

à revoir
62 ex
(deux pages)

7/89

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

«O»

وزارة التعليم العالي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

«O»

ECOLE NATIONALE POLYTECHNIQUE

Département : D'ELECTRONIQUE

«O»

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE - المكتبة
Ecole Nationale Polytechnique

Projet de Fin d'Etudes

S U J E T

CONTRIBUTION

A

L'ANIMATION D'IMAGES

Proposé par :

D. BERKANI

Etudié par :

A. HACENE

F. BENSALAM

Dirigé par :

D. BERKANI

Promotion : Juin 1989

E.N.P. : 10, Avenue Hacén Badi - EL-HARRACH - ALGER

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

«O»

وزارة التعليم العالي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

«O»

ECOLE NATIONALE POLYTECHNIQUE

Département : D'ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Projet de Fin d'Etudes

S U J E T

CONTRIBUTION

A

L'ANIMATION D'IMAGES

Proposé par :

D. BERKANI

Etudié par :

A. HACENE

F. BENSALAM

Dirigé par :

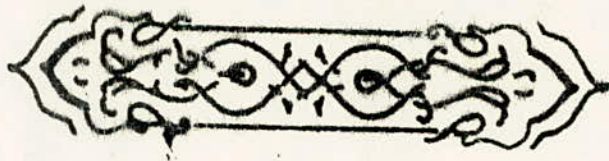
D. BERKANI

Promotion : Juin 1989

E.N.P. : 10, Avenue Hacen Badi - EL-HARRACH - ALGER

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَقُلْ رَبِّ زِدْنِي عِلْمًا



اللَّهُمَّ أَرْزُقْنَا حُسْنَ التَّوَكُّلِ عَلَيْكَ

اللَّهُمَّ إِنِّي أَسْأَلُكَ عِلْمًا نَافِعًا

إِنَّكَ سَمِيعُ الدُّعَاءِ.

R E M E R C I E M E N T



Nous remercions Mr D . BERKANI d'avoir propose le sujet
et qui a accepter de nous encadrer.

Nous tenons a remercier tous ceux qui ont participe de loin ou de
pres a l'elaboration de notre travail .

	page
INTRODUCTION	1
I- GENERALITES SUR L'ANIMATION	3
I.1- INTRODUCTION	3
I.2- GENERALITES	3
I.3- LES APPROCHES DE LA MODELISATION 3D	10
I.3.1- APPROCHE FIL DE FER	10
I.3.2- APPROCHE PAR FACETTE	11
I.3.3- APPROCHE PAR MODELISATION DES SURFACES GAUCHES	11
I.3.4- APPROCHE PAR MODELISATION SOLIDE	11
I.4- QUELQUES METHODES DE CONSTRUCTION D'OBJETS 3D	12
I.4.1- UTILISATION DES COURBES ET DES SURFACES	12
I.4.2- UTILISATION D'OPERATIONS BOOLEENNES	12
I.4.3- PAR EXTRACTION	13
I.5- OUTILS UTILISES POUR LA PRODUCTION ET LA MANIPULATION D'OBJET 3D ...	14
I.5.1- INTRODUCTION	14
I.5.2- PASSAGE DU SYSTEME UTILISATEUR AU SYSTEME OBSERVATEUR	14
I.5.3- LE DECOUPAGE	19
I.5.4- LA PROJECTION	20
I.5.5- ETUDE DE LA VISIBILITE	23
I.5.6- LES TRANSFORMATIONS GEOMETRIQUES	25
I.5.7- COMBINAISON DE TRANSFORMATIONS DE BASE	28
II- TECHNIQUES D'ANIMATION ET DE LA VISION ARTIFICIELLE	29
II.1- TECHNIQUES DE LA VISION ARTIFICIELLE	29
II.1.1- INTRODUCTION	29
II.1.2- CAPTEURS	31
II.1.3- SYSTEME D'ACQUISITION	35
II.1.4- TRAITEMENTS D'IMAGES	37
II.1.5- RECONNAISSANCE DES FORMES	52
II.2- NOTIONS ET PRINCIPES DE L'ANIMATION	57
II.2.1- DEFINITION	57
II.2.2- L'ANIMATION TRADITIONNELLE	57
II.2.3- L'ANIMATION PAR ORDINATEUR	58
II.2.4- TECHNIQUES D'ANIMATIONS	61
II.2.5- LES DIFFERENTS TYPES DE PRODUCTION D'ANIMATIONS	61
II.3- EDITEUR OBJETS 3D	63

	page
III- PREPARATION A L'ANIMATION ET MISE EN SCENE FINALE	67
III.1- INTRODUCTION	67
III.2- PRINCIPE DE CONSTRUCTION DE TRAJECTOIRE	67
III.3- LES OBJETS CLES	68
III.4- DETERMINATION DE LA VITESSE	70
III.5- VISUALISATION INDIVIDUELLE DU MOUVEMENT	71
III.6- STRUCTURE LOGIQUE D'UN FILM	75
III.7- CONSTITUTION DU SCENARIO	76
IV- RESULTATS ET INTERPRETATION	80
CONCLUSION	87
BIBLIOGRAPHIE	
ANNEXE 1	
ANNEXE 2	
ANNEXE 3	

.....

" La puissance d'evocation de l'image et les differents niveaux de perception dont elle peut faire l'objet en font incontestablement le moyen de communication le plus naturel et le plus efficace .

Alors que l'analyse d'un texte demande au lecteur des connaissances approfondies de vocabulaire et de regles de grammaire utilisees, la connaissance de l'image ne requiert aucune connaissance consciente , aucun apprentissage prealable . "

FRANCIS MARTINEZ

.....

المدرسة الوطنية المتكففة التقنفات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

INTRODUCTION GENERALE

Au cours de ces dix dernières années , les spécialistes du développement des logiciels de modélisation d'objets à 3D et de techniques d'animations ont considérablement amélioré le réalisme de l'image .

Les premières animations 3D se contentaient de mettre en œuvre des structures (FIL DE FER) ou des solides à facettes colorées (SIMULATION DE VOL) .

Grâce à la géométrie Euclidienne et au calcul matriciel , on savait alors modéliser des solides à l'aide de polygones , dans une base de données et visualiser ces objets en perspective sur un écran , avec élimination de parties cachées . Ces techniques offraient la possibilité d'animer des objets simples , comme exemple (GENÉRIQUE DE TÉLÉVISION) et ont été à l'origine des premières animations commerciales .

Néanmoins ces images avaient un aspect synthétique qui les rendait vite désagréable . Pour obtenir des images plus réalistes , la seule solution était de consacrer plus de temps au calcul de chaque image .

