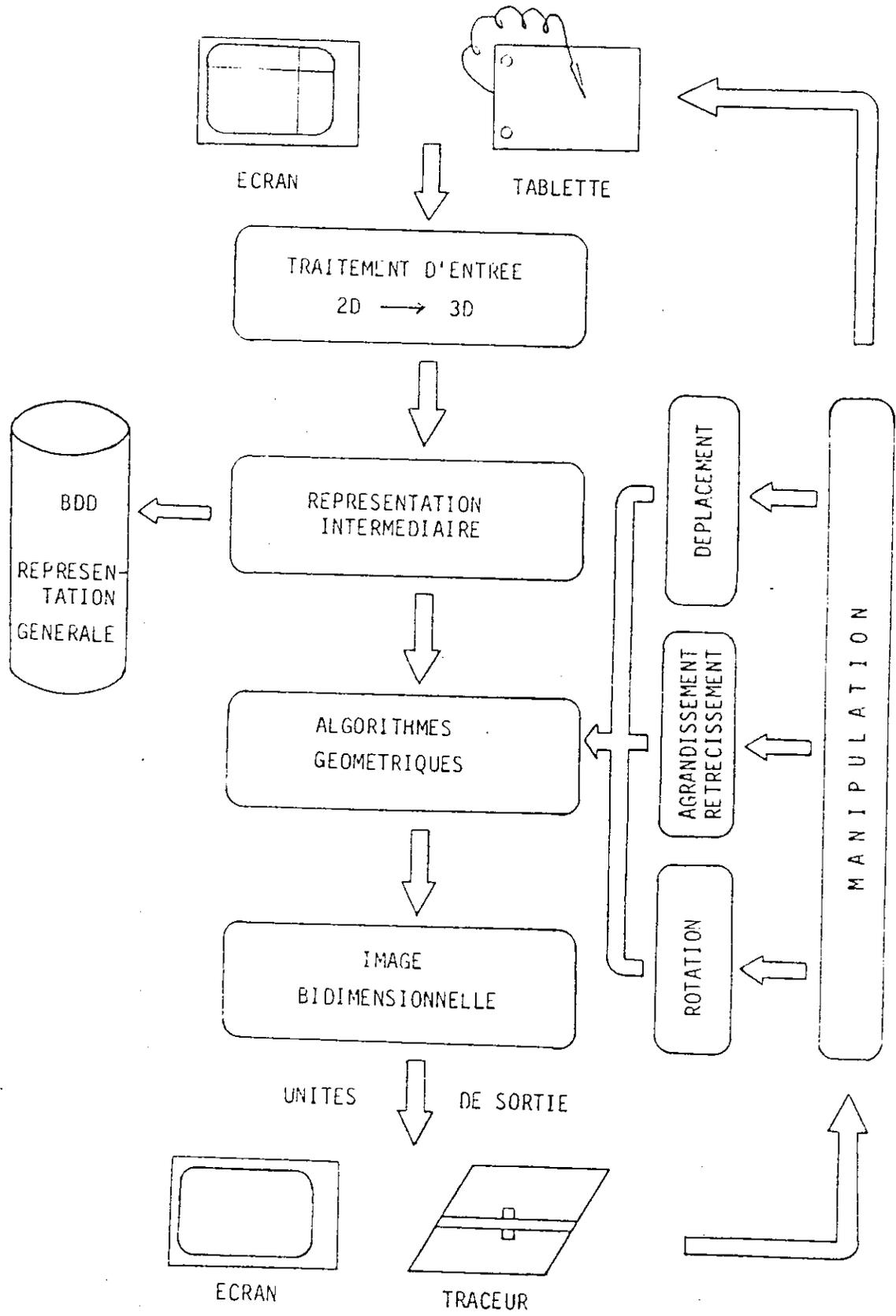


Figure 3.10

UNITES D'ENTREE 2D



SCHEMA DE CREATION MGIT

figure 3.11

On conserve du modèle ainsi créé une approximation polyédrique utilisée pour la visualisation et une description "historique" comportant le contour de base \mathcal{C} , le vecteur v et le facteur de réduction k .

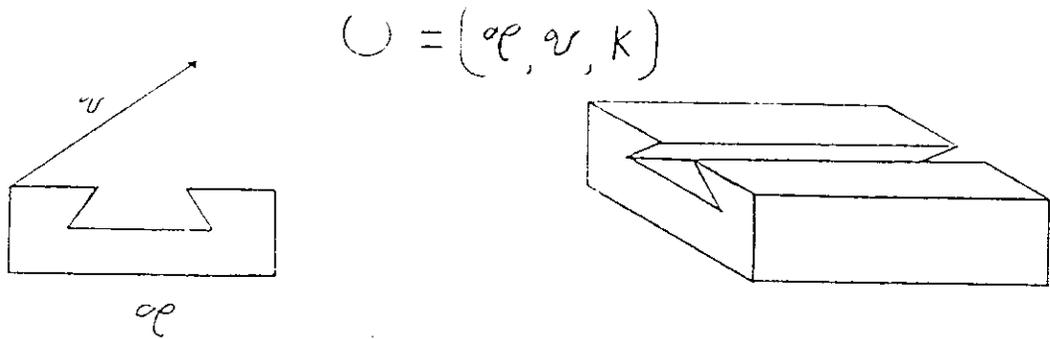
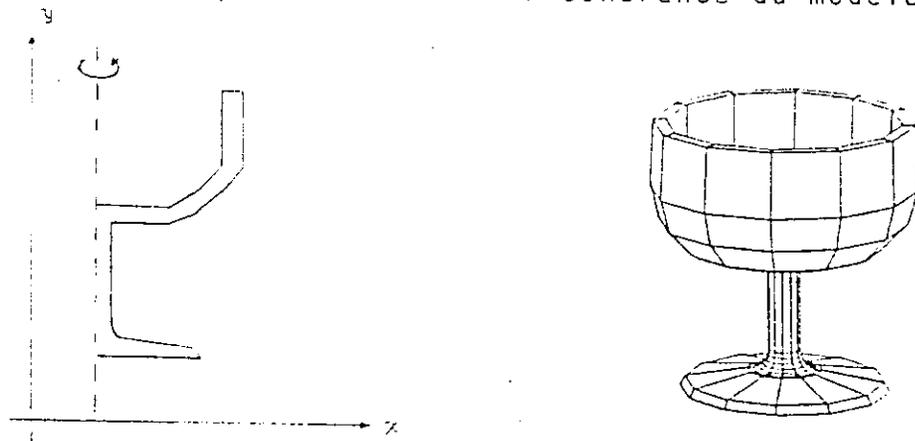


Figure 3.12

3.4 - Modèles de révolution

Un modèle de révolution peut être obtenu en faisant tourner le contour de base, non forcément fermé, autour d'un axe.

Dans le cas d'un contour non fermé, seule est possible la création des modèles par révolution de 360° autour d'un axe passant par l'origine et l'extrémité du contour. Ceci est nécessaire pour conserver la cohérence du modèle.



La méridienne est définie par le contour de base; de même que dans l'opération "Prisme", l'axe est donné à l'aide de la tablette.

L'axe de révolution est défini comme étant parallèle à l'axe oy , ainsi, la méridienne doit être placée dans la position spatiale convenable.

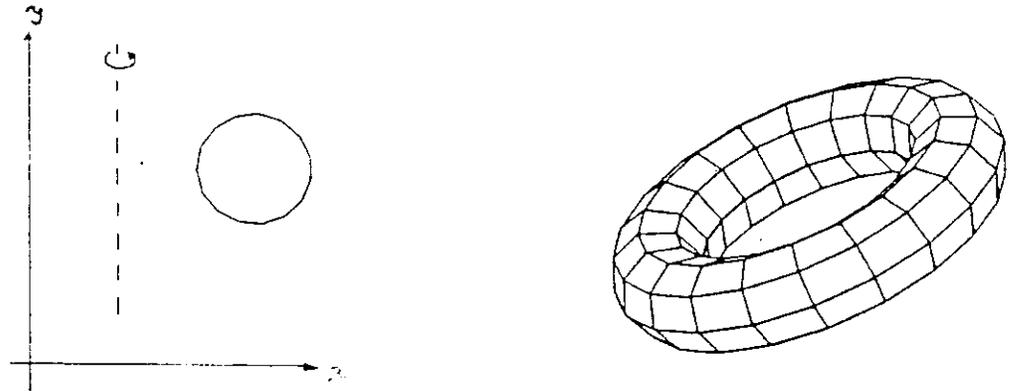


Figure 3.14

On sauvegarde l'historique de création du modèle de révolution à savoir : la représentation du contour \mathcal{C} , l'axe de révolution l et l'angle de révolution φ .

$$\bigcirc = (\mathcal{C}, l, \varphi)$$

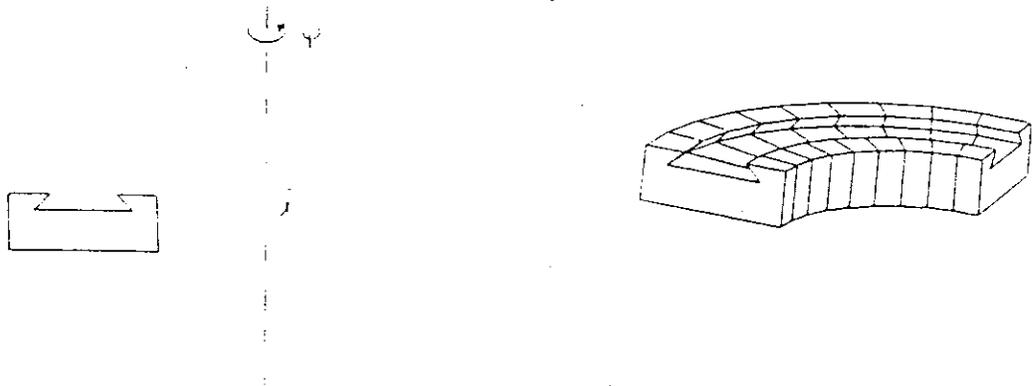


Figure 3.15

ai- 3.5 - Modèles combinés

le Des modèles plus généraux peuvent être obtenus par rotations et déplacements successifs du contour de base.

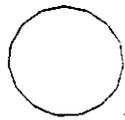
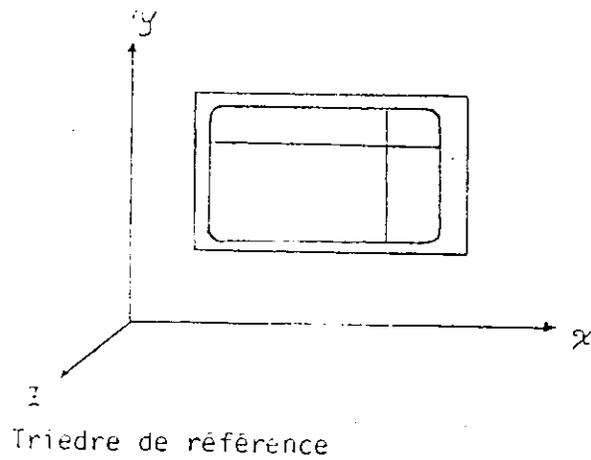
- Le paragraphe qui suit décrit la construction d'un tel modèle.

- 1) Construction d'un contour de base (fig 3.16a).
- 2) Par rotation de 90° autour de oy le plan du contour devient perpendiculaire au plan de l'écran (fig. 3.16.b).
- 3) Par translation parallèle à ox on engendre un corps prismé. La notion de matière est introduite à partir de cette étape (fig. 3.16 c).
- 4) Le contour est déplacé par rotation autour de oz de façon à engendrer un "coude". Le système empêche toute intersection de l'objet par lui-même (fig. 3.16 d).
- 5) On translate parallèlement à oy et on réduit le contour par homothétie. Le centre d'homothétie est le centre du contour courant (qui sera réduit). Le rapport est rentré à l'aide du clavier alphanumérique (fig. 3.16.e).
- 6) Afin de définir un plan de travail autre que le plan xy, on tourne le modèle de 90° autour d'un axe parallèle à oy. L'axe de rotation d'un modèle passe toujours par le centre du prisme rectangulaire

minimum qui le contient (ayant ses côtés parallèles aux axes) et est parallèle soit à ox , soit à oy du trièdre de référence (fig. 3.16. f).

7) Le contour est déplacé par rotation de 90° autour de Oz puis parallèle à ox (fig. 3.16.g).

8) Enfin, deux translations avec agrandissement du contour courant sont effectuées (fig. 3.16.h).



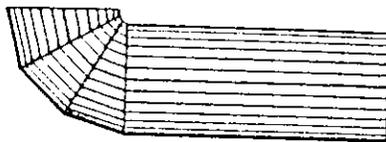
A



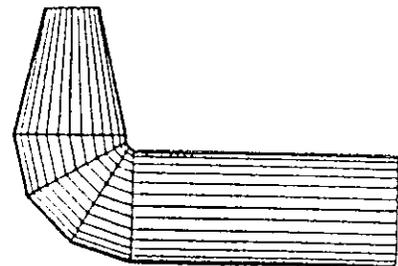
B



C



D



E

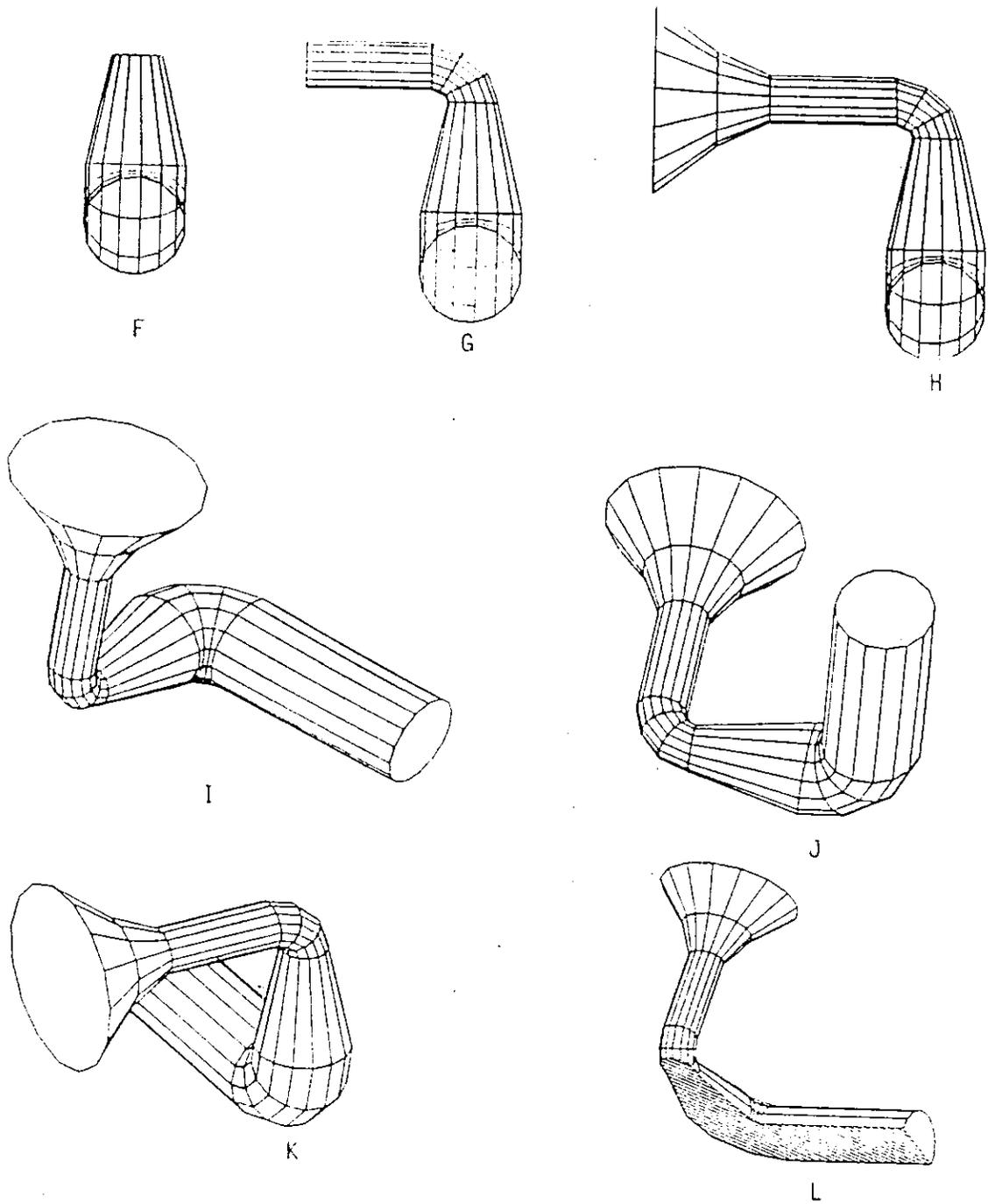


Figure 3.16

Le modèle obtenu a évidemment comme historique de création une combinaison des deux cas précédents.

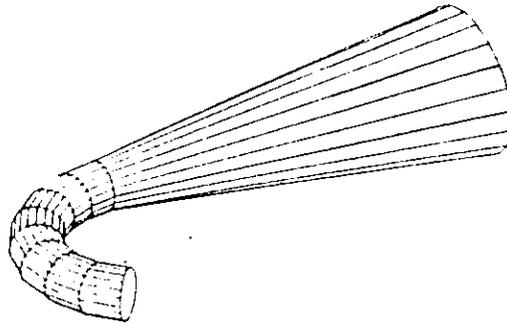
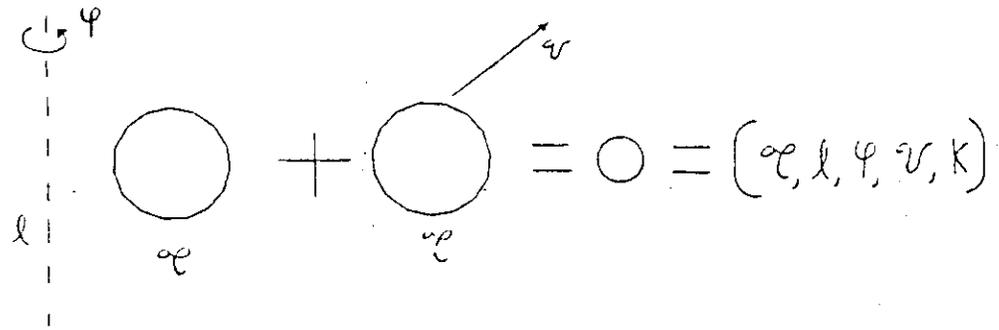


Figure 3.17

Il est ainsi possible d'obtenir un modèle combiné. à partir soit d'un modèle prismé, soit d'un modèle de révolution créé auparavant.

3.6 - Vérification du modèle

Une visualisation avec élimination des parties cachées du modèle conçu peut être obtenue. Cette visualisation permet une vérification rapide de la forme attendue ainsi que de la cohérence du modèle. Une réponse suffisamment rapide de l'algorithme d'élimination des parties cachées permet, tout en gardant l'interactivité, d'observer une rotation continue du modèle ainsi qu'un déplacement ou une mise à l'échelle (agrandissement-rétrécissement).

Si la visualisation avec élimination des parties cachées du modèle n'est pas suffisante pour vérification de la forme cherchée, il existe la possibilité de sectionner le modèle puis de lui appliquer une rotation, mise à l'échelle ou déplacement afin de privilégier les parties qui semblent importantes.

Le plan de coupe est perpendiculaire au plan de l'écran et sa trace sur l'écran est définie interactivement. Le modèle peut être positionné convenablement à l'aide d'une rotation autour de l'axe ox ou de l'axe oy . Cette opération ayant un caractère purement visuel ne requiert pas une détermination rigoureuse du plan de coupe par rapport au modèle.