Ils découlent aussi du développement spectaculaire du matériel informatique dus notamment à la grande densité d'intégration atteinte et l'avènement des mémoires spécialisées devant stocker toutes les informations concernant l'objet à manipuler ainsi qu'un matériel adéquat qui forment une station graphique de modélisation et d'animation suivant :

- Une console de visualisation
- Des unites de dialogue : photostyle , table a digitaliser , clavier , etc...
- Processeur graphique

Le but de notre travail est de presenter des methodes et des outils utilises pour faire l'animation d'objets ; ainsi qu'un editeur qui permet de construire ces objets .

Le premier chapitre est consacre a des generalites sur l'animation d'image , ou on a presenter le materiels et les outils de graphisme 3D utilises dans les logiciels pour l'animation des objets 3D .

Le second chapitre portera sur la presentation et la description des differentes techniques d'animations et de vision artificielle , ainsi qu'a la presentation de l'editeur d'objets .

Dans le troisieme chapitre , nous avons defini , comment preparer l'animation et la mise en scene final .

Le dernier chapitre comporte deux parties , le premier traite l'aspect logiciel des differents algorithmes pour les mouvements de notre objet (ROTATION , TRANSLATION ,etc ...) ; et la seconde partie est consacree a la presentation logiciel de l'editeur d'objets ,ainsi que les resultats obtenus des programmes etudie.

Une bibliographie comportant la liste des theses et ouvrages afin de permettre aux interesses de retrouver les sources des techniques presenter .

CHAPITRE I

GENERALITES SUR
L'ANIMATION

I.1- INTRODUCTION GENERALE :

Vu le caractere universel de l'image et sa presence dans presque tous les domaines, son animation ne fera que renforcer d'avantage son champ d'interet .

Notons que la realisation d'un dessin anime revient a creer un grand nombre de dessins, qui, pris dans un ordre chronologique , different legerement l'un de l'autre . L'elaboration manuelle' de ces desssins est une tache longue, fastidieuse, voire meme delicate .

L'apparition des consoles de visualisation et l'evolution des techniques

informatiques graphiques constituent un moyen efficace pour repondre a ce besoin dont la demande est sans cesse croissante .

Apres avoir pris connaissance des principes cinemato-graphiques fondamentaux et de la complexite de realisation traditionnelle de film de dessins animés .

Partant de cette situation et grace a un procede d'interpolation la possibilite nous est offerte d'associer un mouvement quelconque a n'importe quel objet de la scene . Afin d'aboutir au film final , il y'a lieu de synchroniser les differents participants de la scene et eventuellement les scenes entre elles .

I.2 - GENERALITES :

INTRODUCTION A L'INFOGRAPHIE :

Qu'est ce que l'infographie ?

L'infographie , abreviation d'informatique graphique , est l'ensemble des techniques de representation graphiques automatique d'un lot d'informations .

Elle a vu le jour au cours des annees cinquante , presque au meme temps

que les premiers ordinateurs , et son utilisation par des physiciens a l'epoque , se limitait a une transformation de tableaux de chiffres en graphes .

Vers 1963 , IVAN.E.SUTHERLAND a concu le systeme SKETCHPAD ancetre des systemes actuels de synthese d'images . Et ce n'est qu'a partir des annees 70 , avec l'avenement des consoles conversationnelles , que des systemes de synthese d'image ont vu le jour et ont commence a se generaliser .

Pourquoi synthese d'images ?

Son interet reside dans la traduction et la representation des formulations mathematiques , physiques ou autres ... sous forme d'images .

Nous citons a titre d'exemples la representation des phenomenes micro ou macroscopiques (molecules , phenomenes astronomiques etc ...) .

Elle nous permet donc de comprendre la realite dans son aspect , sa representation et son fonctionnement .

Son domaine d'application est tres riche et concerne entre autres :

- L'aeronautique : Simulateurs de vol , ...
- L'industrie electronique : Circuits integres , ...
- L'architecture : Plan d'ensembles immobiliers , ...
- L'education : Les dessins animes , les jeux video ,
l'enseignement assiste , ...
- L'industrie automobile .
- etc ...

A-STRUCTURE D'UN SYSTEME GRAPHIQUE :

Il comprend deux parties :

- Le materiel
- Le logiciel

LE MATEREIL GRAPHIQUE :

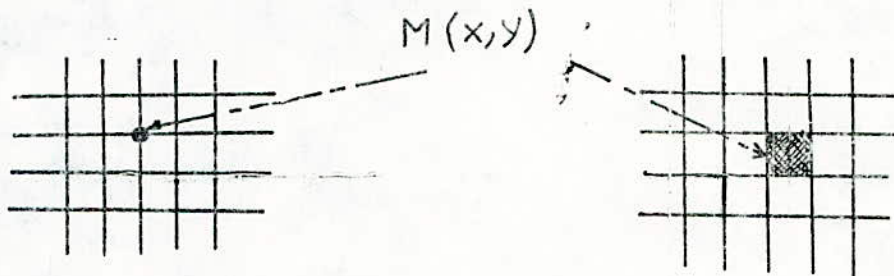
Si le graphisme par ordinateur est devenu une industrie prospere , la recherche dans le domaine du materiel graphique n'a pas cesse depuis . La liste des equipements que nous presentons est loin d'etre exhaustive .

a - L'ecran de visualisation :

C'est un dispositif de sortie pour la visualisation des images , les surfaces de visualisation sont des surfaces constituees d'un quadrillage regulier definissant un systeme de coordonnees . On distingue deux types de surfaces :

- Tout sommet du quadrillage represente un point $M(X, Y)$
- Tout carre elementaire represente un point $M(X, Y)$.

(Voir figure 1) .



Il existe des ecrans a rayons cathodiques , a cristaux liquides , a plasma , etc ...

Notons que la couleur ne peut etre obtenue qu'avec un tube a rayons cathodiques .

Son principe de fonctionnement est base sur des techniques de balayage (balayage recurrent , cavalier , ...) .

b - La memoire d'entretien :

C'est le dispositif qui permet de maintenir l'affichage d'une image sur l'ecran . Dans le cas du balayage cavalier , la memoire d'entretien contient la liste des coordonnees des extremités des segments constituant le dessin . Tandis que pour le cas du balayage recurrent , la

memoire contient la definition point par point de l'image et est organisee en tableau bidimensionnel .

c - Processeur graphique :

C'est un calculateur specialise qui controle les dispositifs d'affichage sur l'ecran . Il joue le role d'interface entre l'unite de calcul et le terminal graphique . Il est charge des taches suivantes :

- * Gestion des transferts
- * Decodage des ordres graphiques
- * Gestion de la memoire de rafraichissement
- * Mise en oeuvre des generateurs de fonctions graphiques , etc...

d - Dispositif de communication :

Ce sont les organes permettant la communication entre le systeme et l'utilisateur . Les dispositifs les plus utilises actuellement sont :

- Le clavier alphanumerique et le clavier de fonction :

Sert a entrer du texte , des commandes ou donnees numeriques ou encore l'execution des fonctions predefiniees .

- Photostyle et Reticule :

Permettent une designation directe sur l'ecran leur position sur l'ecran designe les coordonnees de la fonction a executer .