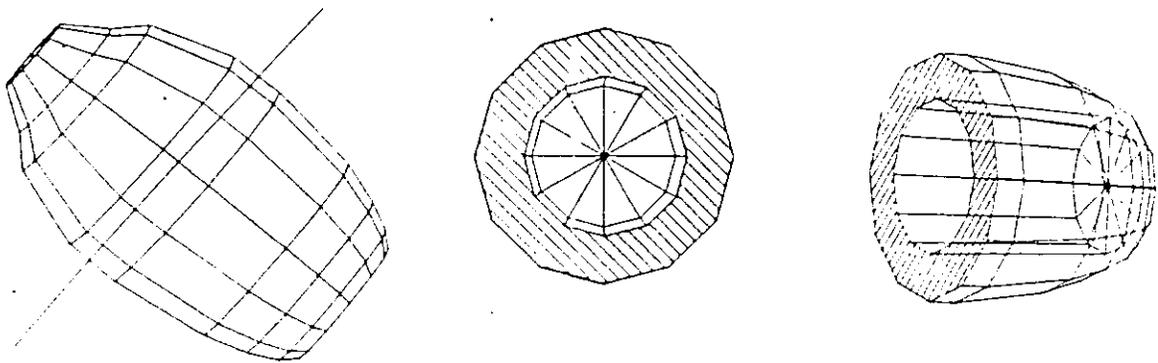
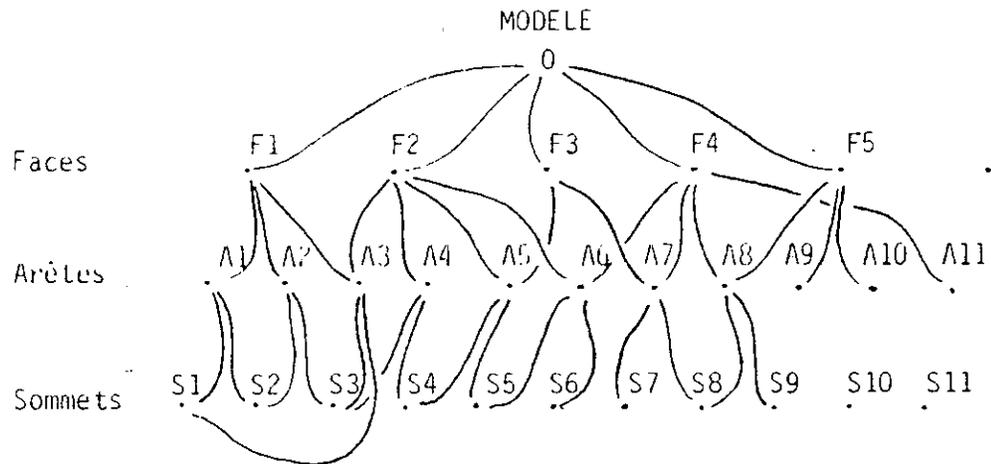


Figure 3.18

La visualisation du modèle sectionné se fait avec une élimination partielle des parties cachées ce qui permet, tout en gardant l'interactivité, d'obtenir une représentation suffisamment claire de la section.

Une approximation du modèle sous forme polyédrique est conservée. Cette représentation contient l'information nécessaire aux visualisations du modèle géométrique.



Représentation topologique

Figure 3.19

Par ailleurs, l'historique de création permet de conserver toute l'information attachée au modèle géométrique. Il pourra être utilisé pour l'usinage de l'objet.

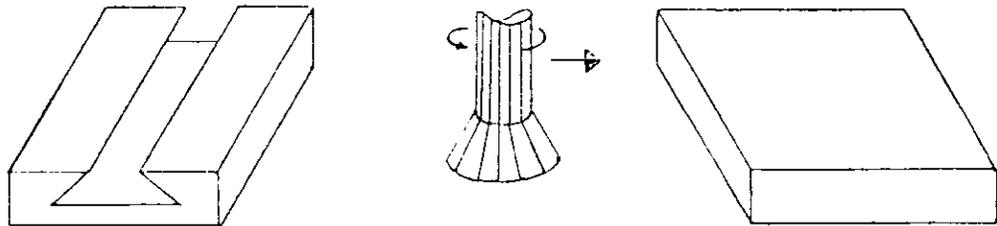
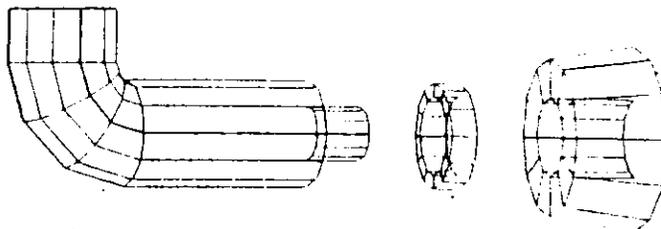
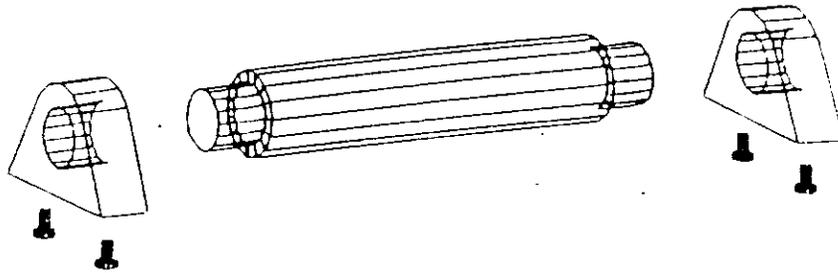
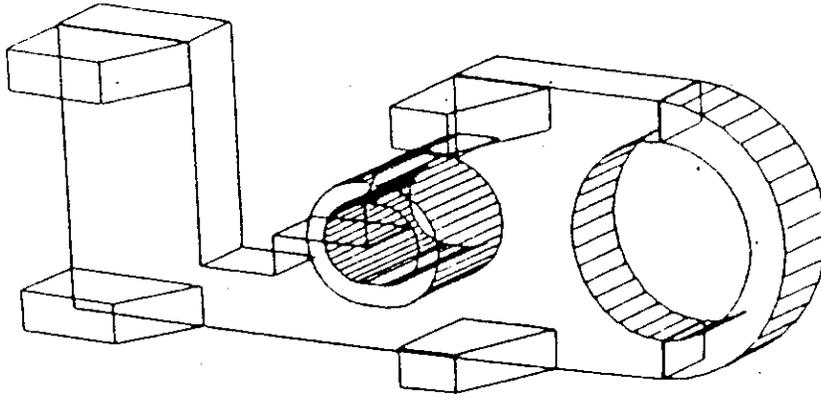


Figure 3.20



modèles construits par MGIT
figure 3.21

CHAPITRE IV

ALGORITHME DE VISUALISATION DE POLYEDRES
AVEC ELIMINATION DES PARTIES CACHEES

" Penser en images demeure une forme très complète de l'acquisition de la conscience, cela approche de plus près les processus inconscients que ne le fait la pensée verbale. "

Sigmund FREUD

4. La modélisation géométrique interactive requiert des algorithmes capables de produire une image de qualité qui permettra à l'opérateur de prendre des décisions. Parmi ces algorithmes, celui d'élimination des parties cachées est d'une grande importance pour la visualisation de modèles géométriques tridimensionnels.

Cependant, le temps nécessaire pour décider quelles parties de l'objet seront visibles peut être important, ce qui représente un sérieux inconvénient pour un système interactif. Les efforts réalisés pour résoudre ce problème ont débouché sur une collection considérable d'algorithmes. Mais ceux-ci n'ayant pas un temps de réponse suffisamment court, nous faisaient perdre le haut niveau d'interactivité que nous nous étions imposé comme caractéristique essentielle de notre modéleur géométrique.

Nous avons donc été amené à concevoir un algorithme d'élimination des parties cachées dont les performances soient compatibles avec notre application.

L'algorithme obtenu calcule la vue du modèle comme un ensemble de traits correspondant aux arêtes de l'objet. Les calculs sont réalisés avec une projection orthographique dans l'espace objet.

La résolution est alors indépendante de l'échelle d'utilisation.

4.1 - Rappel des algorithmes

Les algorithmes existants peuvent être classés en trois groupes : ceux qui calculent dans l'espace objet, ceux qui le font dans l'espace image et enfin ceux dits "algorithmes de liste prioritaire" qui le font dans les deux espaces.

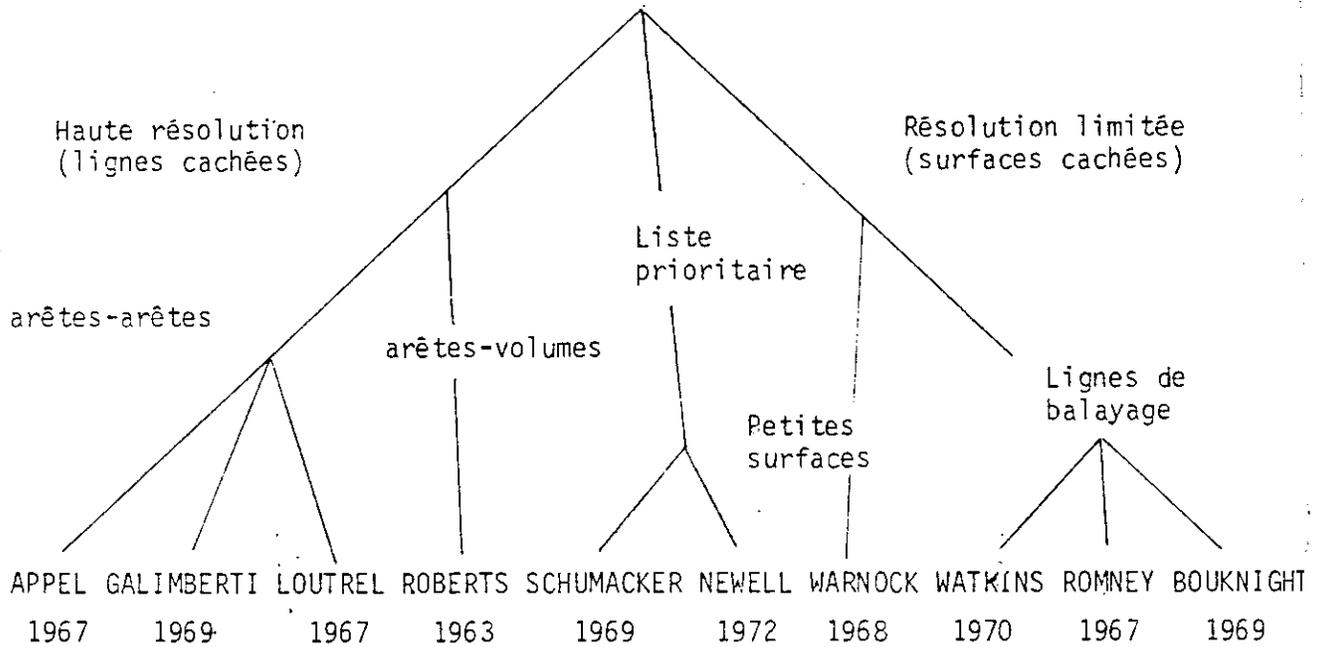


Figure 4.1

Pour un algorithme travaillant dans l'espace-objet, les calculs sont effectués avec une précision au moins équivalente à celle avec laquelle les objets sont définis.

Par contre, pour un algorithme travaillant dans l'espace-image, la précision requise est celle avec laquelle l'image est représentée sur l'écran.

En schématisant à l'extrême, un algorithme du premier groupe répond à la question "telle partie de l'objet est-elle visible"? tandis que ceux du deuxième groupe répondent à la

question "tel point sur l'écran appartient-il à l'image de la partie visible de l'objet"?

Un algorithme du premier groupe calcule donc une image "numériquement" correcte (elle est correcte même agrandie plusieurs fois) tandis que ceux du second calculent une image "visuellement" correcte.

De cette différence, il résulte que le coût d'un algorithme d'espace-objet croît en fonction de la complexité de l'objet tandis que le coût de l'algorithme d'espace-image est constant car le nombre de points adressables sur l'écran est le même, indépendamment de la complexité de la scène.

4.2 - Algorithme

Définition

Un algorithme d'élimination de lignes cachées est un 4 - tuple

$$AELC = (M_0, V, I, \varphi)$$

où

M_0 est le modèle tel que défini dans le paragraphe 2.3.2 et représenté dans la figure 4.2.

V est l'ensemble des segments coplanaires vus par l'observateur.

I est un ensemble fini de représentation I_i intermédiaires avec

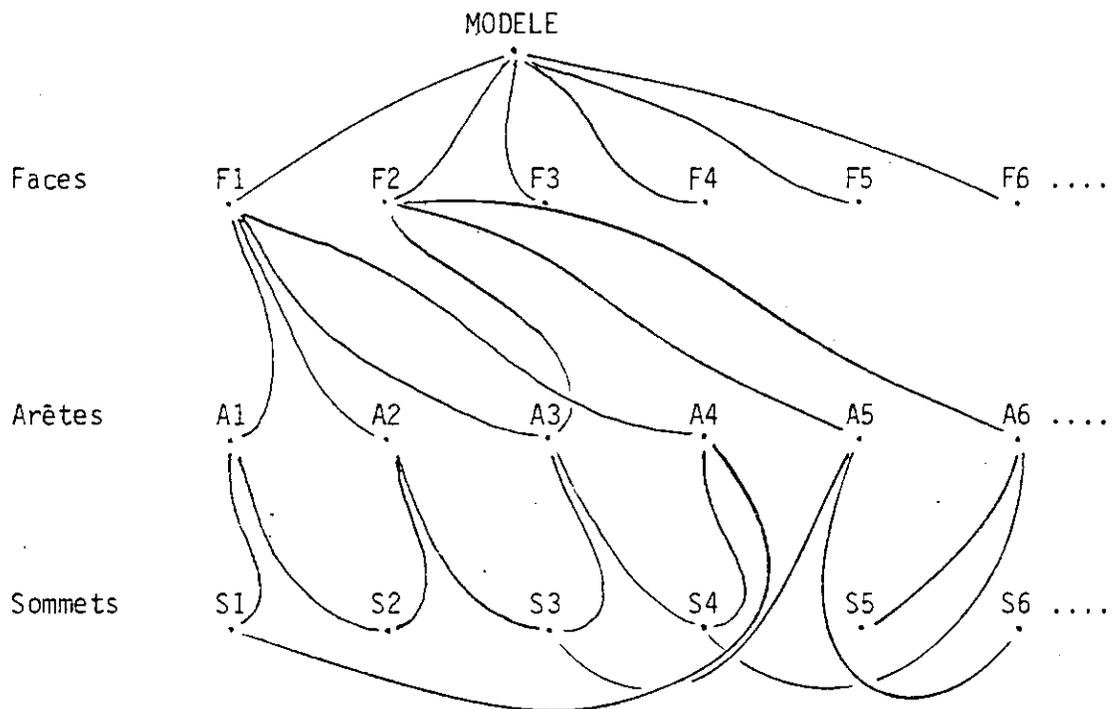
$$\begin{aligned} & I_0 = M_0 \\ \text{et si} & \quad n = \text{Card}(i) \\ & I_n = V \end{aligned}$$

φ est un ensemble ordonné de fonctions φ_i de transition.

$$\varphi = \{ \text{NO, AC, ST, EN, FD, FV} \}$$

4.3 - Description du modèle M0

Le modèle M0 est un ensemble structuré de faces, d'arêtes et de sommets décrit au paragraphe 2.3.2. Nous donnons ci-dessous la représentation topologique I0 du modèle.



Représentation topologique du modèle I0.

Figure 4.2

4.4 - Descriptions des fonctions de transition

4.4.1 - NO est une fonction de normalisation, elle assure pour l'ensemble S des sommets, une coordonnée en x comprise entre 1. et xmax défini par

$$x_{\max} = \text{Card}(S) + 1$$

Cette normalisation permettra de diminuer largement

le coût des tris nécessaires à l'algorithme (voir paragraphe 5.4).

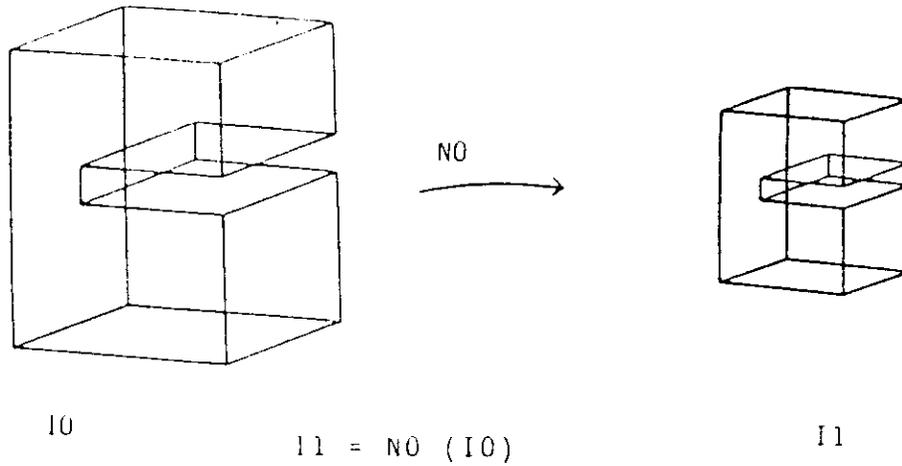


Figure 4.3

4.4.2 - AC est une fonction qui élimine des faces, arêtes et sommets cachés (voir paragraphe 2.5.2).

L'application de cette fonction élimine des morceaux du modèle; on obtient ainsi un ensemble structuré de faces, arêtes, sommets.

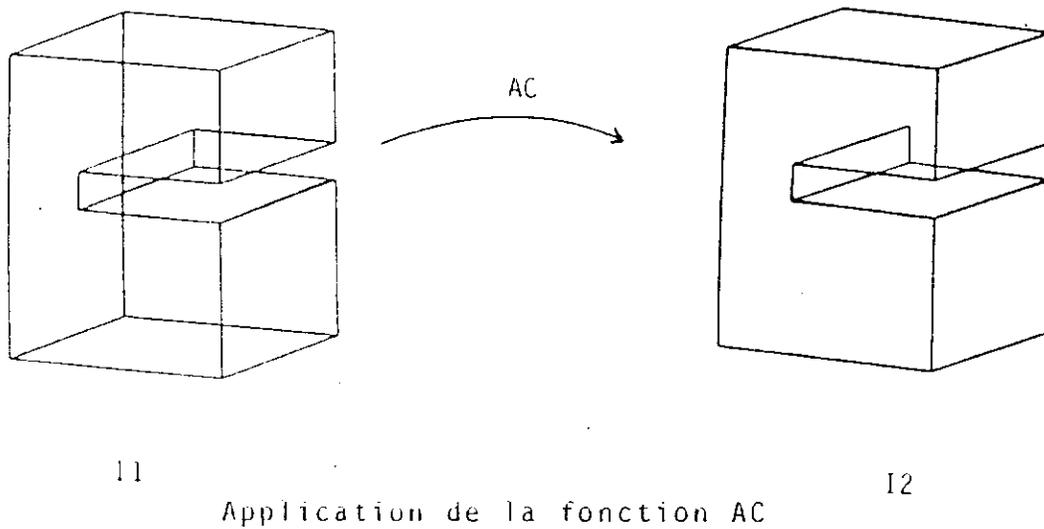
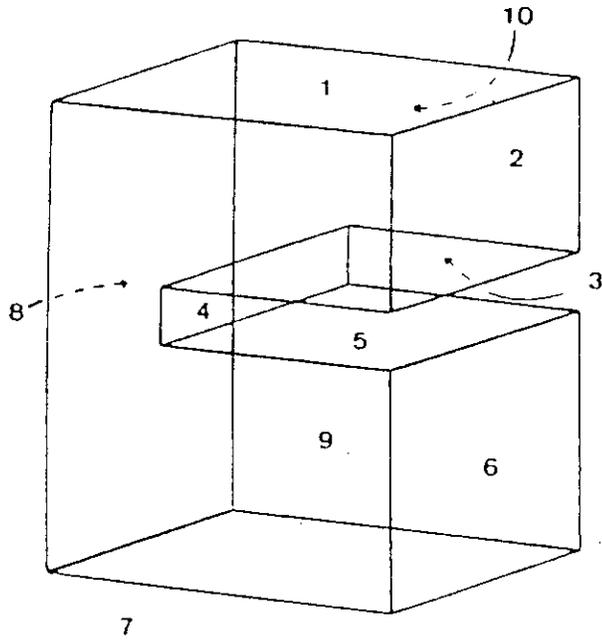
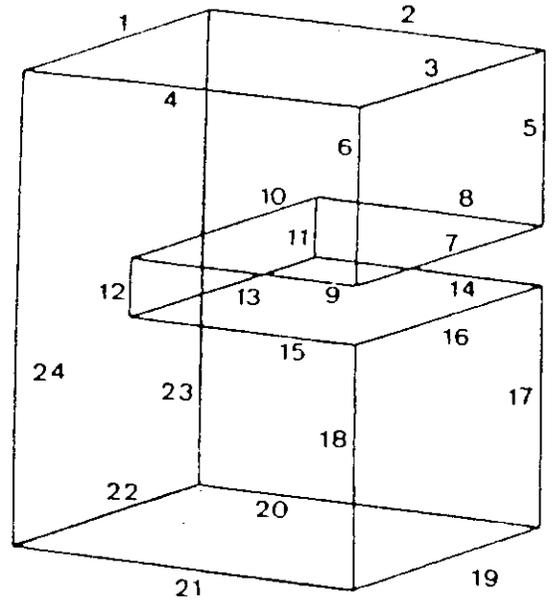


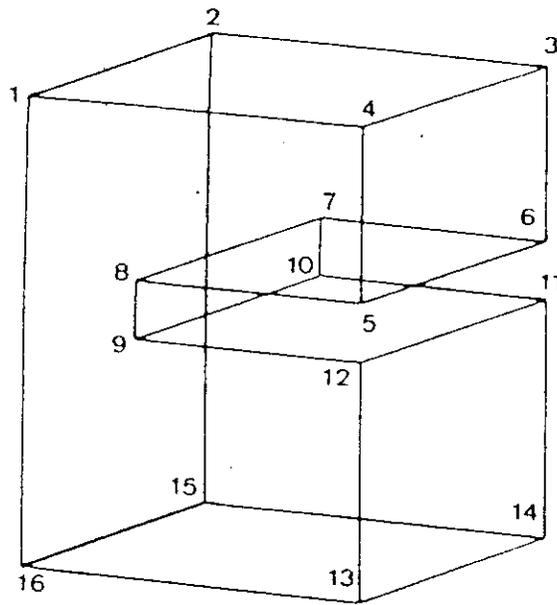
Figure 4.5



10 FACES



24 ARETES



16 SOMMETS
MODELE EXEMPLE

Figure 4.4

Cet ensemble n'est plus l'image du modèle; la représentation intermédiaire I2 est un sousgraphe de I1

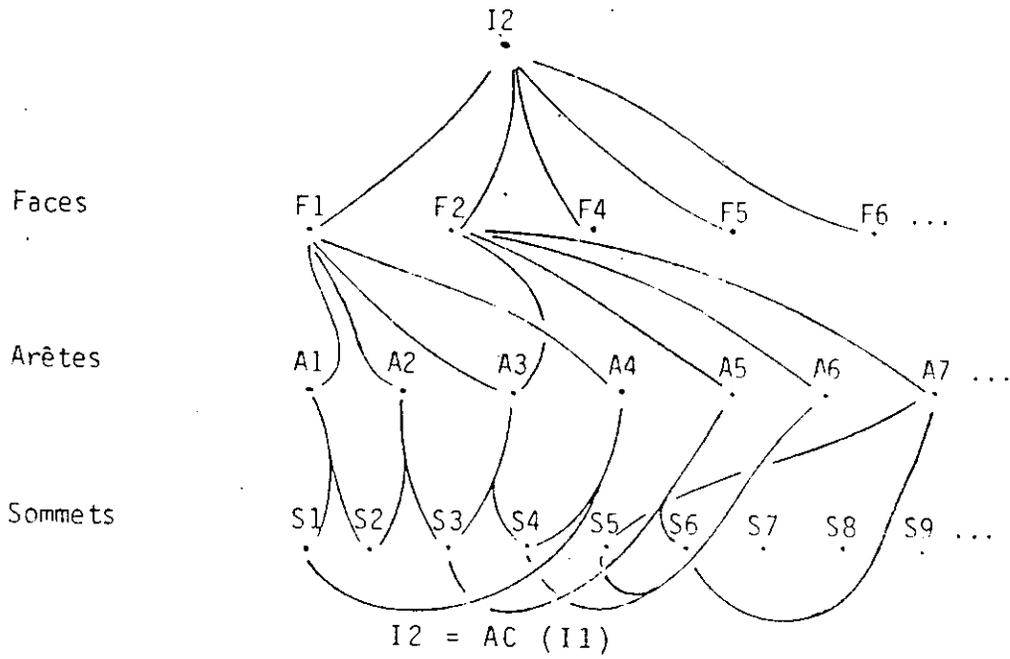


Figure 4.6

Algorithme de détermination des faces et
arêtes cachées correspondant à AC

LIGCCC

Données VV, VN

Logique FAC, ARE

Tableaux FAC (N), ARE (M), VN (N)

FAC (i) = VRAI si la face i est visible sinon elle est cachée
ARE (i) = VRAI si l'arête i est visible sinon elle est cachée
VV est le vecteur normal au plan du dessin et dirigé vers le
modèle.

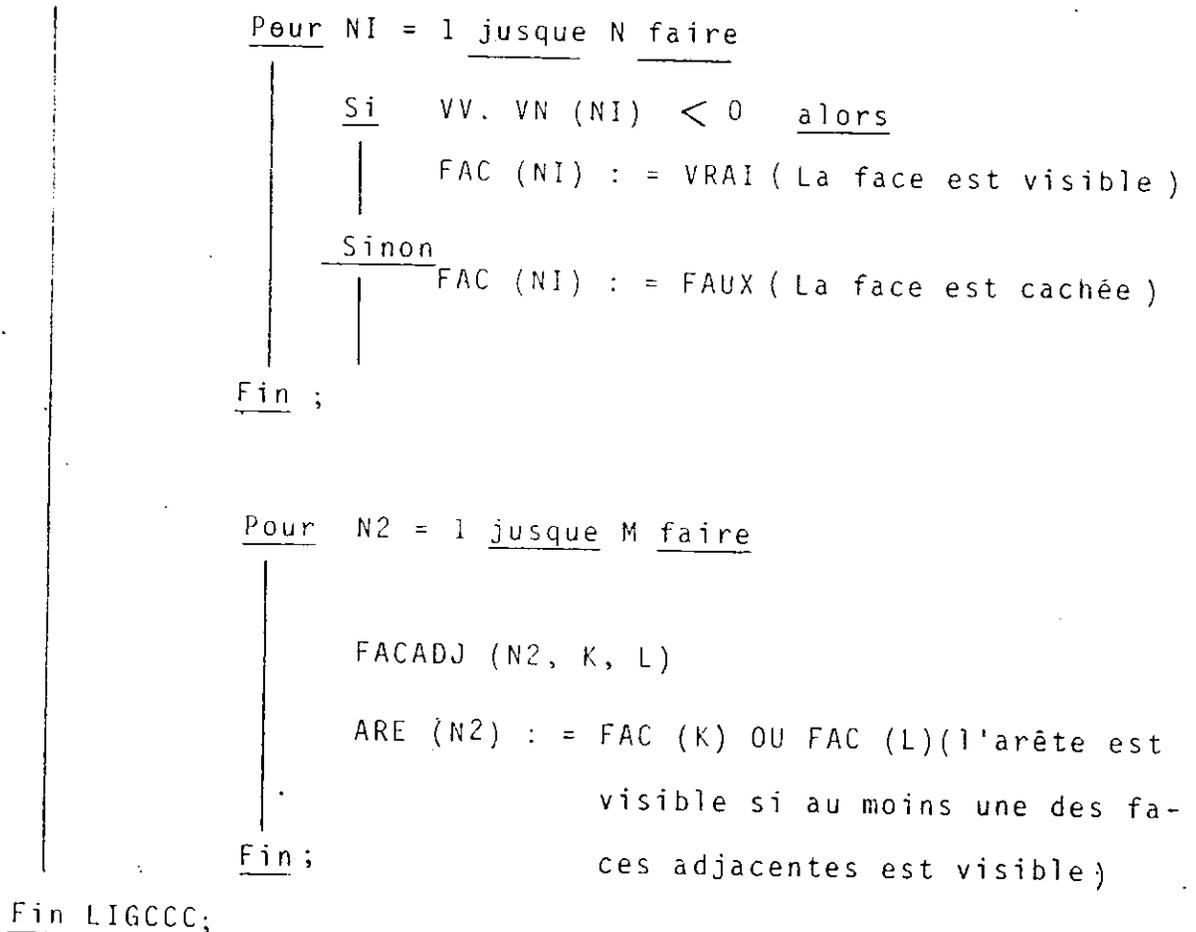
VN (i) contient le vecteur normal à la face FAC (i)

N est le nombre de faces et M le nombre d'arêtes

FACADJ (i, j, k) fournit les faces J, k adjacentes à l'arête i

Visible signifie potentiellement visible.

Début



4.4.3 - ST est une fonction qui permet de découper certaines arêtes de I2.

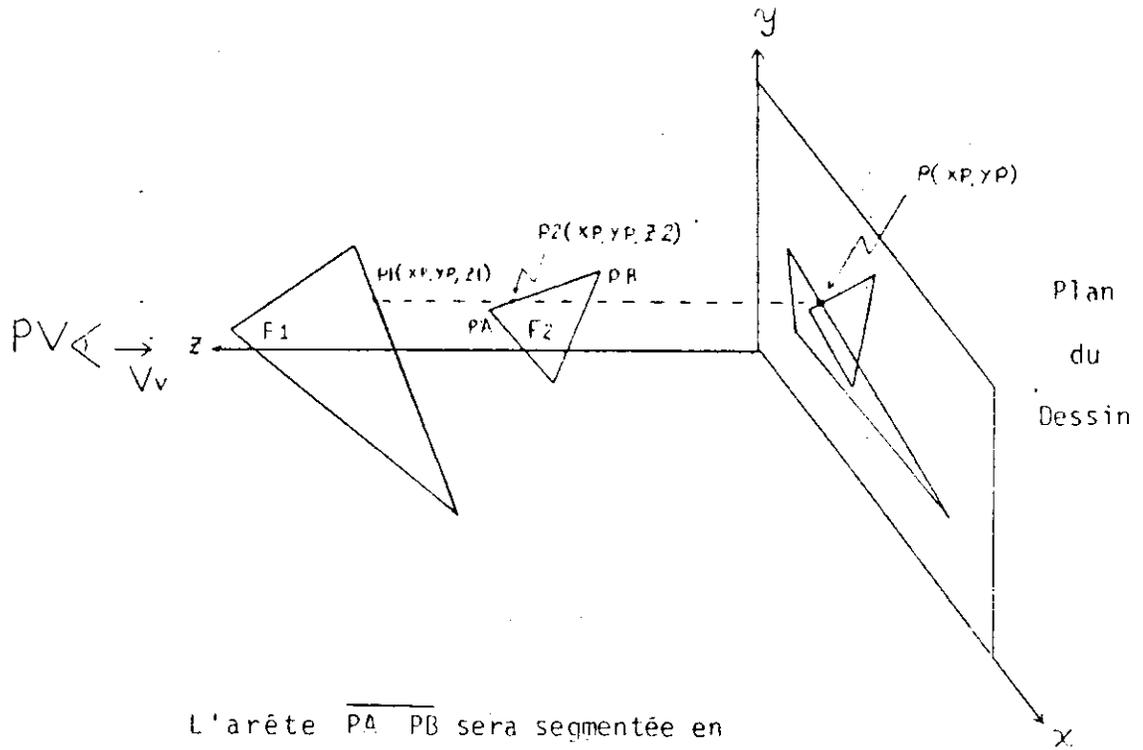
Pour ce faire, les arêtes sont comparées deux à deux de la façon suivante :

- I - Les arêtes sont projetées sur le plan du dessin.
- II - Si les projections des arêtes ont une intersection P (qui n'est pas la projection d'un sommet)
 - 1) On calcule les coordonnées du point P
 - 2) Soit P1 et P2 les points sur les deux arêtes ayant P comme projection. On choisit parmi ces deux points le plus éloigné de l'observateur et on l'affecte à l'arête correspondante.

Ainsi, à chaque intersection d'arête dans le plan du dessin correspond une arête et une seule. Une arête peut se voir ainsi affectée aucun ou plusieurs points.

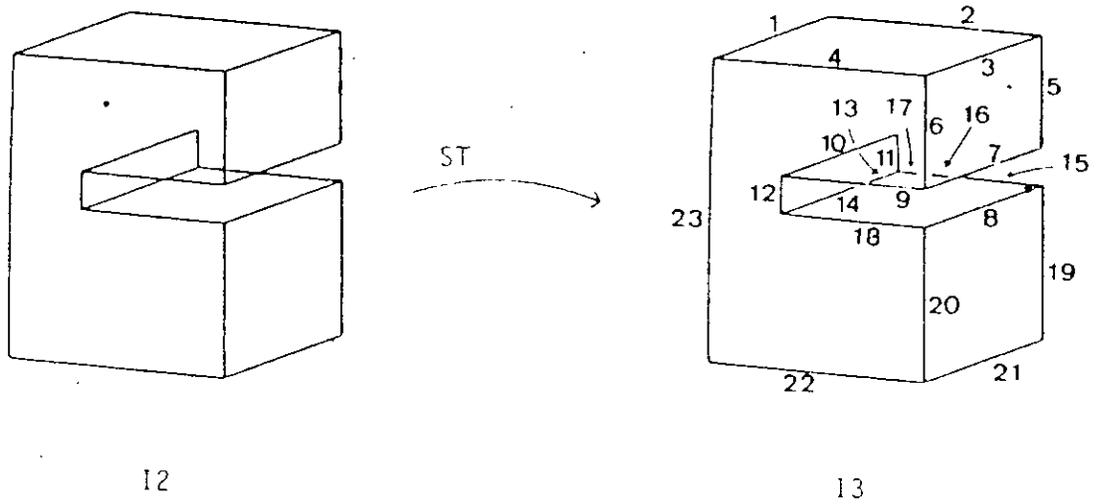
Ensuite, pour chaque arête les points qui lui sont affectés sont ordonnés suivant leur abscisse curviligne de telle sorte qu'ils déterminent une partition de l'arête en segments jointifs.

On associe enfin à chaque point d'intersection l'arête qui lui a donné naissance (c'est-à-dire celle à laquelle il n'a pas été affecté).



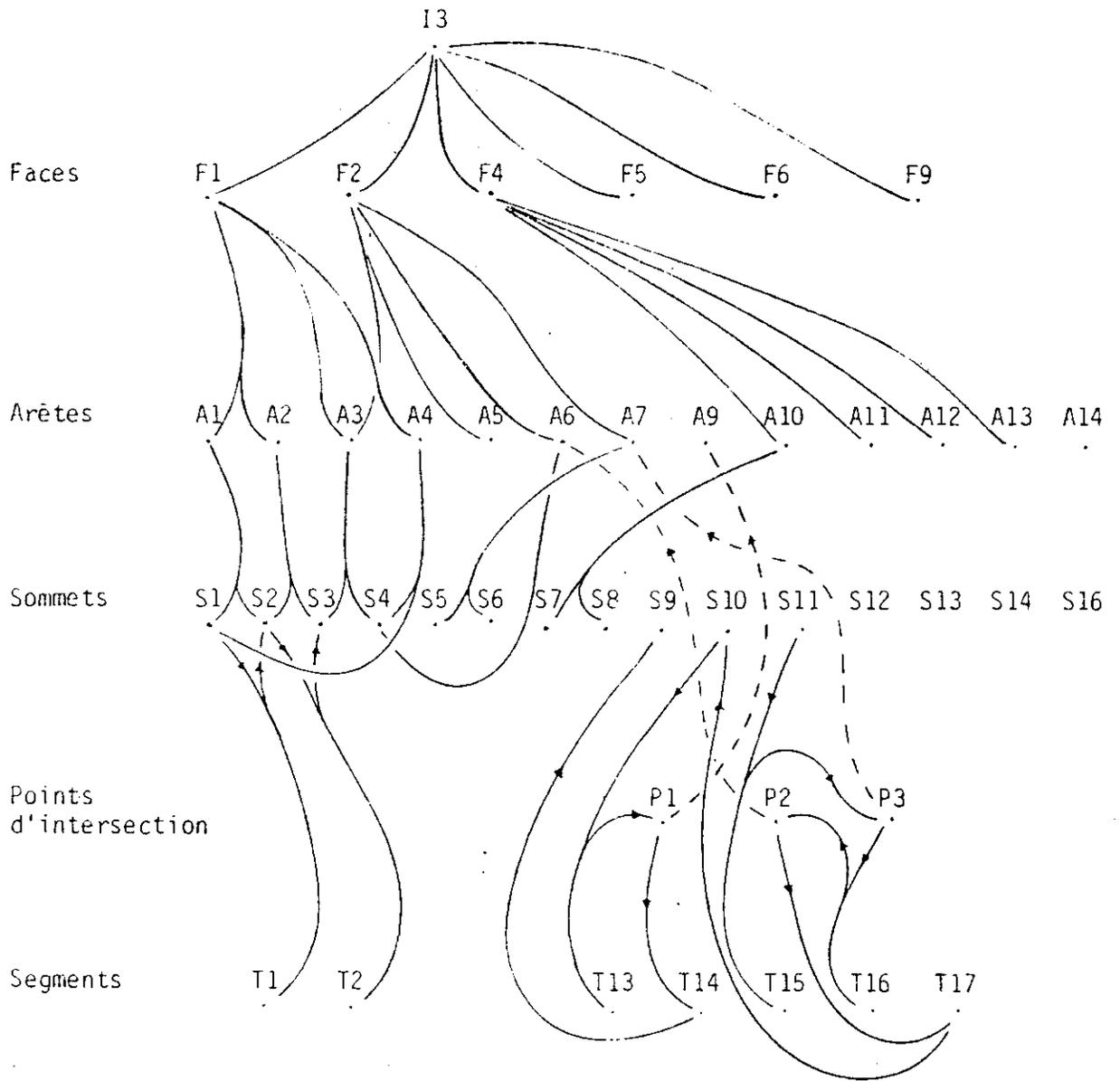
L'arête $\overline{P_A P_B}$ sera segmentée en $\overline{P_A P_2}$ et $\overline{P_2 P_B}$

Figure 4.7



Application de la fonction ST

Figure 4.8



$$I3 = ST(I2)$$

Structure de I3

Figure 4.9

La fonction ST crée au moins un segment par arête, c'est-à-dire que si une arête n'a pas été coupée, elle est confondue avec le segment qui lui est associé (cas de l'arête A1 dans la figure 4.9); on trouve ainsi des segments délimités par

un sommet et un point d'intersection (premier segment d'une arête), des segments délimités par deux points d'intersection (segments intermédiaires), des segments délimités par un point d'intersection et un sommet (dernier segment d'une arête) et enfin des segments délimités par deux sommets (segment unique).

4.4.4 - EN est une fonction qui détermine l'ensemble naturel de chaque sommet de I3.

Définition

On appelle ensemble naturel ES du point S l'ensemble des faces f de I3 telles que :

- la projection de S soit intérieure à la projection de f sur le plan du dessin.
- Le point \hat{S} soit du côté opposé au point de vue par rapport à f .

L'ensemble naturel ES contient les faces qui cachent le point S .

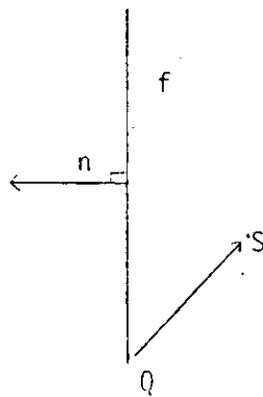
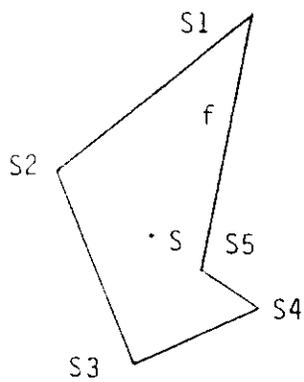
Le calcul des ensembles naturels de tous les sommets requiert la comparaison sur le plan de projection de tous les sommets avec toutes les faces (voir paragraphe 5.6). Ensuite, lorsque le polygone délimitant une face contient un sommet, il faut déterminer si cette face se trouve entre l'observateur et le sommet. Pour ce faire, on procède de la façon suivante :

soit S le sommet, f la face en question et \vec{n} la normale à f .

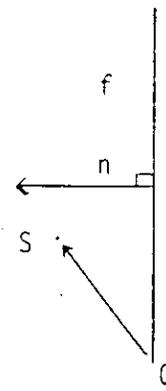
On choisit un point Q du contour de f et on calcule le produit scalaire.

$$\vec{n} \cdot \vec{QS}$$

si $\vec{n} \cdot \vec{QS} < 0$ la face f se trouve entre l'observateur et le point S . Pour le vérifier, il suffit de remarquer que ce calcul est effectué lorsque les faces cachées ont été éliminées et donc le vecteur normal \vec{n} est dirigé vers le demi-espace contenant le point de vue.