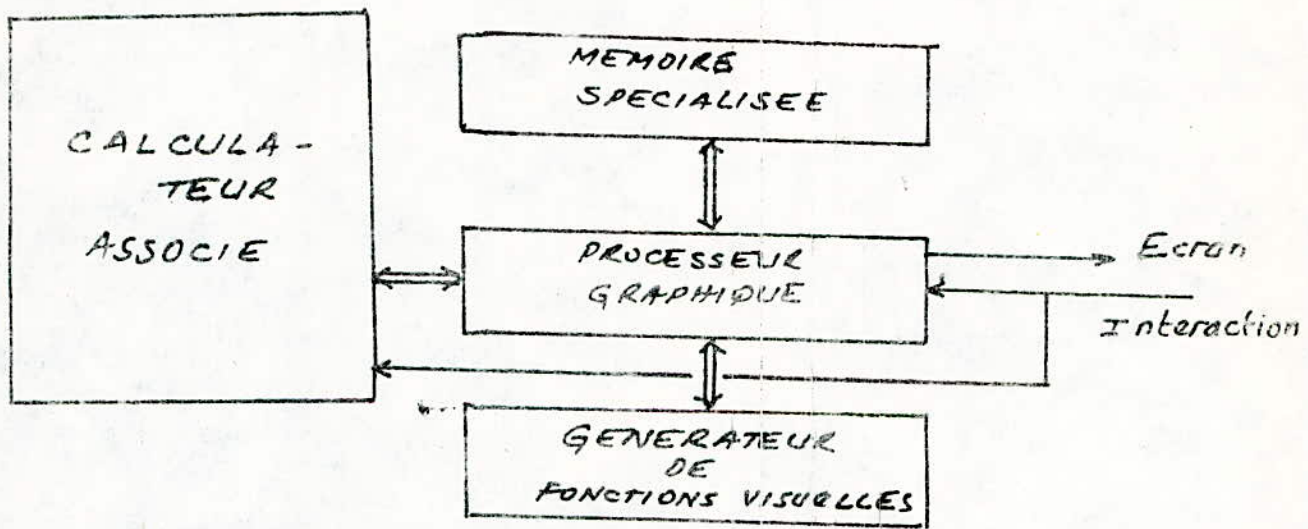
- Tablette graphique , boule roulante et souris :

Permettent une designation indirecte sur l'ecran un symbole lumineux s'affiche sur l'ecran pour designer la position indiquee .

- Ecran sensitif (Tactile) :

la designation se fait a l'aide du doigt sur l'ecran .

- Camera , etc ...



STRUCTURE D'UN SYSTEME GRAPHIQUE

LE LOGICIEL :

On retrouve deux types de Logiciels :

- Le Logiciel de base : C'est un ensemble de sous-programmes généralement stocké dans une bibliothèque particulière permettant l'utilisation facile d'un système graphique.

- Le Logiciel d'application : Il concerne un type d'application particulière et utilise les modules du logiciel de base.

B - OUTILS DE GRAPHISME 3-D

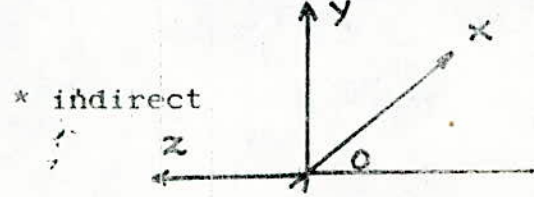
B.1 - INTRODUCTION :

Un objet 3D peut être modélisé et représenté sur un dispositif graphique qui n'a que deux dimensions (écran graphique, traceur de courbes etc ...). Il est donc possible de projeter des images en 3D sur un plan de vue, au moyen de certains outils mathématiques et cela sans perte d'informations.

B.2 - GENERALITES :

a - Les systèmes de coordonnées :

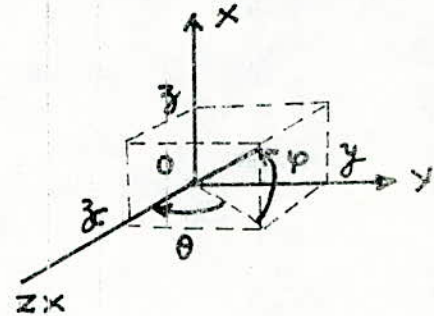
Pour représenter un point de l'espace 3D, on a le choix entre deux systèmes de coordonnées : le système direct ou indirect, ces deux systèmes ont les directions de l'axe Z diamétralement opposées.



Soit P un point de l'espace 3D, ces coordonnées peuvent s'écrire sous la forme suivante :

$$\begin{aligned} X &= R \cos \theta \cdot \cos \varphi \\ Y &= R \sin \theta \cdot \cos \varphi \\ Z &= R \sin \varphi \end{aligned}$$

* Passage d'un système à un autre :



B.3 - MODELISATION GEOMETRIQUE :

Un objet est défini par un ensemble d'informations de natures différentes : géométrique, topologique, aspect, etc ...

Chaque donnée a sa propre signification et son importance. Les données géométriques concernent la représentation des formes et des dimensions de l'objet (coordonnées, équations, etc ...), les données topologiques concernent les relations qui existent entre les différents objets de la scène ou bien entre les différents éléments géométriques constituant l'objet. On trouve aussi d'autres informations concernant les

relations qui existent entre les différents éléments géométriques constituant l'objet . On trouve aussi d'autres informations concernant la couleur , le degré de transparence , etc ...

L'utilité de la modélisation est de permettre un traitement automatique en représentant un objet à partir de ces propriétés géométriques . La modélisation est utilisée pour représenter des entités physiques (objets solides) et/ou abstraites , afin de produire non seulement des dessins , mais en général pour représenter leurs structures et leur comportement .

Un modèle consiste en une structure de données de l'application à laquelle s'ajoute une collection de procédures propres au programme d'application , permettant la définition de sa structure et de son comportement .

I.3- LES APPROCHES DE LA MODELISATION 3D :

La modelisation 3D permet de manipuler et de représenter un objet ou une collection d'objets dans l'espace 3d .

Un modèle 3D manipule trois niveaux d'éléments principaux

- Niveau 0 : Les éléments 2D (points , arcs , courbes)
- Niveau 1 : Ce sont les surfaces (plans , surfaces de révolution , surface gauches , etc ...) .
- Niveau 2 : ce sont les volumes , nous distinguons : Les cylindres , cones , prismes , polyèdres , etc...

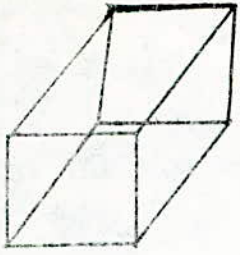
On distingue quatre approches de la modélisation géométriques tridimensionnelle . Le choix du type du modèle est dicté par le domaine d'application .

I.3.1 - APPROCHE FIL DE FER :

C'est la méthode la plus ancienne utilisée pour la construction graphique . elle a été développée par BRAID vers 1970 , à l'université de COMBRIDGE .

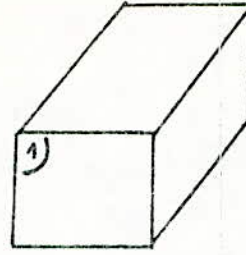
Dans ce type de développement l'objet est connu par son "squelette" , donc seule la connaissance des sommets et des arêtes les reliant suffit pour les décrire .

La qualité de la représentation en fil de fer est en fonction de la complexité de la scène . Ainsi , sa compréhension devient difficile dans le cas où le nombre d'objets d'une même scène devient assez important .

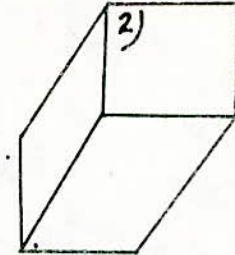


Representation

fil de fer



(1)



(2)

Deux interpretations

possibles

1.3.2 APPROCHE PAR FACETTE :

On définit l'objet grâce à des règles mathématiques d'approximation des surfaces par des facettes polygonales. Cette technique peut être combinée à la précédente pour atténuer l'impact de l'ambiguïté.

1.3.3 - APPROCHE PAR MODELISATION DES SURFACES GAUCHES :

Ce modèle consiste à approximer par interpolation des surfaces. Celle-ci est faite par des fonctions mathématiques (BEZIER, COONS, SPLINE, ...). On approche alors les surfaces par des morceaux de courbes par des courbes plus ou moins complexes cette approche évolue vers une modélisation solide utilisant les ombres, les couleurs ... Elle est utilisée en construction mécanique, aéronautique, ...

1.3.4 - APPROCHE PAR MODELISATION SOLIDE :

Celle-ci permet la représentation d'objets complexes en assurant la cohérence de l'information. Elle tient compte en particulier de la matière qui constitue l'objet. Un objet dans ce cas est connu par un intérieur et un extérieur et est défini par un volume borné par des surfaces planes ou gauches. La représentation n'est donc pas ambiguë.

Cette méthode est apparue au niveau industriel vers 1979 et a été utilisée dans des logiciels connus tels que : GMSOLIDE, EUCLID, SYNTAVISION,

On distingue trois types de representations solides :

- Par contour (boundary representation)
- Par volume de base CGS (CONSTRUCTIVE SOLID GEOMETRY)
- Par balayage (Sweeping)

I.4 - QUELQUES METHODES DE CONSTRUCTION D'OBJETS 3D :

Il existe plusieurs methodes possibles pour creer , en voici les principales :

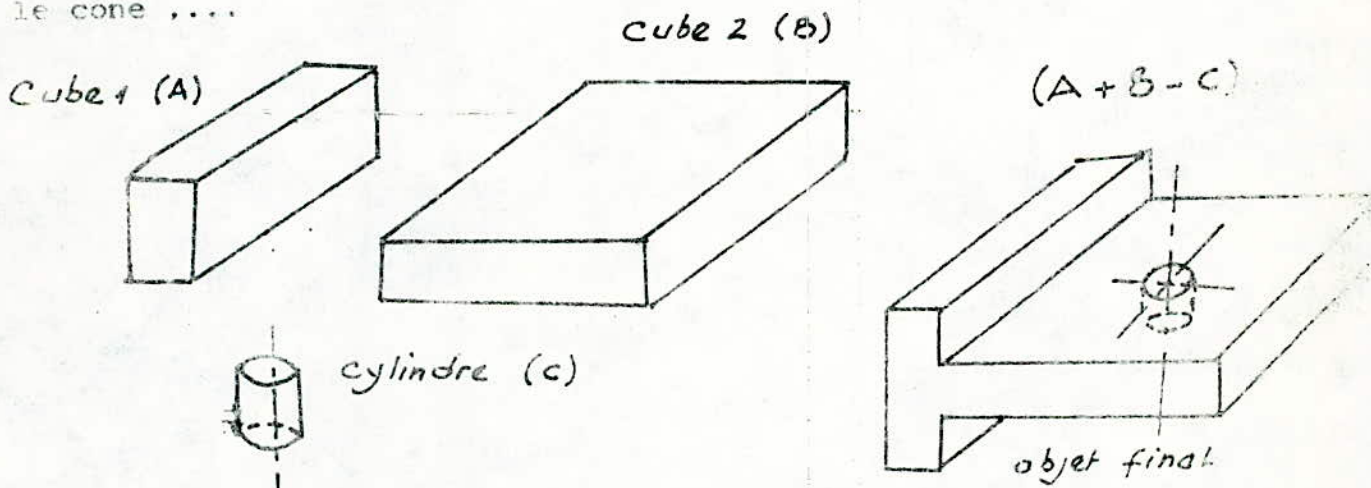
I.4.1 - UTILISATION DES COURBES OU DE SURFACES :

Celles-ci sont obtenues par plusieurs techniques . On trouve :

- Le lissage de points
- La courbe (ou surface) parallele a une courbe (ou surface)
- La concatenation de la courbe (ou surface) initiale
- La deformation de la courbe (ou surface) .

I.4.2 - UTILISATION D'OPERATIONS BOOLEENNES :

Cette methode consiste en la generation d'objets complexes a partir d'operations booleennes sur des objets plus simples . Ces operations sont : l'union , l'intersection et la soustraction . Parmi ces objets de bases on trouve : le cube , le cylindre , la sphere , le cone ,



DESCRIPTION D'UN OBJET A L'AIDE D'OPERATIONS BOOLEENNES

I.4.3 - PAR EXTRACTION :

Soit un contour dans un plan : la translation de ce contour cree un objet " Par epaisseur " . Si le contour comporte des trous . sa translation cree une " epaisseur trouee " . sa rotation autour d'un axe engendre un volume de revolution decrit dans un plan a l'aide de fonctions en 2D .

I.4.4 - CONCLUSION :

C'est le modele " fil de fer " qui est le mieux a mettre en oeuvre car il permet d'obtenir une representation rapide sur l'ecran (qui est l'aspect le plus important a la mise en oeuvre de l'animation) .

1.3 - OUTILS UTILISES POUR LA PRODUCTION ET LA MANIPULATION D'OBJETS 3D

1.3.1 - INTRODUCTION :

Dans cette partie nous essayons de presenter une chaine de production et de manipulation d'objets 3D . Evidement , cette phase qui prepare la visualisation de ces objets et completement abstraite pour le lecteur et resoud le plus grand probleme de l'infographie tridimensionnelle : le passage d'un objet a trois dimensions a une image en deux dimensions .