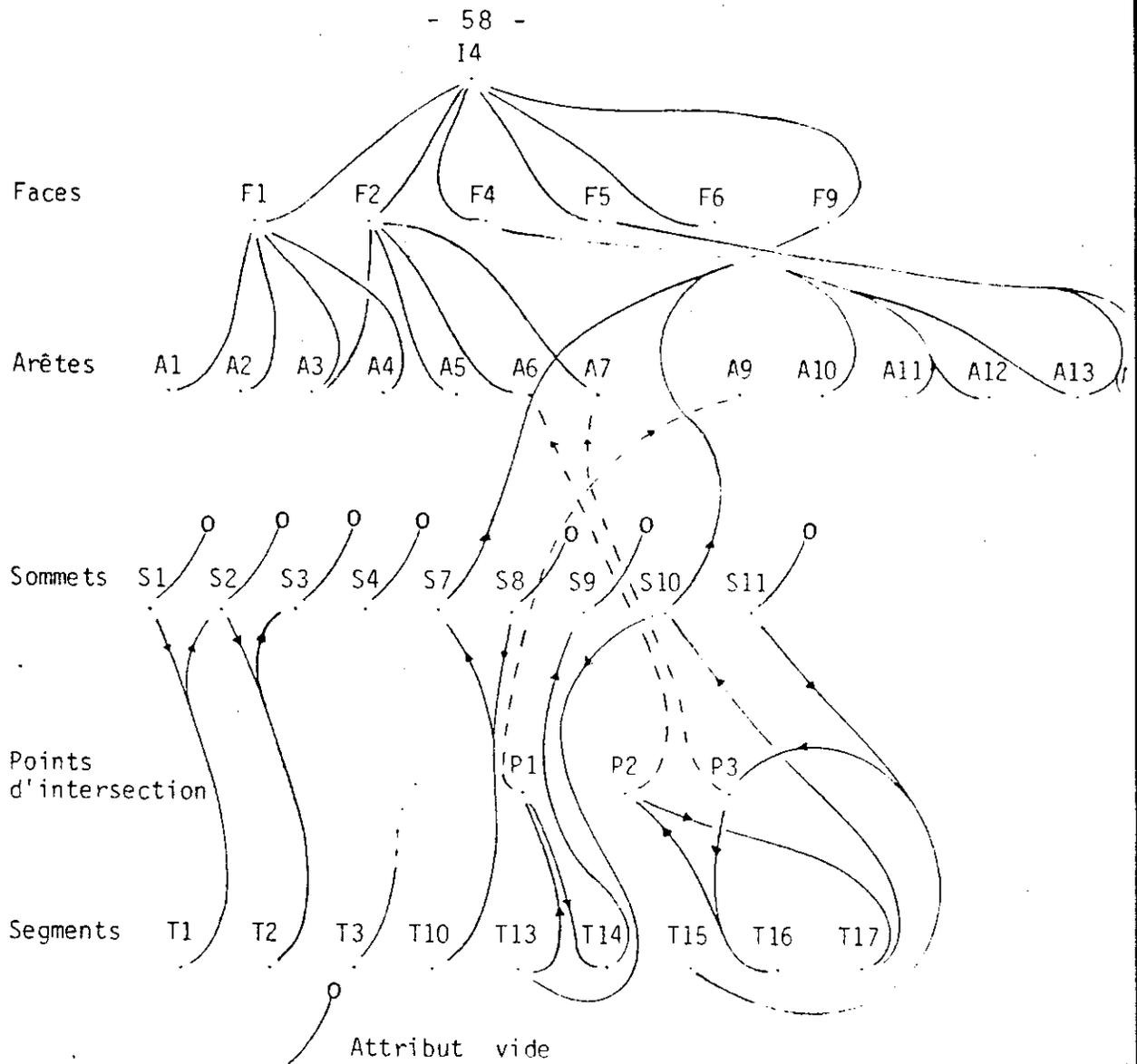


f cache S



f ne cache pas S

Figure 4.10



I4 = EN (I3)

Figure 4.11

La représentation intermédiaire I4 se compose :

- de segments coplanaires ordonnés T_i définis par leurs deux extrémités (qui sont soit l'image d'un sommet sur le plan du dessin, soit un point d'intersection).
- De sommets du modèle initial ayant comme attribut leur ensemble naturel.

- De faces et d'arêtes du modèle initial avec leur relation d'incidence.
- Des relations segment-arête et points d'intersection-arête créés par la fonction ST.

4.4.5 - FD est une fonction qui calcule l'ensemble naturel E_j associé au premier segment T_j d'une arête.

Il faut tout d'abord noter qu'étant donné un segment T de I_4 , d'origine O_1 et d'extrémité O_2

$$ES_1 = ES_2 \quad \forall \quad S_1, S_2 \in]O_1 O_2[$$

c'est-à-dire : l'ensemble naturel est invariant le long de T

L'ensemble naturel est constant tout au long d'un segment car le découpage effectué par ST a permis d'obtenir des segments coplanaires qui ne s'intersectent que par leurs extrémités, donc ils ne traversent jamais l'image du polygone délimitant une face, seule condition qui peut faire changer l'ensemble naturel.

Etant donné une arête $A \in I_4$ ayant pour origine le sommet O_1 et pour extrémité le sommet O_2 , on se propose de déterminer quel sera l'ensemble naturel ET_1 associé au premier segment T_1 , de A .

L'ensemble naturel E_{O_1} est évidemment inclus dans ET_1 : toutes les faces qui cachent le sommet O_1 cachent T_1 . Par ailleurs soit F une face incidente à O_1 et ne contenant pas A , F n'appartient pas à E_{O_1} . La question qui se pose est de savoir si F cache T_1 et si de ce fait F doit être

incluse dans ET1.

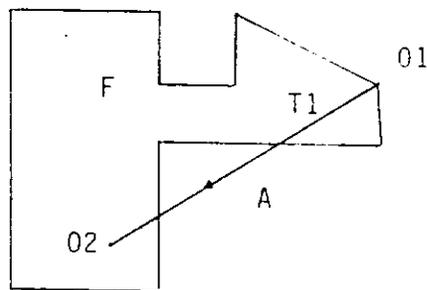
Pour y répondre on utilise le critère suivant :

soit m le nombre d'intersections sur le plan du dessin entre la projection de l'arête A et le contour de F (éventuellement $m=0$ et dans ce cas m doit être considéré comme étant pair) :

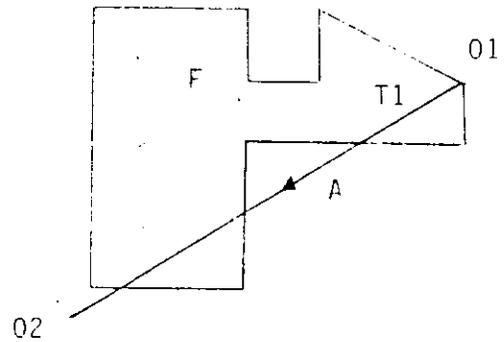
si $F \in E02$ et m pair alors $F \in ET1$

si $F \notin E02$ et m impair alors $F \in ET1$

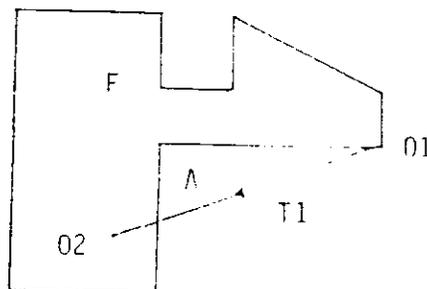
sinon $F \notin ET1$.



$F \in E02$ et $m = 2 \Rightarrow F \in ET1$



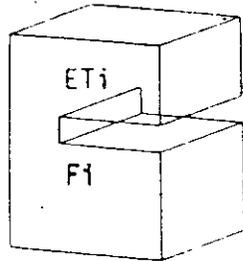
$F \notin E02$ et $m = 3 \Rightarrow F \in ET1$



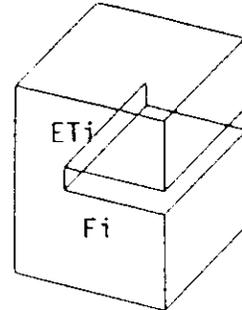
$F \notin ET1$

Figure 4.12

Donc l'ensemble naturel du premier segment d'une arête est le même que celui du sommet origine de l'arête plus éventuellement des faces obtenues par l'analyse décrite ci-dessus.



$F_i \in E_{T_i}$



$F_i \in E_{T_i}$

Figure 4.13

4.4.6 FV est un prédicat qui assigne la qualité vu ou non vu à chaque segment de I_4 .

On définit comme degré de visibilité K_i du segment T_i l'opposé du cardinal de l'ensemble naturel E_{T_i} . Un segment sera vu si et seulement si son ensemble naturel est vide, c'est-à-dire si son degré de visibilité K est égal à zéro.

Le degré de visibilité du premier segment d'une arête se déduit de son ensemble naturel calculé par la fonction FD .

Pour tous les autres segments de l'arête, les ensembles naturels donc les degrés de visibilité sont obtenus par récurrence.

Supposons donc connu $E_{T_{i-1}}$ l'ensemble naturel du segment T_{i-1} et intéressons nous au segment T_i d'origine O_i . Soit A l'arête associée à O_i (elle a été obtenue par la fonction ST) et F les faces incidentes à A .

On obtient ET_i à partir de ET_{i-1} de la façon suivante :

- 1) toutes les faces de ET_{i-1} sont recopiées dans ET_i
- 2) toutes les faces $f \in F \cap ET_{i-1}$ sont éliminées de ET_i
- 3) toutes les faces $f' \in F - ET_{i-1}$ sont ajoutées à ET_i

c'est-à-dire si

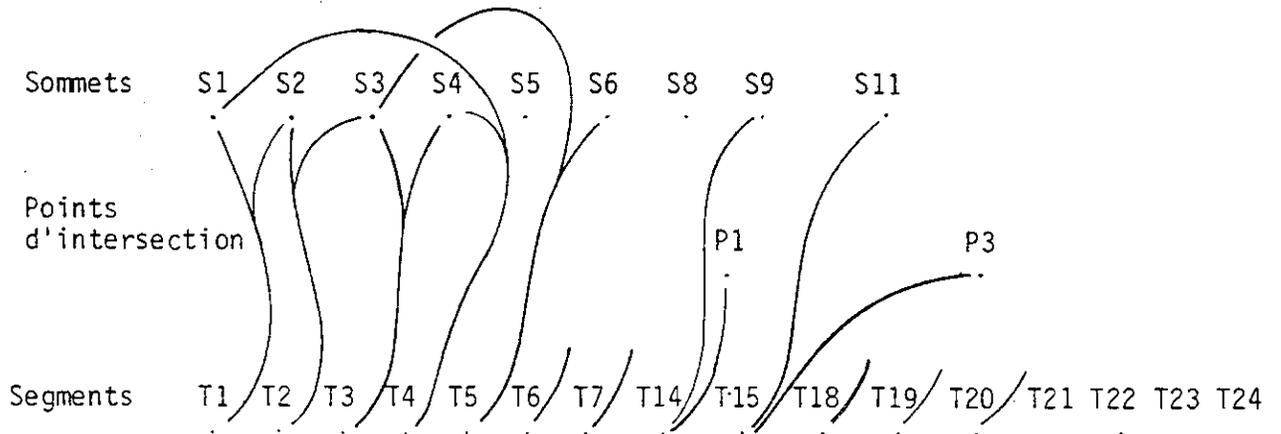
$$X = (f \mid f \in F, f \in ET_{i-1})$$

$$Y = (f' \mid f' \in F, f' \notin ET_{i-1})$$

$$ET_i = ET_{i-1} \cup Y - X$$

Comme le nombre de faces incidentes à une arête est de deux au maximum, l'ensemble naturel ne pourra varier que d'un élément au maximum entre deux segments consécutifs d'une même arête.

V



$$V = I5 = FV (I4)$$

Figure 4.14

Exemple 1 :

Pour déterminer la visibilité du segment T10, le seul segment d'arête A10 (voir figures 4.8 et 4.11), on applique la fonction FD qui détermine l'ensemble naturel du segment. L'origine du segment (sommet S8) a un ensemble naturel vide mais l'arête A10 coupe un nombre pair de fois (zéro fois), le polygone délimitant la face contenue dans l'ensemble naturel de l'extrémité de l'arête (sommet S7); alors l'ensemble naturel du segment T10 contient cette face. Ce segment ne sera donc pas vu par l'observateur.

Exemple 2 :

Le parcours des segments de l'arête AI4 se fait du sommet S11 vers le sommet S10 .

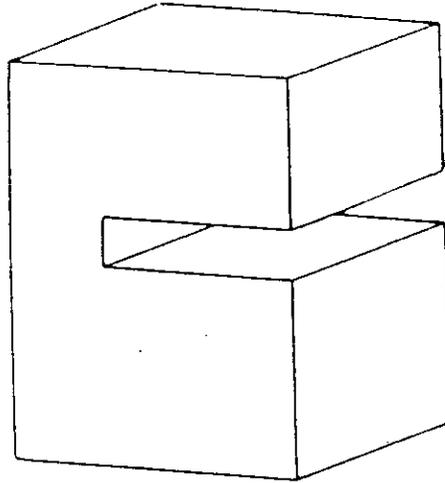
La fonction FD calcule un ensemble naturel du segment T15 (allant du sommet S11 au point d'intersection P3). comme étant vide. Ce segment sera donc vu par l'observateur. Puis on compare les faces délimitées par l'arête associée au point d'intersection P3 , la seule face délimitée est F2 et elle n'est pas contenue dans l'ensemble naturel du segment T15 alors pour obtenir l'ensemble naturel du segment T16 (allant de P3 vers P2) on ajoute F2 .

Le segment T16 n'est pas vu par l'observateur car son ensemble naturel contient la face F2 . Pour calculer l'ensemble naturel du segment T17 on analyse les faces F2 et F9 délimitées par l'arête associée au point d'intersection P2 . La face F2 est contenue dans l'ensemble naturel donc on l'élimine et on ajoute F9 . Le segment T17 allant de P2 à S10 ne sera non plus vu par l'observateur.

Il faut noter que la cohérence du modèle a des répercussions à ce niveau; ainsi, si un segment T_i a pour extrémité un point d'intersection dont le nombre de faces incidentes à l'arête associée est deux, une de ces deux faces est nécessairement contenue dans l'ensemble naturel ET_i et l'autre dans l'ensemble naturel ET_{i+1} du segment suivant.

Ce résultat peut, dans certains cas, simplifier le parcours. Si pour une arête le nombre de segments restant est

inférieur au cardinal de l'ensemble naturel du segment courant, tous les segments restants seront non vus. En effet, l'ensemble naturel ne pouvant varier de plus d'un élément d'un segment au suivant, il ne pourra jamais être vide.



Représentation finale du modèle

Figure 4.15

Algorithme de parcours des arêtes d'I 4

LIGCAC

Réel P

Entier K, ET, ES, F1, F2, NINT, S1, S2

Tableaux ET, ES, P

Ti est l'i- ième segment de l'arête courante

Ai est l'i- ième arête d'I4

Esi est l'ensemble naturel du sommet Si

Eti est l'ensemble naturel du segment Ti

Ki = -card ETi est le degré de visibilité du segment i

Pi est l'i- ième intersection de Ai

F1 est une face adjacente à l'arête associée au point d'intersection

F2 est la deuxième face adjacente (si elle existe)

NINT est le nombre d'intersection de l'arête i

S1 sommet origine de l'arête i

S2 sommet extrémité de l'arête i

FD (i) calcule l'ensemble naturel du premier segment de l'arête i

AREASS (i) rend l'arête associée au point d'intersection i

FACADJ (J, F1, F2) rend les F1, F2 faces adjacentes à l'arête j

Pour chaque arête i de I4 faire

ET1 = FD (i)

K1 = - card ET1

Si K1 = 0 Alors

Si NINT = 0 alors (segment $\overline{S1 S2}$ vu)

 On trace segment $\overline{S1 S2}$ (toute l'arête est vue)

 Return

Sinon ($\overline{S1 P1}$ vu)

 On trace segment $\overline{S1 P1}$ (1er segment de l'arête est vu)

Si NINT < ABS (K1) Return (aucun segment de l'arête n'est vu)

Pour N = 1 Jusque NINT faire

L := AREASS (N)

FACADJ (L, F1, F2) (on récupère les faces adjacentes à l'arête L)

Si F1 ∈ ETn Alors

Si F2 existe alors

ETn+1 = ETn - F1 + F2

Sinon

ETn+1 := ETn - F1

Kn+1 := KN + 1

Si Kn+1 = 0 Alors

Si N < NINT alors

on trace segment Pn Pn+1

Sinon on trace segment Pn S2

Sinon

Si F2 existe alors

ETn + 1 := ETn - F2 + F1 (F2 est contenu
forcément dans ETn)

Sinon

ETn + 1 := ETn + F1 (on ajoute F1 à l'en-
semble naturel)

Kn + 1 := Kn - 1

Si NINT - N < ABS (Kn + 1) Return (tous les
segments restants seront non vus)

Fin pour ;

Fin LIGCAC ;

4.5 - Complexité de l'algorithme

La complexité de l'algorithme que nous venons d'exposer est, comme on l'a déjà dit tout au cours de ce chapitre, fonction de la complexité de la scène, ainsi, dans le meilleur des cas, la complexité est de

$$O(N)$$

si le tri réalisé sur tous les sommets, l'algorithme d'intersection de segments et l'algorithme de comparaison de sommets et polygones sont de complexité minimale.

La complexité est dans le plus mauvais cas de

$$O(N^2)$$

si une comparaison deux à deux de tous les éléments était nécessaire.

Notons que la complexité ci-dessus n'est jamais atteinte, on ne compare dans aucun cas tous les éléments deux à deux, il y a toujours des arêtes ayant un sommet en commun, des faces qui partagent une arête, etc. car il existe une cohérence dans la scène à laquelle s'appliquent les algorithmes.

Il faut noter que cet algorithme ne pénalise pas les scènes les plus simples avec une complexité $O(N^2)$

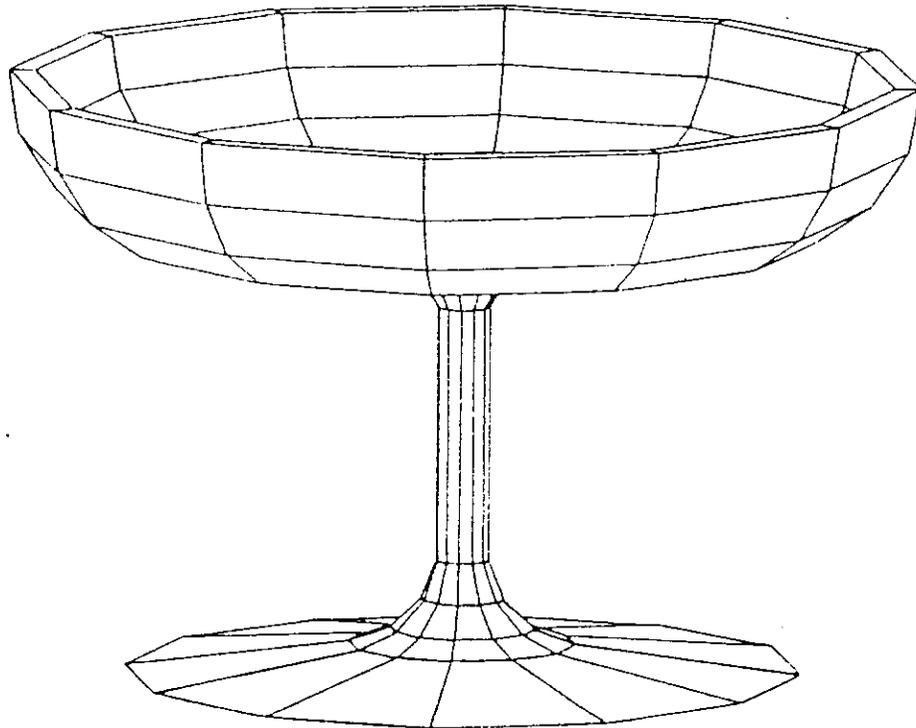
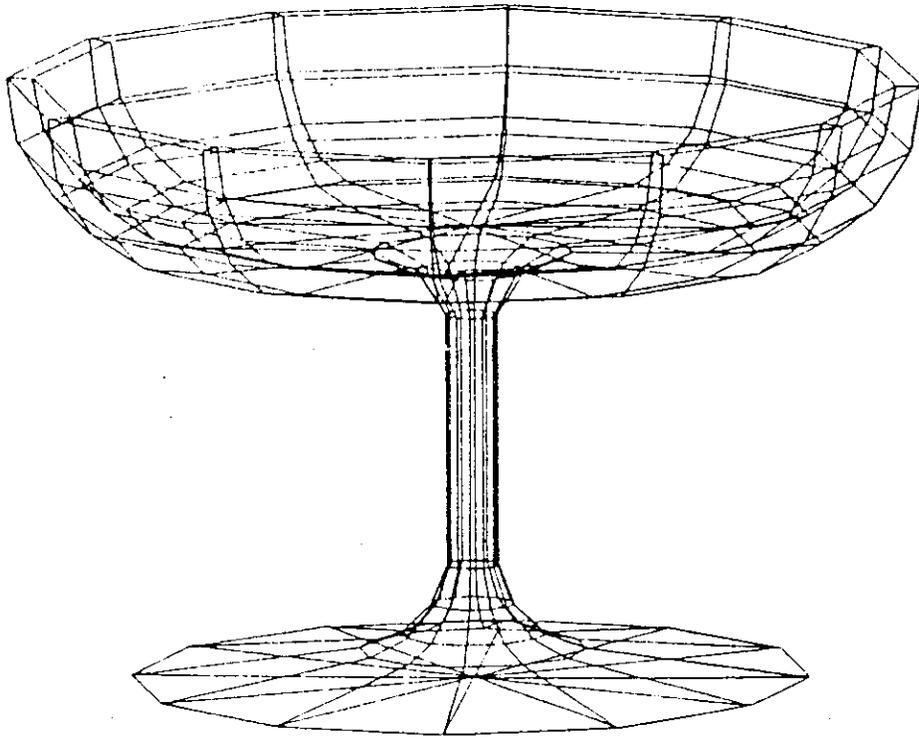
4.6 - Conclusion

Nous avons présenté un algorithme de visualisation de polyèdres avec élimination des parties cachées fournissant une image calculée avec une grande précision.