Tous les traitements (c'est a dire : le clipping , projection perspective , passage du systeme utilisateur au systeme observateur , etc ...)

dependront des deux parametres suivants :

- Position de l'observateur : $O (X_0 , Y_0 , Z_0)$
- Direction des l'observateur (Ligne de visee) , on definit

un point de cette ligne : $D (X_d , Y_d , Z_d)$

1.5 - OUTILS UTILISES POUR LA PRODUCTION ET LA MANIPULATION D'OBJETS 3D :

1.5.1- INTRODUCTION :

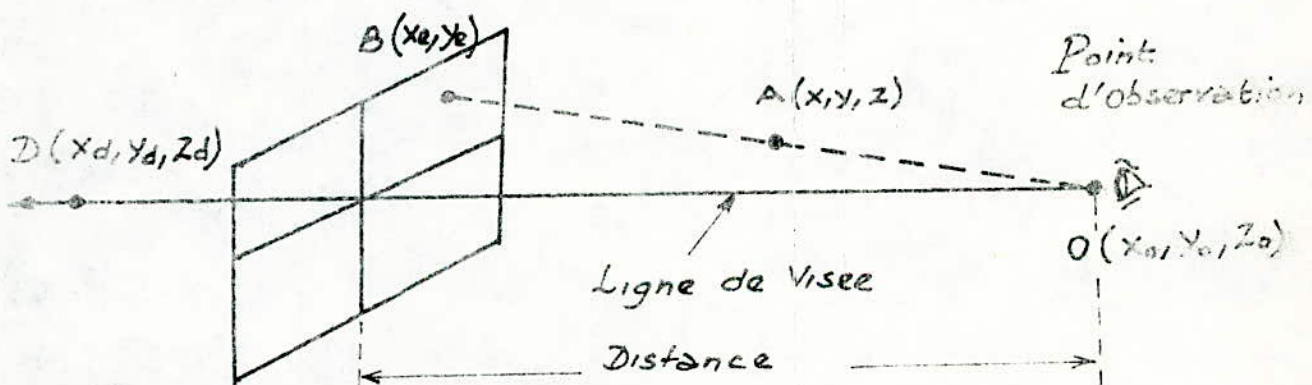
Dans cette partie nous essayons de presenter une chaine de production et de manipulation d'objets tridimensionnels .

Evidemment , cette phase qui prepare la visualisation de ces objets est complètement abstraite pour le lecteur et resoud le plus grand probleme

de l'infographie tridimensionnelle : le passage d'un objet a trois dimensions a une image en deux dimensions .

Tous les traitements qui vont suivre (c'est-a-dire : Clipping, Projection perspective, passage du systeme utilisateur au systeme observateur, etc) dependront des deux parametres suivants :

- Position de l'observateur : $O (X_o , Y_o , Z_o)$
- Dimension de l'observateur (ligne de visee) on definira un point de cette ligne : $D (X_d , Y_d , Z_d)$



A : Point d'un objet (en 3D)

B : Point image de A (en 2D)

1.5.2 - PASSAGE DU SYSTEME UTILISATEUR AU SYSTEME OBSERVATEUR :

Pour voir les objets sous n'importe quel angles , nous avons le choix entre deux possibilites :

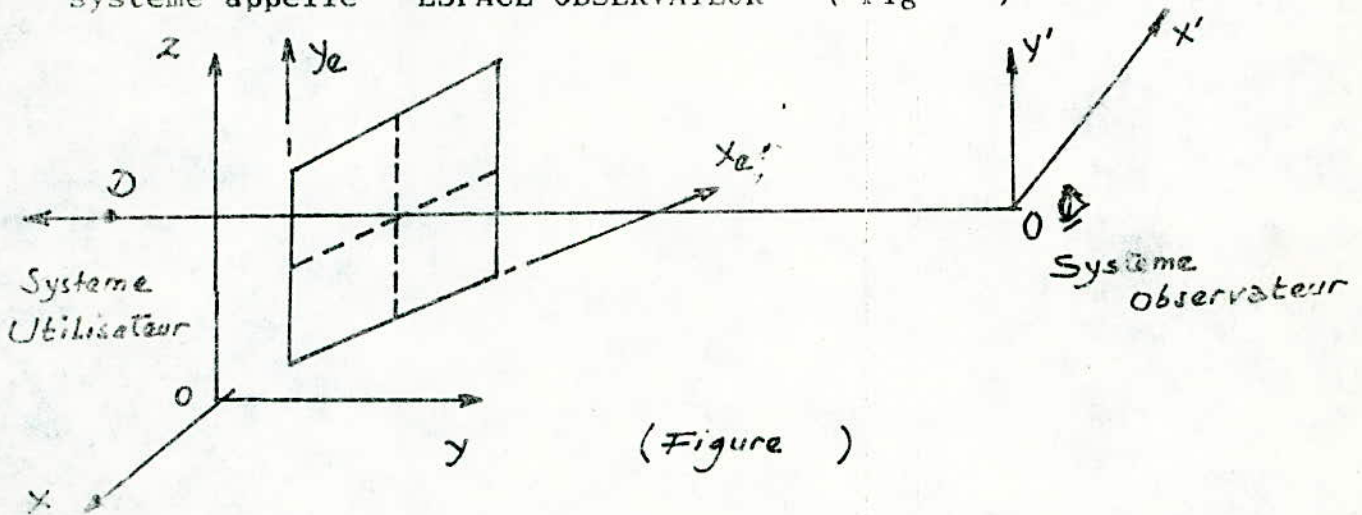
- Le point de vue reste fixe et l'objet subit toutes les transformations

souhaitees ,

- L'objet reste fixe et le point d'observation est positionnee correctement

ment
Le choix a ete porte sur la seconde qui est la plus proche de la realite . La mise en oeuvre de cette solution necessite la procedure suivante :

L'objet etant defini dans un systeme de coordonnees tridimensionnelles, connu sous le nom de systeme d'objets ou systeme utilisateur . On peut generer pour un meme objet plusieurs vues dependant de l'endroit a partir duquel on l'observe . Ceci revient a l'amener dans un autre systeme appelle " ESPACE OBSERVATEUR " (Fig)



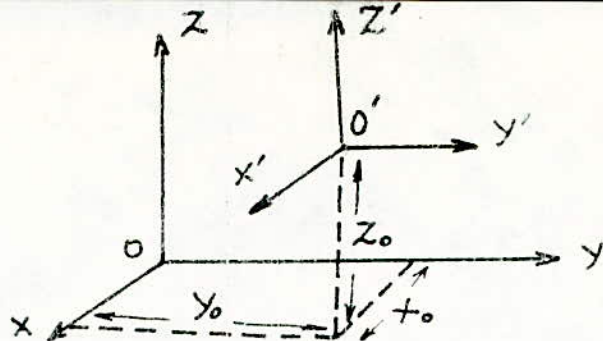
- Les differentes transformations fondamentales , que le systeme utilisateur doit subir pour coincider avec le systeme de l'observateur .