Les calculs réalisés dans l'espace-objet sont indépendants des dispositifs de visualisation : traceurs ou écrans.

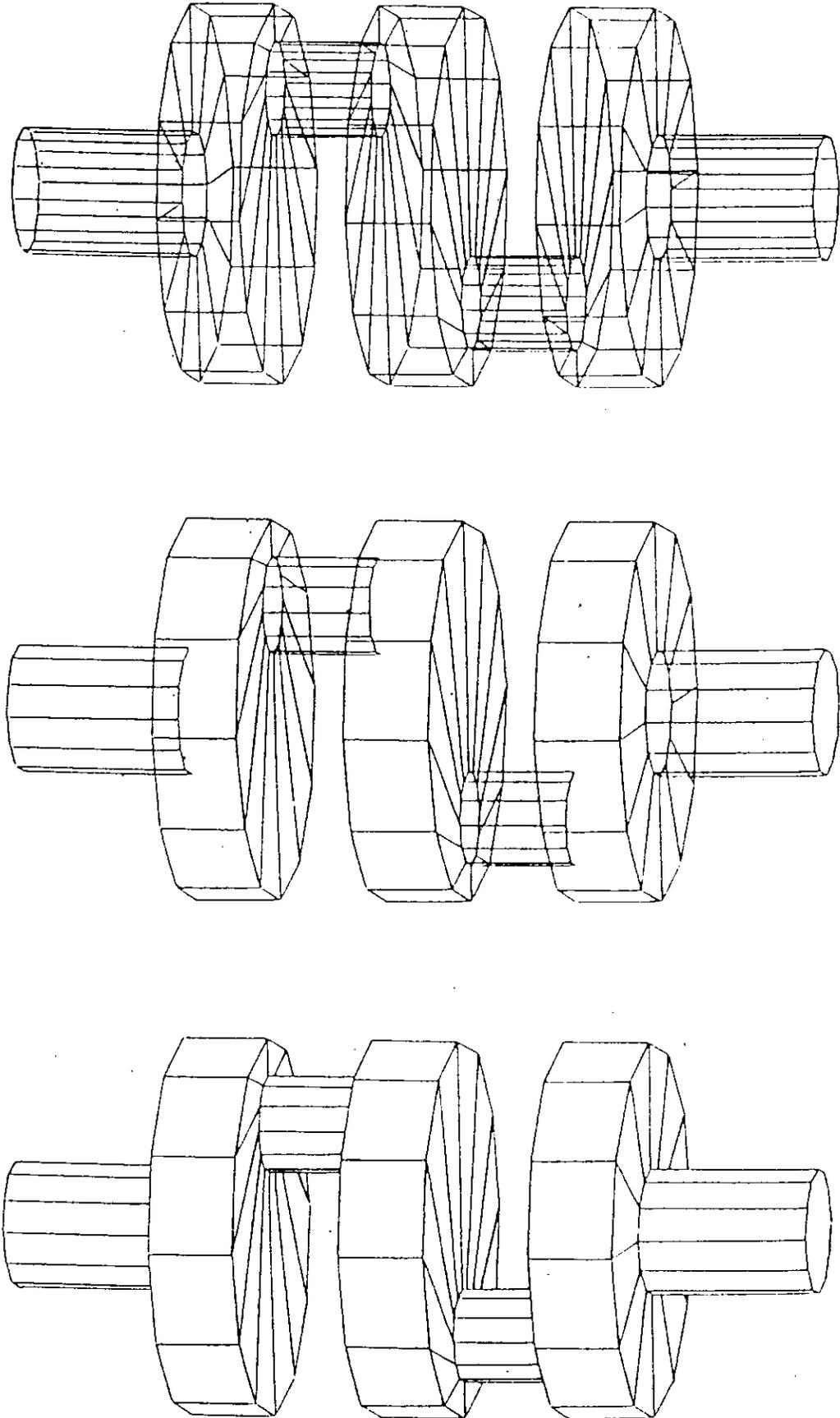
L'algorithme proposé a un temps de réponse suffisamment court pour notre application. Ceci nous a permis de conserver la haute interactivité que nous nous étions fixée comme essentielle à un modéleur géométrique.

Une implantation a été réalisée sur un calculateur PDP-1134 (mots de 16 bits). Le programme comprend trois "overlays" faisant au total 31 k mots, son exécution nécessite 25 k mots mémoire (données comprises).



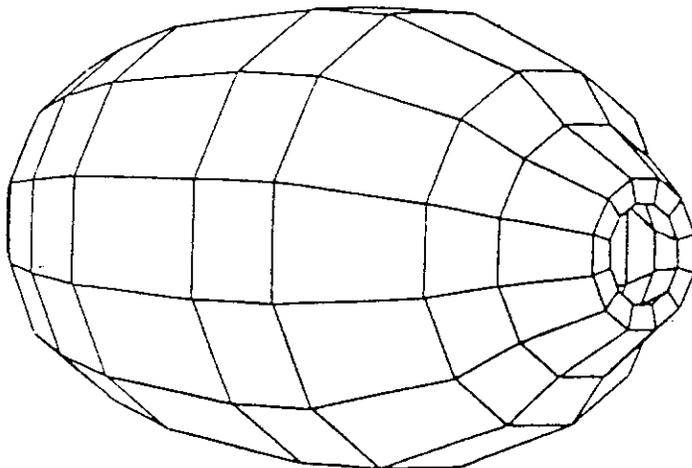
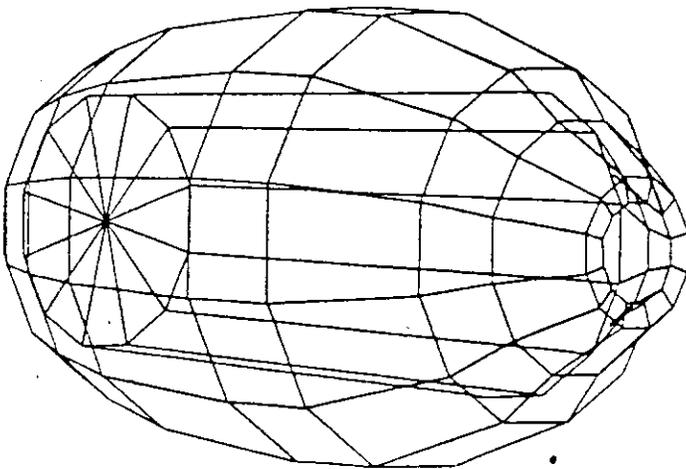
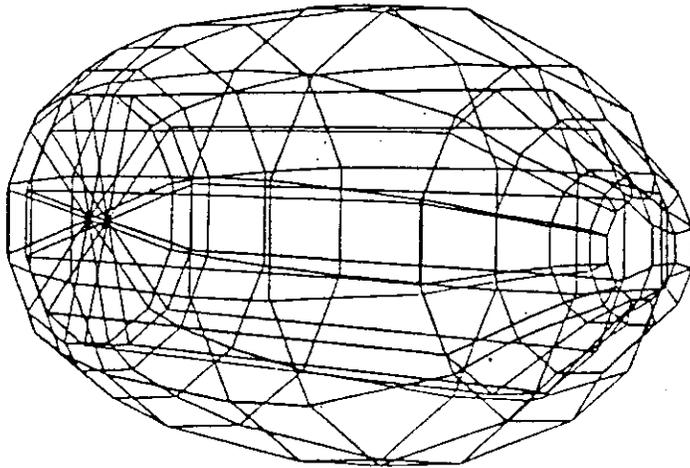
vue calculée en 3.5 secondes

figure 4.16



vue calculée en 2.3 secondes

figure 4.17



vue calculée en 3.3 secondes

figure 4.18

CHAPITRE V

ALGORITHMES GEOMETRIQUES EFFICACES

" ... Le second, de diviser chacune des difficultés que j'examinerais, en autant de parcelles qu'il se pourrait et qu'il serait requis pour les mieux résoudre... "

DESCARTES

Discours de la méthode

5. La conception d'un modèle requiert un grand nombre d'algorithmes géométriques afin de traduire les actions de l'opérateur de façon cohérente et de calculer une représentation graphique des résultats obtenus. Du temps de réponse de ces algorithmes dépend l'interactivité du système.

La plupart des constructions géométriques applicables aux polyèdres (union, différence, lignes cachées etc) requièrent une comparaison exhaustive de tous les éléments (faces, arêtes, sommets). Les algorithmes naïfs correspondants ont donc une complexité en $O(N^2)$ ce qui dans le cas d'un système interactif est trop coûteux en temps.

Nous allons présenter des algorithmes efficaces pour la modélisation géométrique et tout particulièrement lors de la visualisation de polyèdres avec élimination des parties cachées.

5.1 - Test minimax.

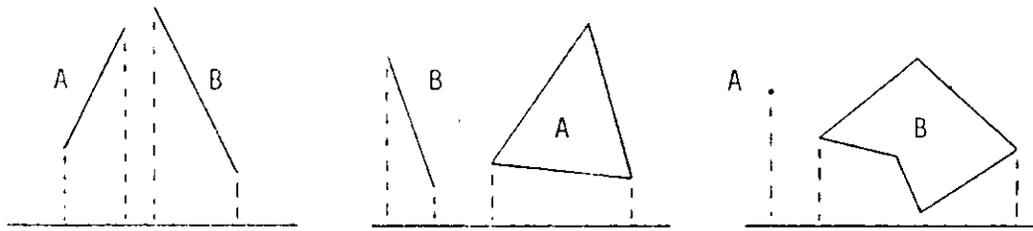
Le test minimax permet de comparer des segments, points ou polygones coplanaires et ainsi de savoir rapidement si deux de ces éléments (segment, point ou polygone) ne peuvent pas entrer en conflit (s'intersecter par exemple).

Le test minimax consiste à comparer les "valeurs extrêmes" de deux éléments suivant une direction privilégiée (couramment la direction des axes de coordonnées).

a) Test minimax suivant la direction de l'axe OX.

Si la coordonnée maximale en x d'un élément est

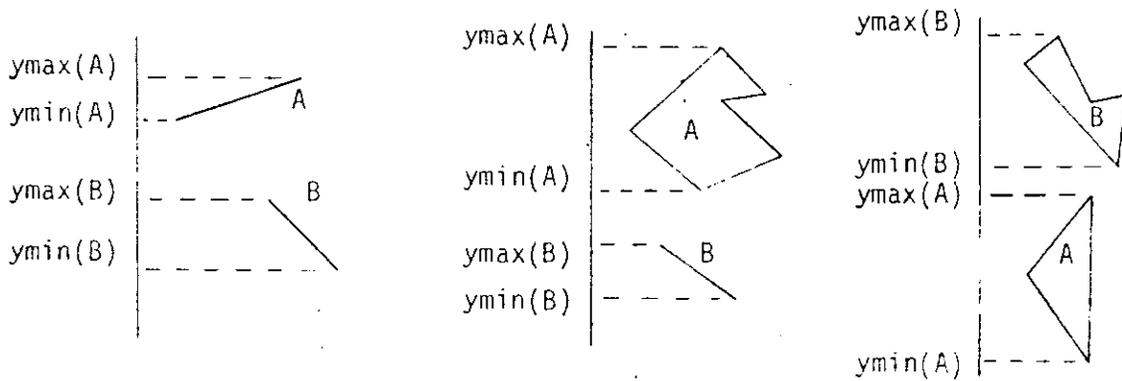
plus petite que la coordonnée minimale en x d'un autre, ces deux éléments ne peuvent pas avoir des conflits.



Si $x_{\max}(A) < x_{\min}(B)$ ou si $x_{\max}(B) < x_{\min}(A)$
les éléments ne peuvent pas s'intersecter

Figure 5.1

b) Test minimax suivant la direction de l'axe Oy



Si $y_{\max}(A) < y_{\min}(B)$ ou si $y_{\max}(B) < y_{\min}(A)$
les éléments ne peuvent pas s'intersecter

Figure 5.2

c) Test de boîtes

Vus ensemble, les tests ci-dessus sont équivalents à la comparaison des rectangles minima de côtés parallèles aux axes Ox et Oy contenant chaque élément.

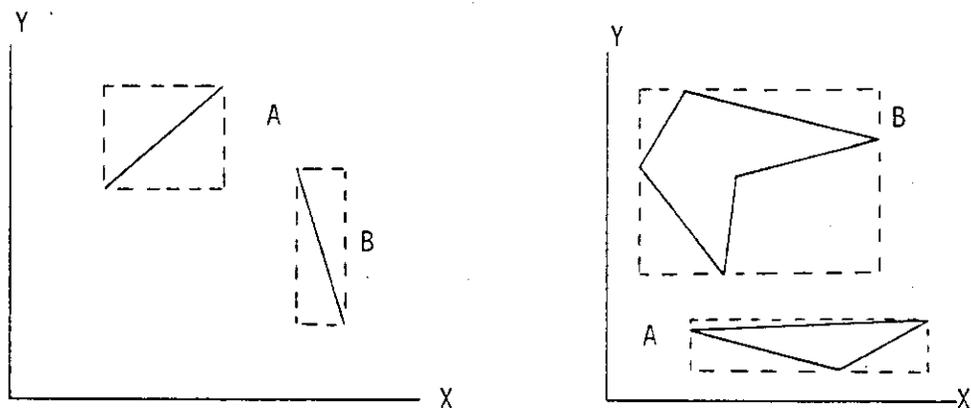


Figure 5.3.

5.2 - Comparaison segment-segment

Soit deux segments A et B coplanaires déterminer s'ils s'intersectent.

La méthode consiste à calculer le signe des distances entre les deux extrémités du segment A et la droite portant le segment B. Si les deux signes sont égaux les segments ne se coupent pas.

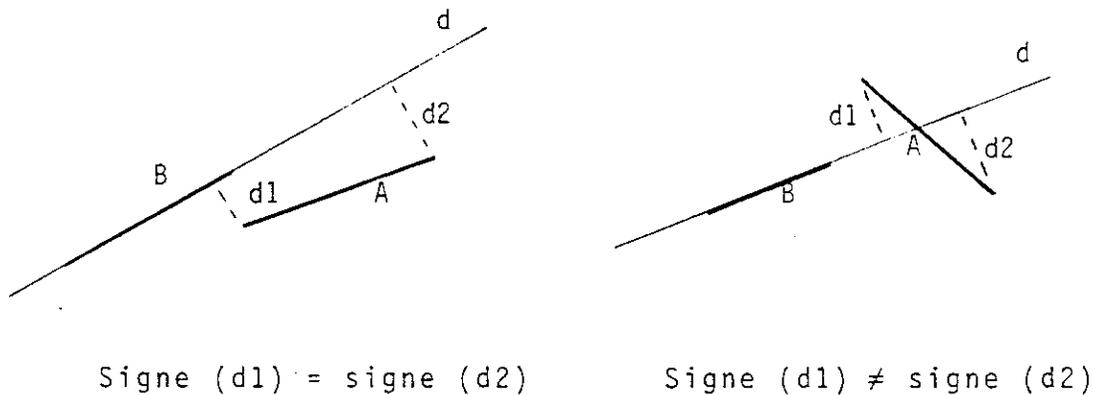
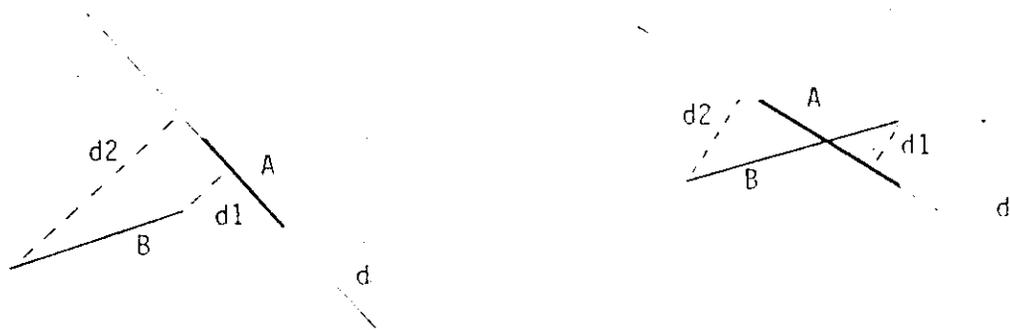


Figure 5.4

Si les signes calculés ne sont pas égaux un deuxième test est nécessaire : on calcule le signe des distances entre les deux extrémités du segment B et la droite portant

le segment A. Si ces deux signes sont égaux les segments ne se coupent pas, sinon ils se coupent.



Pas d'intersection

A et B s'intersectent

Figure 5.5

Intersection de deux segments

Logical Fonction I CROIX (P1, P2, P3, P4)

Réel P1, P2, P3, P4

P1 (x1,y1) et P2 (x2, y2) sont les points extréma du segment A

P3 (x3,y3) et P4 (x4, y4) sont les points extréma du segment B

I CROIX = VRAI si A et B s'intersectent.

Début

I CROIX := FAUX

A := (y2 - y1) / (x2 - x1)

C := y1 - A . x1

D1 := A . x3 - y3 + C

D2 := A . x4 - y4 + C

Si signe (D1) = signe (D2) alors

Return (A et B ne se coupent pas)

Sinon

A := (y4 - y3) / (x4 - x3)

C := y3 - A · x3

D1 := A · x1 - y1 + C

D2 := A · x2 - y2 + C

Si signe (D1) = signe (D2) alors

Return (A et B ne se coupent pas)

Sinon

I CROIX := VRAI

Return (A et B se coupent)

Fin I CROIX;

5.3 - Comparaison point-polygone

Soit un point Q et un polygone P dans un même plan, déterminer si le point Q est inclus dans le polygone P.