ETAPE 1 : Changement d'origine au point d'observation .

$$T1 \quad \begin{cases} X' = X - X_0 \\ Y' = Y - Y_0 \\ Z' = Z - Z_0 \end{cases}$$

ETAPE 2 : Changement d'orientation du systeme obtenu

$$T2 \quad \begin{cases} X' = -X \\ Y' = Z \\ Z' = -Y \end{cases}$$



ETAPE 3 : Rotation autour de l'axe Y de façon à amener l'axe Z dans le plan contenant l'axe Y' et le point D .

Soit t l'angle de rotation tel que :

$$\cos t = (Y_o - Y_d) / \sqrt{(X_d - X_o)^2 + (Y_d - Y_o)^2}$$

$$\sin t = (X_o - X_d) / \sqrt{(X_d - X_o)^2 + (Y_d - Y_o)^2}$$

On obtient :

$$T3 \quad \begin{cases} X' = X * \cos t - Z * \sin t \\ Y' = Y \\ Z' = X * \sin t + Z * \cos t \end{cases}$$

ETAPE 4 : Cette dernière étape permet d'effectuer une rotation autour de l'axe OX' , de telle façon que OZ' soit sur la ligne de visée .

Sans trop insister sur les démonstrations nous obtenons :

$$T4 \quad \begin{cases} X' = X \\ Y' = Y * \cos V + Z * \sin V \\ Z' = Y * \sin V + Z * \cos V \end{cases}$$

En conclusion , nous pouvons dire que le passage du système utilitaire au système observateur , on effectue les transformations suivantes :

$$(X' , Y' , Z' , 1) = (X , Y , Z , 1) * V$$

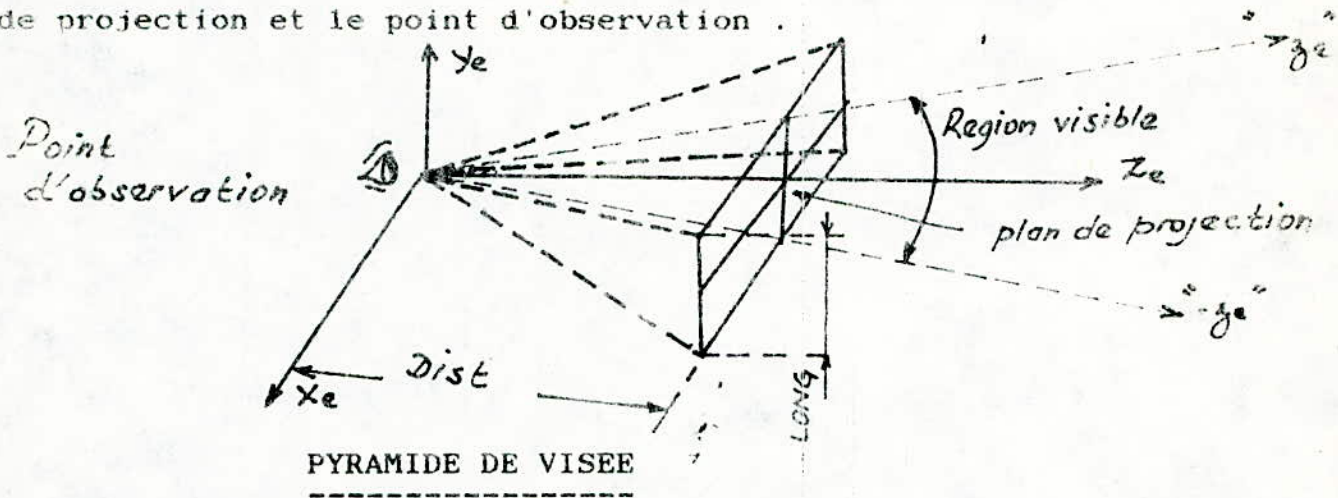
Avec V , la matrice carrée résultant des quatre transformations vues précédemment . Donc :

$$V = T1 * T2 * T3 * T4$$

I.5.3 - LE DECOUPAGE (CLIPPING) :

Le clipping est une operation qui consiste essentiellement a eliminer tous les objets ; ou parties d'objets, situes hors du champ de vision de l'observateur .

Ce champ de vision n'est autre que la pyramide de visee definie par une base rectangulaire et un sommet , qui sont respectivement le plan de projection et le point d'observation .



Les conditions requises pour qu'un point appartienne au plan de projection (c'est-a-dire visible) sont :

$$-Z_e \leq \frac{Dist}{LNG/2} \times X_e \leq Z_e$$

$$-Z_e \leq \frac{Dist}{LRG/2} \times Y_e \leq Z_e$$

Avec LNG et LRG etant respectivement la longueur et la largeur du plan de projection .

Posons : $LNG/2 = a$ et $LRG/2 = b$

On obtient les conditions sous la forme matricielle suivante :

$$(X_c , Y_c , Z_c , 1) = (X_e , Y_e , Z_e , 1) \times N$$

$$\text{Avec : } N = \begin{bmatrix} \text{Dist}/b & 0 & 0 & 0 \\ 0 & \text{Dist}/a & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Il existe plusieurs techniques de découpage .Le choix de l'une d'elles se fait généralement en fonction des entites manipulées et des sorties souhaitées .

On trouve deux grandes familles d'algorithmes de découpage :

- a - Ceux dont le découpage se fait par rapport au plan de vue , donc apres projection . c'est le découpage bidimensionnel .
- b - Ceux dont le découpage se fait dans l'espace tridimensionnel et s'effectue par rapport aux quatres plans de la pyramide de vision .

REMARQUE :

 Parmi les algorithmes les plus utilises nous citerons celui de SUTHERLAND-HOGMANN (voir annexe 2) .

celui-ci traite directement des polygones et permet d'avoir un temps de traitement acceptable .

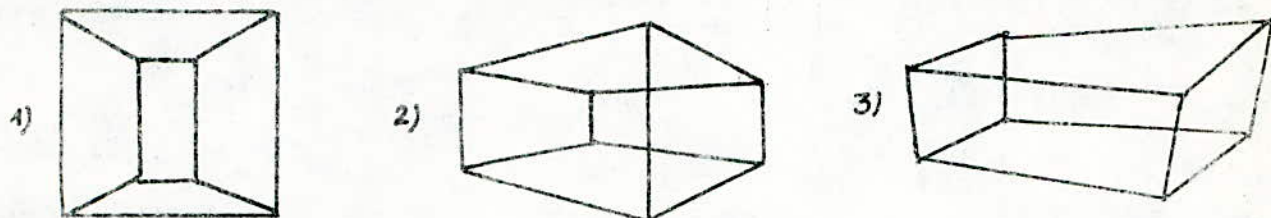
1.5.4 - PROJECTION :

 En general , les projections transforment les points d'un systeme de coordonnees de dimension N en des points de dimension inferieur a N .