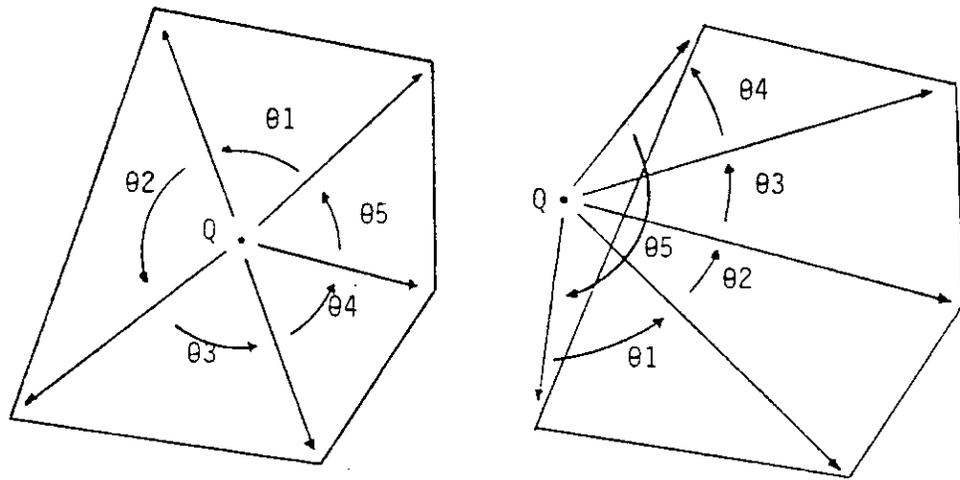
Deux méthodes peuvent être envisagées :

a) : Test sur la somme des angles

On définit un système de coordonnées polaires de la façon suivante :

- On choisit le point Q comme pôle
- Soit P₁ un sommet du polygone, on prend $\overrightarrow{QP_1}$ comme axe polaire

Le test envisagé requiert le calcul pour chaque sommet P_i du polygone, de l'angle du vecteur $\overrightarrow{QP_i}$ avec $\overrightarrow{QP_{i-1}}$. La somme de ces angles est toujours nulle ou multiple de 2π . Si la somme est nulle le point Q est extérieur au polygone sinon il est intérieur.



$$\Sigma \theta = 2\pi$$

Le point Q est inclus dans P

a

$$\Sigma \theta = 0$$

Le point Q n'est pas inclus dans P

b

Figure 5.6

Un cas particulier peut cependant se présenter : la somme des angles est égale à π , le point Q se trouve alors sur le polygone.

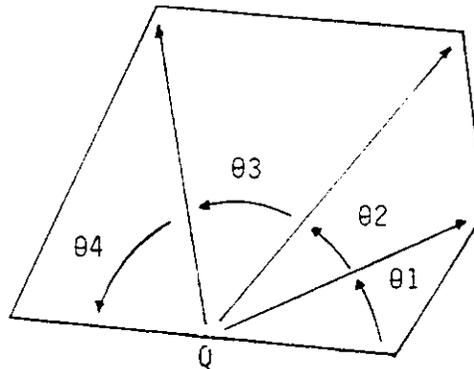
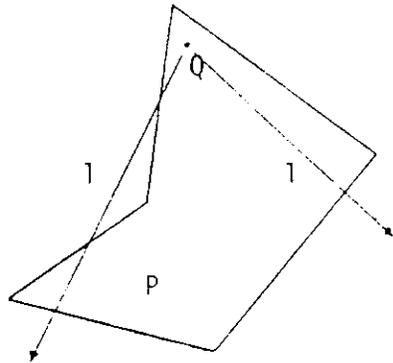


Figure 5.7

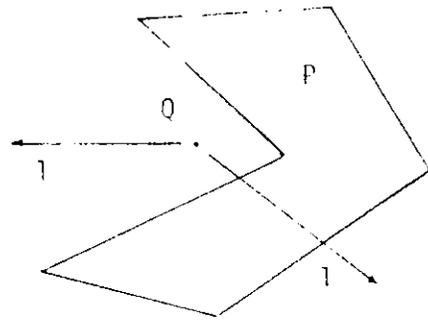
b) Test sur le nombre d'intersections

On calcule le nombre d'intersections entre une demi-droite l passant par Q avec tous les côtés du polygone.

Si le nombre d'intersections est pair le point Q est extérieur au polygone, autrement il est intérieur.



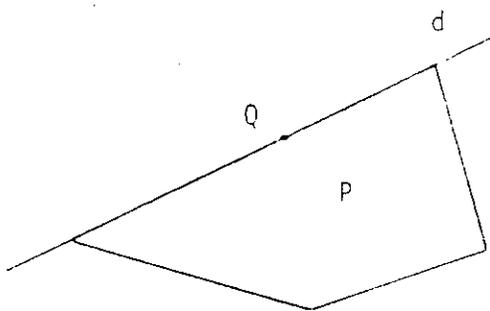
Le nombre d'intersections
est impair
 $Q \in P$



Le nombre d'intersections
est pair
 $Q \notin P$

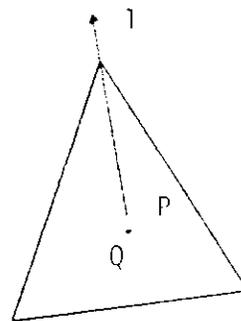
Figure 5.8

Ce test présente deux inconvénients : un test supplémentaire est nécessaire pour déterminer si Q se trouve sur le polygone P et deuxièmement, des cas particuliers peuvent se présenter quand la demi-droite l passe par un sommet de P .



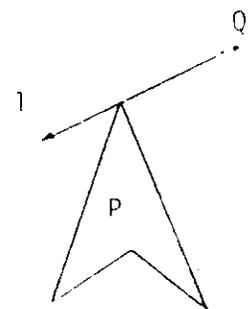
Le point Q se trouve sur
le polygone

a



Nombre impair
d'intersections

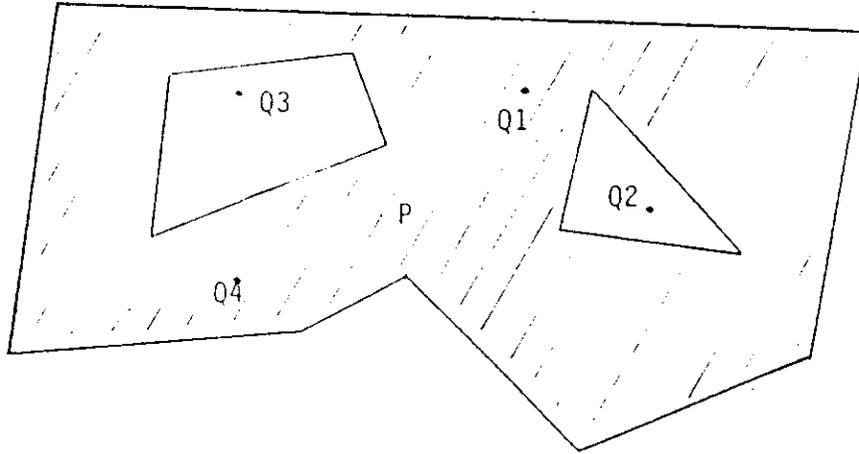
b



Nombre pair
d'intersections

c

Figure 5.9



Q1 et Q4 sont inclus dans P

figure 5.10

Test d'inclusion d'un point dans un polygone

LOGICAL FONCTION I DANS (PT, POLYG, K)

REEL POLYG, PT, PEXT

TABLEAU POLYG, PEXT, PT

ENTIER K, NDINT

PT est le point à tester; POLYG contient K+1 sommets du K-Gone, le dernier est confondu avec le premier; ICROIX est une fonction qui teste l'intersection de 2 segments, ICROIX :=TRUE. S'ils se coupent; PEXT contient un point non inclus dans le polygone (infini p. ex); MOD est une fonction qui rend le module 2 d'un entier.

```
Début      NDINT := 0
            IDANS := FAUX
            Pour I := 1 jusque K faire
                Si ICROIX (PT, PEXT, POLYG (K), POLYG (K+1)=TRUE ALORS
                    NDINT := NDINT + 1
                Fin pour
                Si MOD (NDINT) = 0 alors
                    RETURN (le point est dehors)
                Sinon
                    IDANS := VRAI
                    RETURN (le point est dedans)
            Fin IDANS
```

5.4 Tri par ventilation

Etant donné N points P coplanaires, trier ces points selon les abscisses croissantes

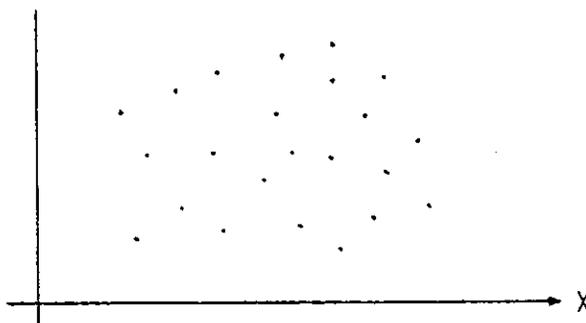


Figure 5.11

Cette phase de tri est cruciale, les performances de l'algorithme de lignes cachées dépendront essentiellement de son efficacité. Pour cette raison nous avons développé une version de tri par ventilation, de façon à obtenir une complexité moyenne linéaire.

- Etape 1
- Normalisation.

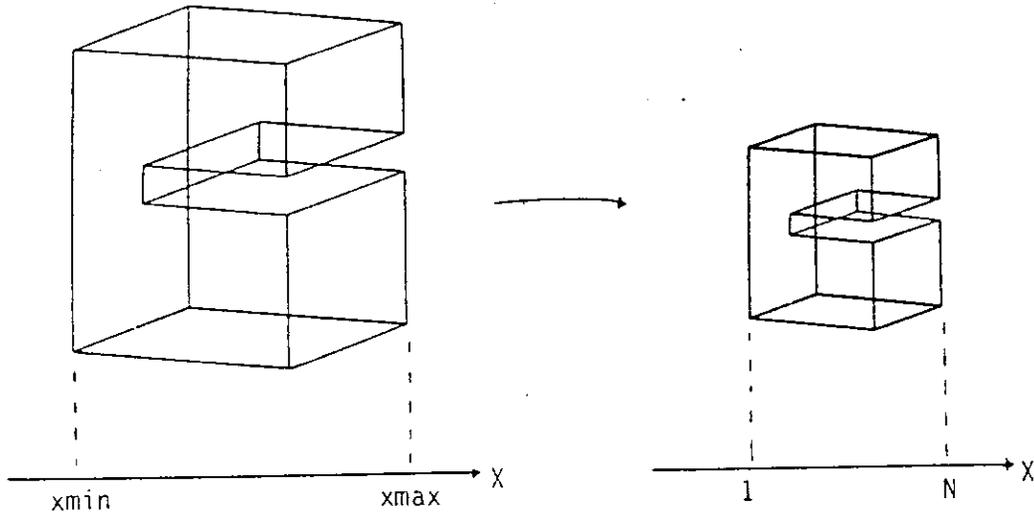
Soit

$$x_{\max} = \max (x (P_i))$$

$$x_{\min} = \min (x (P_i))$$

Par changement d'échelle on se ramène au cas :

$$x_{\min} = 1 \text{ et } x_{\max} = N$$



$$\bar{x}_i = \frac{N-1}{x_{\max}-x_{\min}} (x_i - x_{\min}) + 1$$

Figure 5.12

- Etape 2

On note $E(x)$ la partie entière de x .

Dans le meilleur des cas les points P_i se répartiront de telle sorte que

$$E(x(P_i)) \neq E(x(P_j)) \quad i \neq j \in [1, N]$$

Dans ce cas idéal à tout entier $i \in [1, N]$ correspond un point P_j et un seul tel que

$$i = E(x(P_j))$$

et on pourra alors "ventiler" P_j dans la case R_i d'une zone de rangement R .

Evidemment cette situation est rare et en général

pour une valeur i donnée plusieurs points risquent d'entrer en collision. Il faudra alors calculer tout d'abord pour chaque entier $i \in [1, N]$ le nombre de collisions C_i c'est-à-dire le cardinal de l'ensemble S_i

$$C_i = \text{Card } S_i$$
$$S_i = \left\{ P_j \mid P_j \in P, x(P_j) = i + E, 0 \leq E < 1 \right\}$$

Connaissant C_i on peut calculer A_i définis par

$$A_1 = 1$$

$$A_i = A_{i-1} + C_{i-1} \quad 2 \leq i \leq N$$

A_i fournit en fait un pointeur sur le tableau R . On rangera à partir de $R(A_i)$ et à la suite les uns des autres les points P_j vérifiant.

$$E(x(P_j)) = i$$

- Etape 3

On utilise ensuite un algorithme de tri classique (par exemple par comparaison) pour classer chaque famille de points résultant des collisions.

— Complexité de l'algorithme

La complexité de l'algorithme de tri par projection est dans le meilleur des cas de

$$O(N)$$

et dans le pire des cas

$$O(N \log N)$$

La complexité minimale est obtenue si les N points sont "ventilés" sans collision dans le tableau R ; la complexité est maximale si tous les N points sont en collision. Le coût

de ce tri est donc fonction de la complexité de la scène et
la place mémoire maximale nécessaire pour effectuer ce tri est
de $2N$.

Tri par ventilation de N points

TRI (POINT, N)

REEL POINT, R

ENTIER ITAB

TABLEAUX ITAB (N), R (N), POINT (N)

POINT contient les N points à trier; ITAB est une liste de travail initialisée à zéro; R contient les N points triés. E (Xi) est une fonction qui rend la partie entière du réel xi
TRISOM (panier, N) ordonne les N points contenus dans PANIER.

Début

Pour chaque P (xi, yi) faire

I := E (xi)

ITAB (I) = ITAB (I) + 1

Fin pour

L = ITAB (1)

ITAB (1) = 1

Pour K = 2, N faire

J = ITAB (K)

ITAB (K) = ITAB (K-1) + L

L = J

Fin pour

Pour chaque P (xi, yi) faire

I = ITAB (E (xi))

RTAB (I) = P (xi, yi)

ITAB (I) = ITAB (I) + 1

Fin pour

Pour chaque panier M faire

J = ITAB (M) - ITAB (M-1)

TRISOM (RTAB (ITAB (M-1)), J)

Fin pour

Fin TRI;

5.5 - Algorithme d'intersection de segments

- Soit N segments coplanaires, déterminer tous les points d'intersection (autres que les extrémités) des segments pris deux à deux.

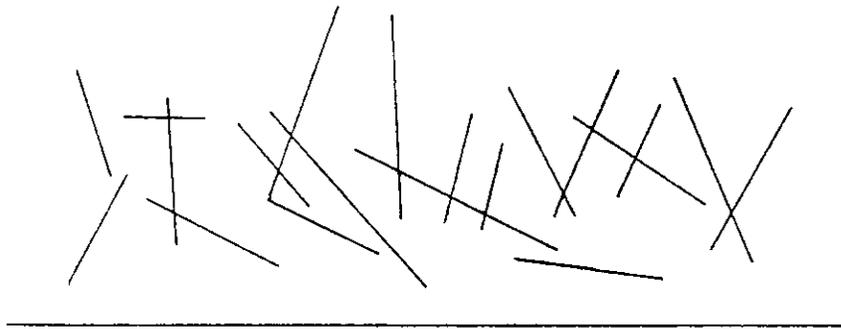


Figure 5.13

Si tous les segments s'intersectent deux à deux, il est évident que le coût du calcul des points d'intersection sera nécessairement $O(N^2)$. Nous allons donner un algorithme capable de déterminer ces $n(n-1)/2$ intersections si elles existent mais sans pour cela pénaliser les situations plus simples. L'algorithme présenté ici est une extension de celui proposé par Dan Hoey.

Les segments sont considérés comme orientés; on distingue donc leur origine O et leur extrémité E et bien sûr :

$$x(O) \leq x(E)$$

- Etape 1

On trie d'abord les origines et extrémités des N segments. La relation d'ordre (partiel) utilisée est la suivante :

Etant donné 2 points 01 et 02

$$01 < 02 \iff x(01) < x(02)$$

Pour ce faire on utilise le tri par ventilation décrit au paragraphe (5.4) et on obtient ainsi une liste ordonnée de points que l'on note S.

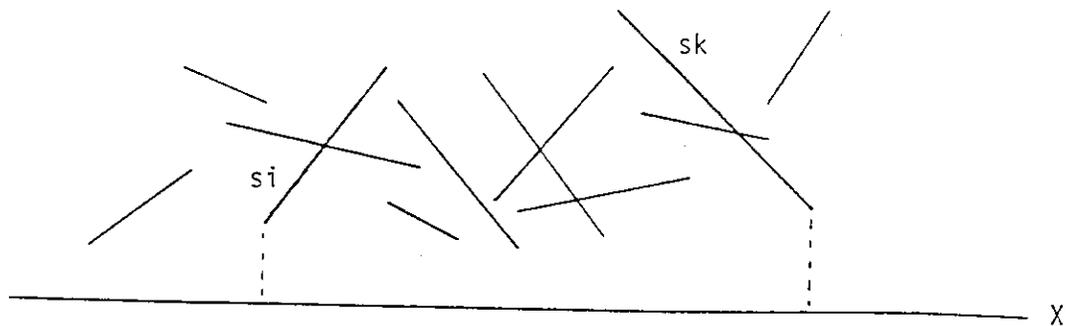
- Etape 2

— On initialise à vide une liste L. On parcourt S et on note i l'indice courant sur S et P_i le point correspondant ($1 \leq i \leq N$)

— On élimine de L tous les segments ayant P_i pour extrémité.

Pour tous les segments s d'origine P_i on calcule les intersections éventuelles de s avec les segments contenus dans L.

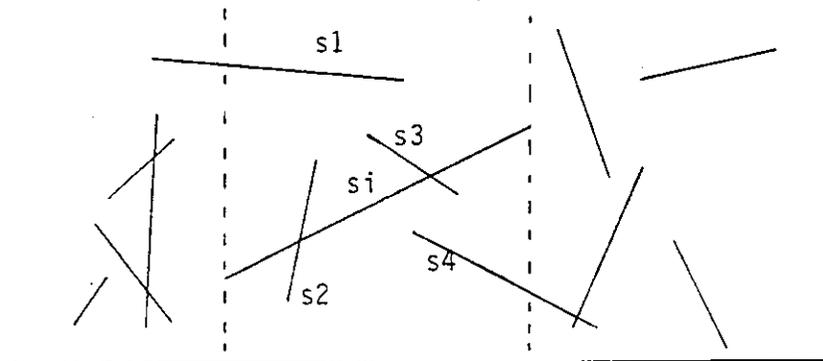
— On ajoute s à L.



On insère le segment s_i on élimine le segment s_k

Figure 5.14

Un segment s ne peut donc avoir d'intersections qu'avec les segments qui ont partagé avec lui la liste L .



Seuls les segments s_1, s_2, s_3, s_4 peuvent intersecter le segment s_i

Figure 5.15

— Complexité de l'algorithme

Le coût de l'algorithme, tri des points non compris, est déterminé par la complexité de la scène.

Ainsi, dans le cas de la figure 5.16 b, seul un algorithme de complexité $O(N^2)$ permet de calculer toutes les intersections existantes.

La structure de données utilisée est telle que l'on peut pour un point retrouver les segments incidents en un temps constant. Alors, dans le meilleur des cas, quand la liste L ne contient jamais plus d'un segment la complexité est de

$$O(N)$$

Dans le plus mauvais des cas on a comme complexité

$$O(N^2)$$

si une comparaison de toutes les arêtes entre elles s'avère nécessaire.

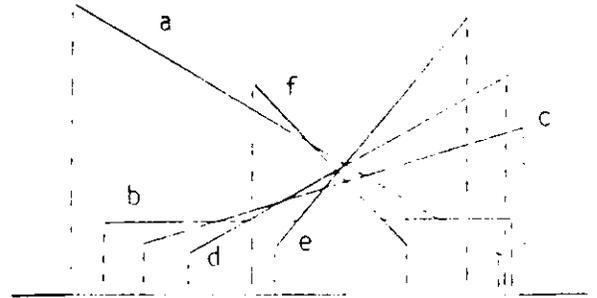
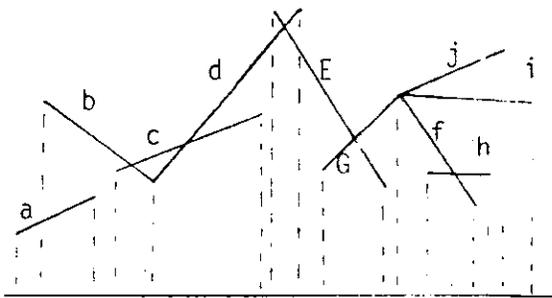
Notons : $\leftarrow x$ l'inclusion du segment x dans la liste

$x \uparrow (a, b, c, \dots)$ la comparaison de x avec a, b, c, \dots

$\rightarrow x$ l'élimination de x de la liste.

Complexité moyenne

Complexité maximale



10 segments 16 extrémités

6 segments 12 extrémités

1. - $\leftarrow a$
2. - $b \uparrow (a); \leftarrow b$
3. - $\rightarrow a$
4. - $c \uparrow (b); \leftarrow c$
5. - $\rightarrow b; d \uparrow (c); \leftarrow d$
6. - $\rightarrow c$
7. - $E \uparrow (d); \leftarrow E$
8. - $\rightarrow d$
9. - $G \uparrow (E); \leftarrow G$
10. - $\rightarrow E$
11. - $\rightarrow G; \leftarrow F; \leftarrow i; \leftarrow j$
12. - $h \uparrow (f, i, j); \leftarrow h$
13. - $\rightarrow f$
14. - $\rightarrow h$
15. - $\rightarrow j$
16. - $\rightarrow i$

1. - $\leftarrow a$
2. - $b \uparrow (a); \leftarrow b$
3. - $c \uparrow (a, b); \leftarrow c$
4. - $d \uparrow (a, b, c); \leftarrow d$
5. - $f \uparrow (a, b, c, d); \leftarrow f$
6. - $e \uparrow (a, b, c, d, f); \leftarrow e$
7. - $\rightarrow f$
8. - $\rightarrow e$
9. - $\rightarrow a$
10. - $\rightarrow d$
11. - $\rightarrow b$
12. - $\rightarrow c$

a

Figure 5.16

b

Intersection de segments

INTSEG

REEL S, SP

TABLEAUX S (N), SP (N)

N est le nombre de segments

M est le nombre de points extrêmes des segments $M \leq 2 N$

S contient les N segments

SP est une liste de travail

GETSEG (I, S, N) récupère tous les N segments dont l'i-ème point est leur origine.

S (i) c'est le i-ème segment candidat à la liste SP.

NPIL c'est le nombre de segments à un instant donné dans la liste SP

ELIM (i) élimine de la liste SP tous les segments dont l'i-ème point est leur extrémité.

INTER (SP (i), s (k)) compare l'i-ème segment avec le x-ème segment candidat et détermine s'il s'intersectent INTER =VRAI s'ils se coupent

INSRT (I) ajoute dans la liste SP tous les segments dont l'i-ème point est leur origine

Début

```

  Pour chaque I point extrême faire
    ELIM (I) (on élimine les segments de la LISTE SP)
    GETSEG (I, S, N) on récupère les N segments candidats
      à la liste)
      si N > 0 alors
        pour j = 1 jusque N faire
          pour k = 1 jusque NPIL faire
            si INTER (s (j), SP (k)) alors
              On calcule l'intersection de
                s(j) avec SP (k)
          Fin pour;
        Fin pour;
      Fin pour;
    INSRT (I) (on ajoute les segments dans la liste )
  Fin pour
Fin GETSEG
```

5.6 - Algorithme de comparaison de points et polygones

Soit K polygones ayant un total N sommets, déterminer pour chaque sommet les polygones qui le contiennent.

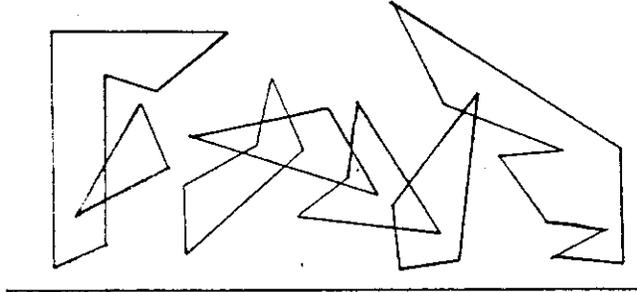


Figure 5.17

— Analyse naïve

On compare chaque sommet à chaque polygone. Un tel algorithme nécessite $O(KN)$ tests d'inclusion (point intérieur à un polygone).

— Algorithme

Etape 1

On trie les N sommets en utilisant l'algorithme exposé au paragraphe 5.4

Etape 2

On calcule quels sont les sommets d'abscisses minimale $S(0)_j$ et maximale $S(E)_j$ du polygone K_j .

Etape 3

On initialise à vide une liste L de travail et on parcourt ensuite la liste des N sommets triés; notons i l'indice du sommet courant S_i .

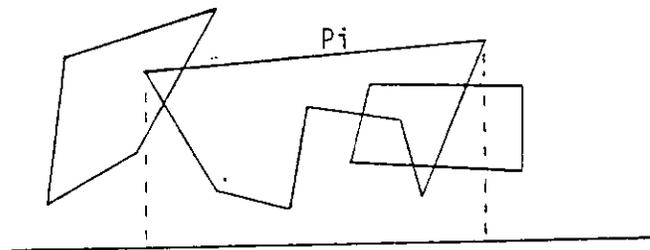
On élimine de L les polygones K_m vérifiant

$$S(E)_m = S_i$$

On compare le sommet S_i avec les polygones dans L. Cette comparaison permet de déterminer s'il y a dans la liste L des polygones pouvant contenir ce sommet.

On ajoute à L les polygones K_n vérifiant

$$S(O)_n = S_i$$

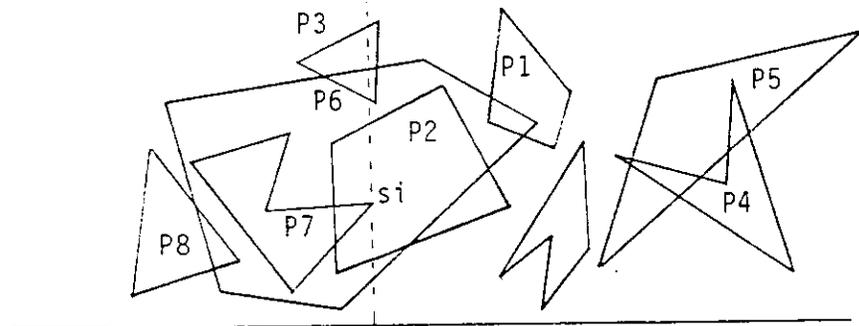


On insère P_i

On élimine P_i

Figure 5.18

Etant donné une perpendiculaire s à l'axe Ox passant par le sommet courant S_i , la liste L contient tous les polygones intersectant s . Le sommet S_i ne peut être contenu que dans les polygones appartenant à L.



Seuls les polygones P_2 , P_7 , P_6 peuvent contenir le sommet S_i

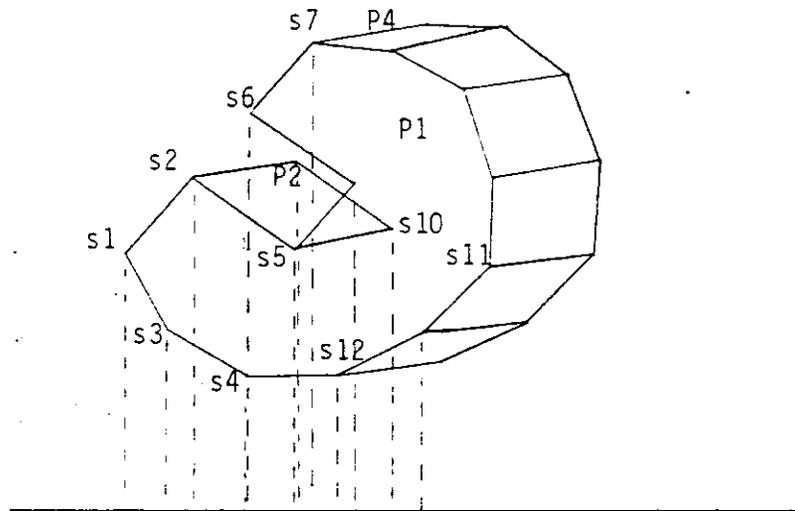
Figure 5.19

— Complexité de l'algorithme

La complexité de l'algorithme dépend de la scène. La structure de données utilisée permet pour chaque sommet d'accéder aux polygones incidents en temps constant. Alors, l'algorithme proposé tri non compris est d'une complexité minimale de

$$O(N)$$

En gardant la même notation de la figure 5.16 on a :



1. - ← P1
2. - S3 ↑ (P1)
3. - S2 ↑ (P1); ← P2
4. - S4 ↑ (P1, P2)
5. - S6 ↑ (P1, P2)
6. - S5 ↑ (P1, P2)
7. - S8 ↑ (P1, P2)
8. - S7 ↑ (P1, P2); ← P4
9. - S12 ↑ (P1, P2, P4); ← P3
10. - S9 ↑ (P1, P2, P4, P3)
11. - → P2; S10 ↑ (P1, P3, P4)
12. - S11 ↑ (P1, P3, P4)
13. -
14. -

Figure 5.20

Si la liste ne contient jamais plus d'un polygone

La complexité maximale

$$O(K N)$$

est obtenue si tous les polygones restent dans la liste pendant le parcours des sommets.

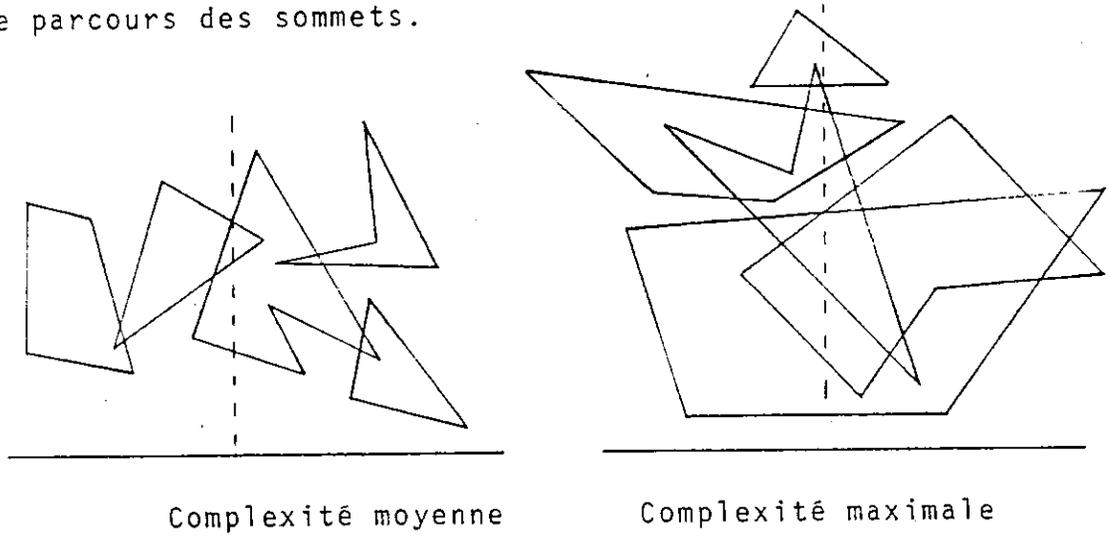


Figure 5.21

CONCLUSION

" Qu'on ne dise pas que je n'ai rien dit de nouveau : la disposition des matieres est nouvelle: quand on joue à la paume c'est une même balle dont jouent l'un et l'autre mais l'un la place mieux ."

PASCAL

Nous avons présenté les travaux menés sur l'étude d'un système de conception de volumes dont l'interactivité joue le rôle fondamental.

Des algorithmes géométriques compatibles avec notre application ont été développés; ceci nous a permis de conserver la qualité de l'interface homme-machine et le haut niveau d'interactivité que nous nous étions imposés comme élément essentiel d'un modèleur géométrique.

Les objectifs fixés au début de ce travail ont été atteints de manière satisfaisante tant au niveau de la puissance du système (une grande variété d'objets peuvent actuellement être créés) qu'à ceux des temps de réponse obtenus et taille mémoire nécessaire sur un minicalculetur.

Le système MGIT n'est pas à l'heure un modèleur géométrique complet; certains algorithmes généraux d'union, intersection, différence et des restrictions de ces algorithmes (perçage etc.) devront par la suite être étendus. Leur développement devra être conforme avec la partie déjà réalisée, c'est-à-dire que l'adjonction de nouvelles fonctions ne doit pas pénaliser le système ni au niveau de l'interactivité ni au niveau de la place mémoire demandée.

Divers utilitaires devront être également développés : en particulier les modules d'habillage des dessins obtenus à partir de vues des objets, les programmes de commande numérique etc...

BIBLIOGRAPHIE

- 1- AHO Alfred V. - HOPCROFT John - ULLMAN J.D.
The Design and analysis of computer algorithms
Addison-Wesley publishing company
- 2- AISH R.
3D Input For Caad Systems Computer Aided design
Vol. 11 n°2, mars 1979, p. 66
- 3- APPEL A.
On calculating the illusion of reality
Proc. IFIP Congr. Hardware
Vol. 2 Booklet E. pp. 72-78
- 4- APPEL A.
The visibility problem and machine rendering of solids
IBM Res. Rep. RC 16-18 May 1966
- 5- APPEL A.
• The notion of quantitative invisibility and the machine rendering
of solids
IBM Watson Research Center
Yorktown Heights N.Y. February 27, 1967
- 6- APPEL A. - STEIN A.
A system for the interactive design of polyhedra
IBM Thomas J. Watson - Research Center N.Y. U.S.A.
- 7- APPEL A.
Modeling in 3D
IBM Systems J. Vol. 7, N° 3, 4 pp. 310-321 1968
- 8- APPLICON-FRANCE
APPLICON - Matériel et logiciel en CAO.
- 9- ARCHULETA M.
Hidden surface line-drawing algorithm
AD-762023 - June 1972
- 10- ARMSTRONG G.T. - The Univ. of Leeds.
Geometric Modeling : Industrial input and out put processes
Computer aided drafting and géometric modeling. April 80. London

- 11- ARMSTRONG G.T.
PADL - A constructive solid geometry based geometric modeller
University of Leeds England
- 12- BAER A. - EASTMAN C. - HENRION M.
Geometric Modelling : A survey
C.A.D. Vol. II - 5 sept. 1979, pp. 253-272
- 13- BAUMGART B.G.
A Polyhedron representation for computer vision
Proc. AFIPS Nat. Comput. conf. vol. 44 - 1975, pp. 589-596
- 14- BAUMGART B.G.
Geometric modelling for computer vision
Stanford Artificial Intelligence
Computer Science Departement
Stanford University USA october 1974
- 15- BELLEVAUX C. - CRESTIN J.P.
About Geometrical Computations for cad in mechanical engineering
Prolamat 79 - Mai 1979
- 16- BELLEVAUX C. - GARRIDO J.A.
Construction interactive de volumes
3ème journée scientifique
Ct Techniques de l'ADEPA -Toulouse, juin 1981
- 17- BELLEVAUX C. - SMADJA M.D. - FATHI M.
Système de construction géométrique
2ème journée scientifique et technique de l'ADEPA
Nancy juin 1979
- 18- BENTLEY J.L. - FRIEDMAN J.H.
Data structure for range searching
Computing survers Vol. II n°4, déc. 1979
- 19- BENTLEY J.L. - FRIEDMAN J.H.
A Survey of algorithms and data structures for rance searching
Août 1978
- 20- BELTLEY J.L. - HAKEN D. - SAXE J.B.
A general method for solving divide-and-conquer recurrences
Carnegie-Mellon University - déc. 1978

- 21- BENTLEY J.L. - KUNG H.T. - SCHKOLNICK M. - THOMPSON C.D.
On the average number of maxima in a set of vectors
and applications
Feb. 1977
- 22- BENTLEY J. L. - SHAMOS M.
A problem in multivariable statistics : algorithm, data
structure, and applications (extended abstract). 1977
- 23- BENTLEY J.L. - SHAMOS M.
Divide and conquer for linear expected time
Carnegie-Mellon Univ.
Information processing letters. Vol. 7 n°2, Feb. 1978
- 24- BENTLEY J.L.
Multidimensional divide-and-conquer
Carnegie-Mellon Univers.
Communications of the ACM Apris 1980. Vol. 23, p.214
- 25- BLINN James F. - NEWELL Martin
Clipping using homdgenecus coordinates
Computer Graphics. Vol. 12 n°3. Aug. 1978
- 26- BORGERSON B.R. - JOHNSON R.M.
Beyond cad to computer aided engineering
Information processing IFIP 1980
- 27- BOUKNIGHT W.J.- KELLEY K.
An Algorithm for producing half-tone computer graphics
presentations with shadows and movable light sources
Proceedings of the spring joint computer conference
Vol. 36 pp. 1-10
AFIPS Press, May 1970
- 28- BOUKNIGHT W.J.
An improved procedure for generation of halftone computer
graphics representations
Co-ordinated science laboratory
University of Illinois USA, september 1969
- 29- BOURNIGHT W.J.
A procedure for generation of three dimensional half-toned
computer graphics representations
CACM 13, 9, 527. Sept. 1970

- 30- BOYSE John W. General motors research laboratories.
Interferences detection among solids and surfaces
Communication of the ACM. January 1979 Vol. 22 n°1
- 31- BRAID I.C. - LANG C.A.
Computer-Aided design of mechanical components with
volume building bricks
Automation Vol. 10, pp. 635-642. Pergamon Press, 1974
Prolamat 73 Budapest 10-13 apris 73.
- 32- BRAID I.C.
Note on a geometric modeller
CAD Groupe document IDI, juin 1871. Cambridge Univ.
- 33- BRAID I.C.
Designing with volumes : computation of weight...
Cad groupe document n°81. Univ. of Cambridge
- 34- BRAID I.C.
Designing with volumes
Ph. D. dissertation
Contab Press, England. 1974.
- 35- BRAID I.C. - HILLYARD R.C. - STROUD I.
Stepwise construction of polyhedra in geometric modelling
Cad group document n°100. Cambridge Univ.
Computer Laboratory.
- 36- BRAID I.C. 1976
A New shape desing system
CAD Group document n°88
Cambridge Univ. Computer laboratory
- 37- BRAID I.C.
Geometric modelling ten years on
Computer Laboratory. Cambridge England. 7 nov. 1979.
- 38- BRAID I.C.
New directions in geometric modelling (BUILD)
Cambridge, march, 1978
- 39- BREWER J.A. - ANDERSON D.C.
Techniques for interactive three dimensional design
User oriented design of interactive graphics systems
Nov. 1978, pp.13-28

- 40- BRUN J.M. - BAUDSON C. - DALLET J. - ISABELLE D. - THERON M.
Animation de formes spatiales par ordinateur
Xème congrès UNIATEC-BERLIN, sept. 1972
- 41- BRUN J.M.
Le concept de filtrage progressif et l'élimination des parties cachées dans EUCLID
- 42- BURNS R.A.
Simulating the visual environment in real time via software
Proc. AFIPS. Conf. Vol. 41, 1972 pp. 153-168
- 43- BUTHIÒN M.
Un programme qui résout formellement des problèmes de construction géométrique.
Rairo informatique. Vol. 13, n°1 1979 pp. 73-106.
- 44- BUTLAND J.
Surface drawing made simple
CAD. Vol. II-I - janv. 1979, pp. 19-22
- 45- CAD 1976
Second international conference on computers in engineering and building design
Imperial collège London. March 1976
- 46- CHATAVAL Y.
A combinatorial theorem in plane geometry
J. Comb. Theory. B 18 1975, pp. 39-41
- 47- CLARK James H.
A fast scan-line algorithm for rendering parametric surfaces
ACM. April 1979
- 48- CLEVELAND M.R.
Hidden line elimination for 3D graphical objects
Master's thesis, Univ. of North Carolina
Chapel Hill USA Déc. 1973
- 49- COHEN D.
Incremental methods for computer graphics
PhD thesis ESD TR 69-193. Harvard USA 1969

- 50- COMBA P.G.
A procedure for detecting intersections of 3D objects
J. ACM Vol. n°3 - July 1968, pp. 354-367
- 51- COMBA P.G.
A langage for three-dimensional geometry
IBM systems journal, vol. 7, n°3 and 4 pp. 292-308 1968
- 52- CONTENSOU J.N.
Monographie CAO
CERCI
- 53- COURANT R. - ROBBINS H.
What is Mathematics?
Oxford 1941
- 54- COURTIEUX G. - CONTENSOU J.N.
CAO Promesses et difficultés
- 55- CROW F.C. - University of Texas, Austin Texas
Shadow algorithms for computer graphics
Computer graphics, vol. II, aug. 1977, pp. 242-248
- 56- COXETER H.M.S.
Regular polytopes
Dover Press USA, 1963
- 57- DEECKER G.F.P. - PENNY J.P.
Standart input forms for interactive computer graphics
Computer graphics Vol. 11 n°1, 1977, pp. 32-40
- 58- DESENS R.B.
Computer processing for display of three-dimensional structures
Technical Report
Naval postgraduate school, oct. 1969.
- 59- DIEDENHOVEN H. - FRITSHE B.
Géometric modelling with PROPEN 2
- 60- DOBKIN D. - LIPTON R.
Multidimensional Searching
SIAM J. Comput. 5, 2, June 1976, pp. 181-186
- 61- DUNCAN J.P. - VICKERS G.W.
Simplified method for interactive adjustment of surfaces
CAD; vol. 12-6, nov. 1980, pp. 305-308