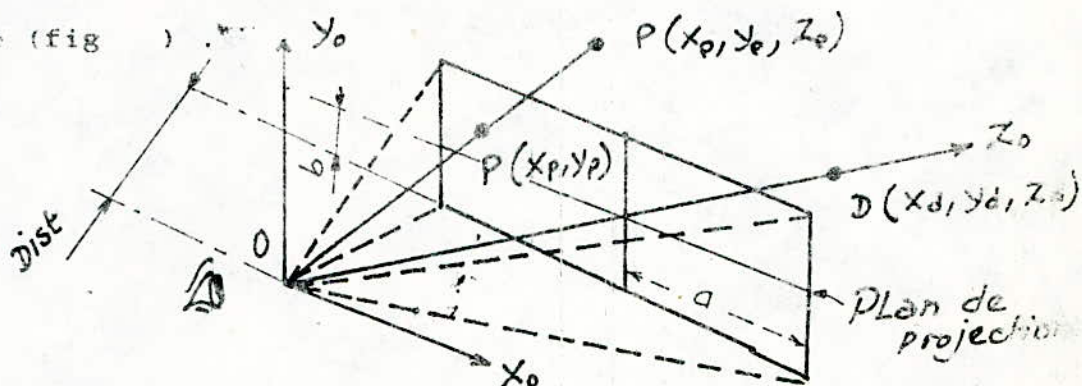
1.5.4.1 - PROJECTION PERSPECTIVE :

 Dans le cas de la projection , le centre de projection est la distance finie du plan de vue , la projection perspective cree un effet visuel similaire au systeme photographique et notre systeme de vision des objets , on dira qu'on a une perspective a :

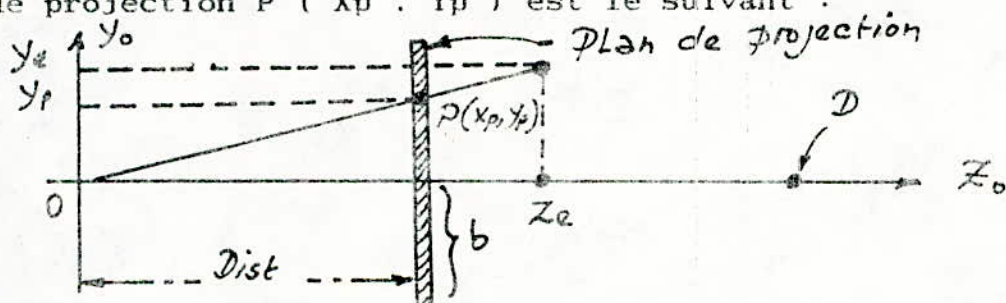
- Un point de fuite : Lorsque le plan de vue coupe un des axes .
- Deux points de fuite : Lorsque le plan de vue coupe deux axes .
- Trois points de fuite : Lorsque le plan de vue coupe les trois axes .



* Cette methode (projection perspective) permet de conserver l'illusion de la troisieme dimension et presente un degre de realisme appreciable (fig)



Le procede de passage du point de l'espace 3D $P(X_e, Y_e, Z_e)$ au point du plan de projection $P'(X_p, Y_p)$ est le suivant :



D'apres les proprietes des triangles semblables nous pouvons ecrire :

$$Y_p = (\text{Dist} / b) \times (Y_e / Z_e)$$

De meme sur le plan $X_0, 0, Y_0$ on trouve :

$$X_p = (\text{Dist} / a) \times (X_e / Z_e)$$

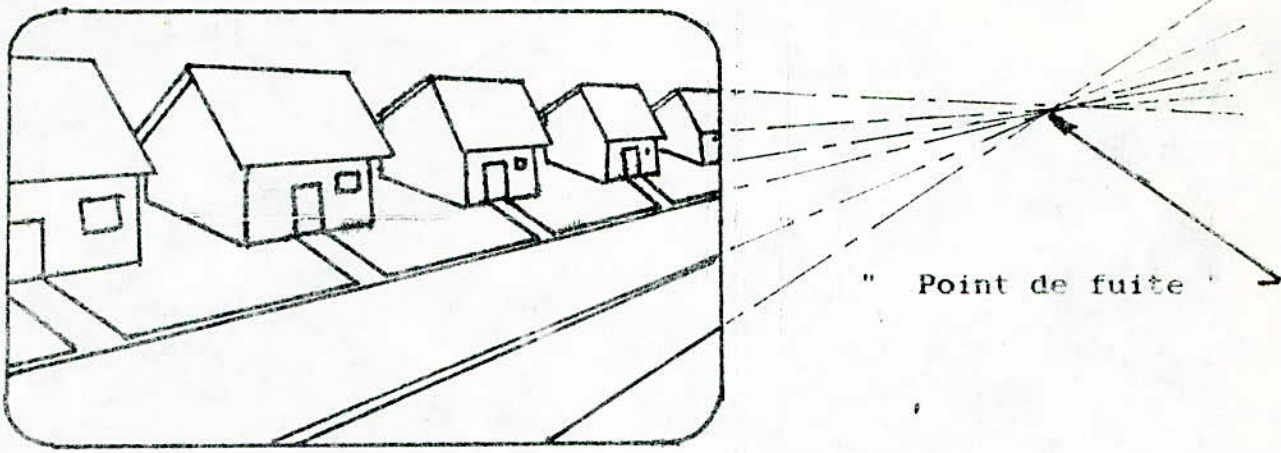
Avec : b = demi-largeur de plan de projection

a = demi-longueur du plan de projection

Dist = Distance entre le point observateur 0 et le plan de

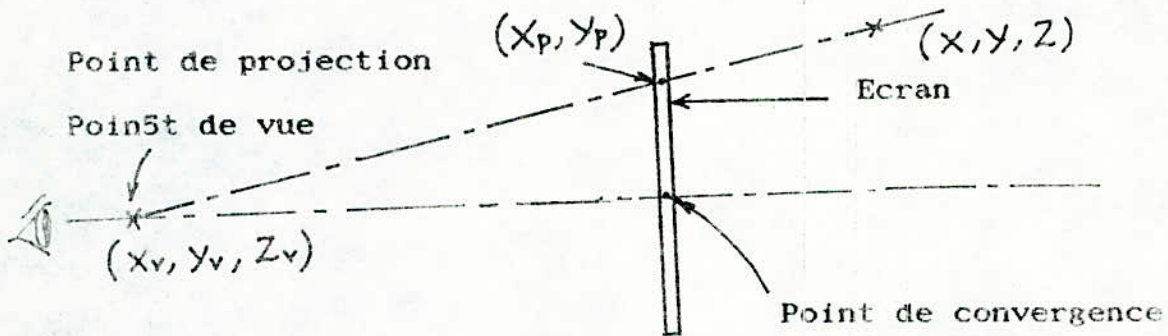
projection .

EXEMPLE DE PROJECTION PERSPECTIVE A UN SEUL POINT DE FUIITE :



Dans une vue en perspective , les droites paralleles convergent vers un point de fuite pour que les objets les plus eloignes soient plus petits que les objets les plus proches . Ceci est un moyen de parachever le realisme et la profondeur des images .