- 62- EARNSHAW J.L. - YUILLE I.M.
A method of fitting parametric equations for curves and surfaces to sets of points defining them approximately
Computer aided design, vol. 3 n°2, pp. 19-22, winter 1971
- 63- EASTMAN C.
The concise structuring of geometric data for CAD
Data structures computer graphics and pattern recognition
Academic Press USA 1977
- 64- ENCARNACAO J.
A survey of new solutions for the hidden line problem
Symp. Interactive computer graphics
Delft Netherlands, oct. 1970, pp. 26-28
- 65- ENCARNACAD José - Technical University of Berlin
New programming and hidden-line detection techniques for the three-dimensional display of functions of two variables
Crane, Russack, New-York 1973, pp. 61-104
- 66- ENGELI M. - HRDLICZKA V.
Euklid eine einföhrung
- 67- FELL C.
Medusa 2D Mechanical drafting with interface To 3D modellers
Computer aided drafting and geometric modeling
(a training seminar) April 1980 London
- 68- FOLEY J.F. - WALLACA Y.L.
The art natural graphic man-machine conversation
Proceedings of the IEEE, vol.62 n°4 April 1974, pp. 462-471
- 69- FRANKE W. Herbert
Computer graphics computer art
Phaidon Press Limited 1971
- 70- FREEMAN H. - LOUTREL P.P.
An algorithm for the solution of the two-dimensional hidden-line problem
IEEE Trans. Elec. Comp. 1967
- 71- FURUKAWA S.A.
A solution to the hidden line problem
Rep. Faculty of Engineering
Yamanashi University Japan, vol. 22, 1971 pp.1-5

- 72- GALIMBERTI and MONTANARI
An algorithm for hidden line elimination
Communication of the ACM, vol. 12, n°4, avril 1969
- 73- GAUTIER J.Y.
CAO exemples d'utilisation (CADAM)
IBM, april 1980
- 74- GILOI Wolfgang K.
Interactive computer graphics
Berlin Prentice-Hall 1978
- 75- GOLDSTEIN A. - NAGEL R.
3D visual simulation
Simulation vol. 16 n°1, pp. 25-32 January 1971
- 76- GOURAUD H.
Computer display of curved surfaces
University of Utah UTEC-CSC-71-113 June 1971
IEEE transactions on computers C-20 p. 623 June 1971
- 77- GRAVE Michel
Manipulation de surfaces en conception assistée par ordinateur
IRIA-LABORIA
- 78- GRAYER A.R.
Alternative approaches in geometric modelling
CAD Vol. 12-4 juillet 1980, pp. 189-192
- 79- GRIFFITHS J.G.
Eliminating hidden edges in line drawings
Computer aided design, vol. II n°2 March 1979
- 80- GRIFFITHS J.G.
A data structure for the elimination of hidden surfaces
by patch subdivision
- 81- GRIFFITHS J.G.
A surface display algorithm
Computer aided design, vol. 10 n°1, january 1978

- 82- GRIFFITHS J.G.
Tape-Oriented hidden-line algorithm
CAD, vol. 13-1, janv. 1981, pp. 19-26
- 83- GROSSMAN
Procedural representation of three dimensional objects
IBM Research report, RC 5314, march 1975
- 84- GUEDJ R.A.
Le système "Grace" communication homme-machine
R.I.R.O. 1970, pp. 31-44
- 85- GUILLET J.P. (LCA)
Conception assistée par ordinateur
Compte-rendu d'activité 1972-1974
ADEPA Conseil scientifique
- 86- GUZMAN A.
Decomposition of a visual scene into three-dimensional bodies
AFIPS proceedings, fall joint computer conference, pp. 291-172
août 1977
- 87- HANRATTY Patrick J.
AD 2000
Product Brochure
- 88- HENRIKSEN THORE
3 D Geometry description AND F.E.M.
Experiences from use of interactive graphics aids for sesam-69
- 89- HILBERT D. - COHN-VOSSEN S.
Geometry and the imagination
Chelsea publishing company USA, 1952
- 90- HOSAKA M. - KIMURA F.
An interactive geometrical design system with handwriting input
information processing 77
B. Gilchrist, editor. IFIP, North-Holland 1977, pp.167-172
- 91- HYMAN Anthony
The computer in design
Sutdio Vista London 1973

- 92- JACKINS C. - TANIMOTO S.
Oct-trees and their use in representing three-dimensional objects
Computer graphics and image processing
Vol. 14, n°3, nov.. 1980
- 93- JOHNSON T.E.
Sketchpad III. A computer program for drawing in three dimensions
AFIPS conference
Proceedings 1963, vol. 23, IEEE, vol. 23 pp. 347-353
- 94- JONES C.B.
A New aproach to the 'hidden line' problem
The computer journal, june 1970
- 95- KIMURA Fumihiko - MOSAKA Mamdru
Program package ~~GEOMAP~~
References Manual, january 1977
- 96- KNUTH D.E.
The art of computer programming vol. 3
Sorting and Searching - Addison-Wesley 1973
- 97- KUNG H.T. - LUCCIO F. - PREPARATA F.P.
On fiding the maxima of a set of vectors
Journal of the association for computing machinery, vol.22 n°4
october 1975
- 98- KURT E. BRASSEL - SUNY Buffalo
An algorithm for shading of regions on vector display devices
Communications ACM, april 1979
- 99- KRAUSE F.L.
Methoden zur gestaltung von cad-system
Dr Ing. dissertaton, TU - Berlin 1976
- 100-KRAUSE F.L. - SPUR G. - MULLER G.
~~COMPAG~~
Technical University of Berlin Germany
- 101-LAFUE G.
Recognition of three dimensional objects from orthographics views
Computers graphics 10,2 Summer 1976

- 102- LAFUE Gilles
Theorem Prover from 2D to 3D
Proceedings of IFIP conference
Grenoble 1978, pp. 391-401
- 103- LANG C.A.
Romulus A geometric modeller for mechanical engineering
CAD/CAM systems
Computer aided drafting and geometric modeling
(a training seminar) London - April 1980
- 104- LAVIN Mark - LIEBERMAN Lawrence
A system for modeling three-dimensional objects
RC 5765 Computer science, dec. 1975.
- 105- LE TUAN Phac. - DAHAN P.L.
Approche théorique d'une technique :
perspectives et ombres calculées.
Thèse Télécom. Juin 1977
- 106- LEVIN Joshua
A parametric algorithm for drawings pictures of solid objects
composed of quadric surfaces
Communications ACM, oct. 1976, vol. 19 N°10
- 107- LEVIN - ZEV Joshua Boeing computer services
Quadril : a computer language for the description of quadric-
surface bodies
ACM 1980
- 108- LEWIS B.A. - ROBINSON J.S.
Triangulation of planar regions with applications
The computer journal vol. 21 n°4 nov. 1976
- 109- LIARDET et AL
Input to CAD systems : two practical exemples.
Proceedings of IFIP WG 5.2 Working conference
Grenoble 1978, pp. 103-128
- 110- LIARDET Michel
Picax-Polyhedron input to the computer using an axonometric
drawing

- 111- LIEBERMAN Lawrence - WIHLSON M. Joseph
Combining computer Synthesized Surfaces
Technical report - University of Pennsylvania, January 1974
- 112- LIEBERMAN Lawrence I. - LAVIN Mark A.
A system for modeling three-dimensional objects
IBM Research report - 17 décembre 1975
- 113- LIPTON R.J. - DOBKIN D.P.
The complexity of searching lines in the plane :
Preliminary version
Yale University New Haven, Connecticut
- 114- LLEWELYN A.I.
The impact of CAD and geometric modeling on industry
Computer aided drafting and geometric modeling
(a training seminar) April 80 - London
- 115- LOUTREL P.P.
Determination of hidden edges in polyhedral figures,
convex case
TR-400-145 Laboratory for electrosicence research N.Y
University USA - September 1966
- 116- LOUTREL Philippe P. - I.E.E.E.
A solution to the hidden-line problem for computer-drawn
polyhedra
Department of electrical engineering
New-York University - september 1967
- 117- LUCAS Michel
Contribution à l'étude des techniques de communication gra-
phique avec un ordinateur-éléments de base des logiciels
graphiques interactif
Thèse docteur-ès-sciences. Déc. 1977
- 118- LUCAS Michel - MORVAN Pierre
Images et ordinateur. Introduction à l'infographie interactive
Larousse Université 1976
- 119- LUH J.Y.S. - KROLAK R.J.
A mathematical model for mechanical part description
ACM fév. 1965, pp. 125-129

- 120- MAHL Robert
Visible surface algorithms for quadric patches
IEEE Transactions on computers. Vol. C-21 n°1 January 72
- 121- MANSUK LEE and ROBERT V. DIMARCO
Computer graphics with hidden surfaces-graphs
Computer graphics, vol. 3 pp. 85-91, déc. 1977.
- 122- MARTIN W.A.
Sorting
Computing Surveys. Vol. III n°4, déc. 1971
- 123- MARUYAMA K.
A procedure for detecting intersections and its applications
PB 201207 - May 1971
- 124- MARUYAMA K.
A study of visual shape perception
Technical report Department of computer science
University of Illinois 1972
- 125- MATSUSHITA Y.
A solution to the hidden line problem
Master's thesis Doc. 335
Department of computer science Univ. of Illinois USA. June 1969
- 126- Mc GRATH F.J.
A method for eliminating hidden lines with polyhedra
Simulation vol. 16 n°1 - January 1971 pp. 37-41
- 127- MOLES Abraham
Art et ordinateur
Synthèses contemporaines - Casterman
- 128- NEGROPONTE Nicholas
Computer aids to design and architecture
Nicholas NEGROPONTE 1975 - Petrocelli/Charter
- 129- NEWELL M.E. - NEWELL R.G. - SANCHA T.L.
A new approach to the shaded picture problem
Proceedings ACM - National conference 1972

- 130- NEWELL M.E. - NEWEL R.G. - SANCHA T.L.
A solution to the hidden surface problem
Proc. ACM 1972 - Nat. Conf. 1972 pp. 443-450
- 131- NEWMAN W.M.
A graphical technique for numerical input
Computer journal - Vol. II n°1 may 1968 pp. 63-64
- 132- NEWMAN William - SPROULL Robert
Principles of interactive
Computer graphics - seconde édition-
Mc GRAW-HILL 1979
- 133- NISHITA T. - NAKAMBE E.
An algorithm for half-toned représentation of 3D objects
Inf. process Japan Vol. 14 1974 pp. 93-99
- 134- NISHITA T. - NAKAMAE E.
An algorithm for hidden line elimination of polyhedra
Inf. process. Japan Vol. 12 - 1972 pp. 134-141
- 135- NITZAN David - AGIN Gerald
Fast methods for finding object outlines
Computer graphics and image processing 1974
- 136- OGDEN D.J. - STANDHAMMER J.
Computer graphics for half-tone 3D objects image
Proc. ACM conf. Interactive Graphics
Boulder Colorado USA - July 1974
- 137- OHLSON MBRK
System design considerations for graphics input devices
IEEE nov. 1978 pp. 9-18
- 138- OKINO NORIO - Hokkaido University
TIPS - 1
Technical information processing system for CAD/CAM
Institute of precision engineering
Horraido university, Sapporo Japan
- 139- PARKINS R.
3D graphics input using a storage tube display Internal Report
ABACUS - University of strathclyde 1977

- 140- PLAYNER J.S. - MANGIN J.C.
Approximation d'un relief naturel
Cahier du Gamsau n°10 - Janv. 1972
- 141- POTMESIL M. - FREEMAN H.
Implementation of two hidden-line algorithms
Computer graphics - Vol. 5 pp. 31-40 - May 1979
- 142- PRINCE M.D.
Interactive graphics for computer aided design
Addison-Wesley Publishing Co. 1971
- 143- RANDOLPH Franklin
A linear time exact hidden surface algorithm
ACM - April 1980
- 144- ROQUICHA A.G. - SAMUAL N. - VOELKER H.
Part and assembly description languages - Part Two
Production automation project
University of Rochester USA 1974
- 145- REQUICHA A.G.
Mathematical models of solids objects
University of Rochester
Collège of Engineering and applied service
Tech. Memo 28 - 1977
- 146- ROBERTS L.G.
Machine perception of 3D solids
TR 315 Lincon Laboratory MIT Cambridge Mass. USA - May 1963
- 147- ROMNEY G.W. - WATKINS G.S. - EVANS D.C.
Real time display of computer generated half-tone perspectives pictures
Boc. IFIP Cong. Booklet E 1968 pp. 72-78
- 148- ROMNEY G.W.
Computer assisted assembly and rendering of solids
TR-4-20 Department of computer science
University of Utah USA 1970

- 149- ROSS D.T.
Computer aided design, a statement of objectives
M.I.T. Project 8436 - Technical memorandum 4 - sept. 1960
- 150- SABIN M.A.
The use of piecewise forms for the numerical representation
of shape
Ph D. Diss. - Budapest ISBN 963 311 035 1
- 151- SCHUMACKER R.A. - BRAND B. - GILLILAND M. - SHARP W.
Study for applying computer generated images to visual
simulation
AFHRL-TR-69-14 - U.S. Air-Force Human Resources Laboratory
Sept. 1969
- 152- SEIFERT H. - BARGELE N. - FRITSCHÉ B.
Different ways to design three-dimensional representations
of engineering parts with PROPEN 2
- 153- SHAMOS M.I.
Problems in computational geometry
Springer verlag - June 1974
- 154- SHAMOS M.I.
Geometric complexity
7 ACM conference on theory of computing 1975 pp. 224-233
- 155- SHAMOS M.I. - HOEY D.
Geometric intersection problems
Symposium on foundations of computer science IEEE
octobre 1976 - pp. 208 - 215
- 156- SHAPIRA R. - FREEMAN H.
The cyclic order property of vertices as an aid in scene
analysis
ACM - June 1979 - vol. 22 n°6
- 157- SHU H. - OYAKE O.
Recent research in geometric modelling with primitive solids
Proc. CAM-1 Int. seminay - Atlanta USA - April 1976
- 158- SMADJA M.D.
Etude d'un système de construction de volumes
Thèse de troisième cycle - Novembre 1979

- 159- SMADJA M.D.
Modélisation topologique des pièces de la mécanique
MECO 79 - Mai 1979
- 160- SMADJA M.D.
Modélisation topologique des pièces de la mécanique
MECO 79 - Grenoble juin 1979
- 161- SPROULL R.F. - SUTHERLAND I.E.
A Clipping divider
AFIPS Conference proceedings - vol. 33 part I, 1968 pp.765-776
- 162- SPUR G. - KRAUSE F.L.
Gedmetric modelling in CAD-technology
Keynote address - Technical U. Berlin
- 163- SPAR G. - KRANSE F.L. - MAYR R. - MULER G. - SCHLIEP W.
A survey about geometric modelling systems
Berlin 1979
- 164- STOER J. - WITZGALL C.
Complexity and optimisation in finite dimensions
Springer-Verlag 1970
- 165- STRAUSS C.M. - POLEY S.
3 DPDP A three-dimensional piping design program
Proceedings of IFIP congress 1963
North Holland - Publishing Co - Amsterdam 1969
- 166- SUTHERLAND I.F. 74
Three-dimensional Data in put by tablet
Proceedings of the IEEE
Vol. 62 n°4 - April 1974 pp. 453-461
- 167- SUTHERLAND I.E.
Computer inputs and outputs
Scientifics american - Vol. 215 n°3 - sept. 1966, pp. 86-96
- 168- SUTHERLAND I.E. - SPROUL R.F. - SCHUMACKER R.A.
Sorting and the hidden surface problem
Proc. AFIPS 1973 - Nat. Comput. conf. pp. 685-693

- 169 - SUTHERLAND I.E. - SPROULL R.F. - SCHUMACKER R.A.
A characterization of ten hidden-surface algorithms
Evans & Sutherland computer corporation
Salt Lake city Utah
- 170- SUTHERLAND I.E.
Ten unsolved problems in computer graphics
Datamation 12, 5 - May 1966
- 171- SUTHERLAND I.E.
Sketchpad - A man-machine graphical communication system
AFIPS computer conference proceedings-spring joint computer
conf. 1963 - Vol. 23 pp.329-346
- 172- THERON Michel
Sur le programme Euclid : création, manipulation et visuali-
sation de formes tridimensionnelles dans un langage géomé-
trique à génération dynamique
Thèse, déc. 1972
- 173- THORNTON R.W.
Interactive modeling in three dimensions through two dimen-
sional windows
pp. 204-211 - CAD 78, Brighton IPC Press
- 174- TILOVE ROBERT B.
Set membership classification : A unified approach to geome-
tric intersection problems
IEEE 80
- 175- TIMMER H.G. - STERN J.M.
Computation of global - geometric properties of solid objects
C.A.D. - Vol. 12-6 nov. 1980 , pp. 301-304
- 176- VEENMAN P.
ROMULUS - The design of a geometric modeller
Cambridge University
- 177- VLIETSTKA J. - WIELINGA R.F.
Computer aided design
IFIP north-Holland/american - Elsevier 1972

- 178- VOELKER H. - REQUICHA A.
Geometric modelling of mechanical parts and processes
Computer science eng. - University of Rochester USA 1977
- 179- WALKER B. - Ferrati Cetec graphics limited
Use of A 3D geometric model in a turnkey CAD/CAM system
Computer aided drafting and geometric modeling
(a training seminar) - April 1980 London
- 180- WALLACE V.L.
Conversational ergonomics user-oriented design of interactive
graphics systems
Based on ACM/Sisgraph workshop - october 14-15 1976
Pittsburg USA
- 181- WARNOCK J.E.
A hidden surface algorithm for computer generated half-tone
pictures
TR RADC-TR-69-249 - Department of computer science
University of Utah USA - June 1969
- 182- WARNOCK J.E.
A hidden line algorithm for half-tone pictures representation
TR-4-15 Department of computer science
University of Utah USA - May 1968
- 183- WATKINS G.S.
A real time visible surface algorithm
AD-76-2004 June 1970
Department of computer science - Univ. of Utah USA
- 184- WEILER K. - ATHERTON P.
Hidden surface removal using polygon area sorting
Computer graphics - Vol. II pp. 214-222 - Aug. 1977
- 185- WEISS RUTH A.
Be vision; a package of IBM 7090 fortran programs to dran
orthographic views of combinations of plane and quadric
surfaces
Journal of the association for computing machinerie
Vol. 13 n°2 - April 1966

- 186- WESLEY M.A. - LOZANO-PEREZ T. - LIEBERMAN L.I. - LAVIN M.A.
A geometric modeling system for automated mechanical assembly
IBM Journal - Vol. 24 n°1 - January 1980
- 187- WHITTED TURNER
An improved illumination model for shaded display
Comm. ACM - Vol. 23 n°6 - June 1980
- 188- WILLIAMSON Hugh
Hidden-line plotting program
Communication of the ACM - February 1972 - Vol. 15 n°2
- 189- WILSON P.R.
Industries requirements for geometric modelling
Lucas Industries - England
- 190- WIRTH N.
Algorithms + Data structures = Programs
Prentice hall
- 191- WOLFE Robert - FITZGERALD W. GRACER F.
Interactive graphics for volume modeling
Research report - IBM Thomas J. Watson research center
3 juillet 1980
- 192- WOODSFORD P.A.
The design and implementation of the GINC 3D graphics
Software package - Software...practice and experience
Vol.1 pp. 335-365
- 193- WOON Peter - FREEMAN H.
A procedure for generating visible-line projections of
solids bounded by quadric surfaces
N.Y. University
- 194- WORDENWEBER B.
Build A geometric modeller with application
Computer aided drafting and geometric modeling (a training
seminar) - London - April 1980
- 195- WRIGHT T.
An algorithm for plotting iso-valued surfaces of a fonction
of three variables
ACM - April 1979

- 196- WRIGHT T.
Visible surface plotting program
Communication of the ACM - March 1974 - Vol. 17 n°3
- 197- WRIGHT T.
A two-space solution to the hidden line problem for plotting
functions of two variables
IEEE transactions computers - Vol. C-22 n°1 - January 73
- 198- WYLIE C. - ROMNEY G.W. - EVANS D.C. - ERDAHL A.
Halftone perspective drawings by computer
AFIPS conference proceedings - Vol. 31 1967 pp. 49
- 199- YAO Andrew - RIVEST Ronald
On the polyhedral decision problem
SIAM computer - Vol. 9 n°2 - May 1980
- 200- YAO Frances
On the priority approach to hidden surface algorithms
IEEE 1980
- 201- YI et AL
Three dimensional digitizing techniques for computer-aided
animation
CAD 1976 - IPC Press
- 202- YUTAKA Matsushita
Hidden lines elimination for a rotating object
Communication of the ACM - April 1972, vol. 15 n°4