Pour tout point d'une vue en 3D de coordonnees (X,Y,Z) la nouvelle position sur l'ecran (Xp , Yp) , qui tient compte de la perspective et est calculee de la maniere suivante :



$$X_p = X_v + (X_v - X) * Z_v / (Z - Z_v)$$

$$Y_p = Y_v + (Y_v - Y) * Z_v / (Z - Z_v)$$

La distance d'un point (X, Y , Z) est calculee de la maniere suivante :

$$D = \text{SQR} ((X - X_v) ^ 2 + (Y - Y_v) ^ 2 + (Z - Z_v) ^ 2)$$

1.5.5 - ETUDE DE LA VISIBILITE : (Elimination des surfaces cachees)

L'etude de visibilite constitue l'un des grands problemes qui se sont poses dans le domaine de la synthese d'images , si l'on se contente de la representation graphique obtenue par la projection de tous les points des objets a visualiser , des problemes d'interpretation peuvent surgir a la visualisation.

Aussi l'interpretation d'une representation en fil de fer est d'autant plus complexe que le nombre d'elements composants la scene est important . De nombreux travaux ont ete effectues dans le domaine de l'etude de la visibilite,et chaque annee de nouvelles solutions sont proposees

Les algorithmes traitant de ce probleme peuvent etre repartis selon divers criteres I. SOUTHERLAND et AL ont propose une classification base essentiellement sur trois familles d'algorithmes . nous nous baserons sur ceux de la famille operant dans l'espace objet car il font

avec une tres grandes precision (en general celle de la machine) les calculs permettant de distinguer les parties visibles des parties cachees . parmi les principaux algorithmes de cette categorie , on trouve ceux de : APPEL , LOUTREL et GALIMBERT-MONTANARI

Avantages :

- Grande précision de la solution calculee .
- Resolution dans l'espace utilisateur sans tenir compte des parametres de visualisation .

Inconvenients :

- S'applique aux polyèdres uniquement
- Demandes une orientation des faces .
- Temps de calcul assez long car il est fonction du nombres d'arêtes de la scène .

CONCLUSION :

Dans l'aspect animation, le facteur temps etant plus important que le realisme de l'image , pour resoudre le compromis temps - realisme l'algorithme de GALIMBERT et MONTANARI qui consomme moins de temps de traitement , qui est facile a mettre en oeuvre .

La presentation de cet algorithme est donnee en annexe 2 .

1.5.6 - LES TRANSFORMATIONS GEOMETRIQUES :

Pour les transformations geometriques , il existe un type de coordonnees auquel travail les infographistes , qu'on appelle coordonnees homogenes . il faut en effet , savoir qu'un objet dans l'espace a N dimensions peut parfaitement etre represente dans un espace a N+1 dimensions . c'est un peu l'inverse de ce qui se passe avec la projection ou l'on a N-1 dimensions . seulement ici la dimension supplementaire n'interviendra que pour jouer le role de facteur d'echelle .

C'est ainsi qu'un vecteur tridimensionnel (X , Y , Z) sera represente par son vecteur homogene (SX , SY , SZ , S) dans lequel S donnera la valeur du facteur d'echelle .

L'utilisation des matrices de transformations est un avantages du point de vue temps et simplicite de calcul ;.

En general la description du modele ne permet pas d'obtenir une vue facilement comprehensible de la scene 3D .

A cet effet , des operations geometriques de base sont mises en oeuvre pour obtenir la vue demandee . il s'agit de la Translation, la Rotation, le changement d'echelles et la symetrie .

NB : On note $M'(X' , Y' , Z' , 1)$ transformee d'un point quelconque de $M(X , Y , Z , 1)$ par l'une des transformations de base ou leurs combinaisons .

1.5.6.1 - LE CHANGEMENT D'ECHELLE :

On l'appelle aussi l'echelonnage, qui sert a ajuster les dimensions de notre objet a la taille maximale que peut prendre nos trois axes, si nous representons un point par la matrice vecteur contenant les coordonnees homogenes caracterisant ce point, pour obtenir la matrice vecteur contenant les coordonnees etalonnees de ce point, on fera le produit de

la matrice d'échelle suivante , soit :

$$(X' , Y' , Z' , 1) = (X , Y , Z , 1) \times \begin{bmatrix} Ex & 0 & 0 & 0 \\ 0 & Ey & 0 & 0 \\ 0 & 0 & Ez & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$(X \times Ex , Y \times Ey , Z \times Ez , 1)$$

Se sont les termes de la diagonales de la matrice (4 x 4) qui permettent non seulement d'effectuer des étalonnages , mais aussi des déformations selon l'un ou l'autre axe . On peut donc soit s'en servir pour agrandir ou retrecir un objet en multipliant ou en divisant les trois coordonnées par la meme valeur , soit a déformer l'objet en ne faisant varier qu'une seule (ou deux) des dimensions.

1.5.5.2 - LES ROTATIONS :

Considerons les trois matrices de rotations autour des trois axes:

$$\text{Rot } x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \text{Cos}0 & \text{Sin}0 & 0 \\ 0 & -\text{Sin}0 & \text{Cos}0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rot } y = \begin{bmatrix} \text{Cos}0 & 0 & -\text{Sin}0 & 0 \\ 0 & 1 & 0 & 0 \\ \text{Sin}0 & 0 & \text{Cos}0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rot } z = \begin{bmatrix} \text{Cos}0 & \text{Sin}0 & 0 & 0 \\ -\text{Sin}0 & \text{Cos}0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

En exemple , on examinera Rot x , nous remarquerons que la premiere colonne de coordonnées concerne les X , et donc qu'il est normal que l'element de depart soit egal a 1 (Rotation autour d'un axe n'affecte pas cet axe) . il est tout a fait normal que les deux autres axes piv-

otent , c'est pourquoi chacun de ces axes se voit affecter une valeur SINUS et COSINUS pour etablir l'endroit exact ou se situe la projection du point de coordonnees (X' , Y' , Z') .

REMARQUE :

autour d'un axe arbitraire passant par l'origine du repere, definie par ses Cosinus Directeurs a , b , c et un angle 0 ; s'ecrit sous la forme suivante:

$$A = \begin{bmatrix} a.a (1- a.a).\text{Cos}0 & a.b (1-\text{Cos}0) - \text{Sin}0 & a.c (1-\text{Cos}0)+\text{Sin}0 & 0 \\ a.b(1-\text{Cos}0)+c.\text{Sin}0 & b.b+(1-b.b).\text{Cos}0 & c.b(1-\text{Cos}0)+\text{Sin}0 & 0 \\ a.c(1-\text{Cos}0)-b.\text{Sin}0 & b.c (1-\text{Cos}0) + c.\text{Sin}0 & c.c(1-c.c).\text{Cos}0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1.3.5.3-LES DEPLACEMENTS :

C'est la ou les coordonnees homogenes prennent toute leur valeur.

les déplacements sont tres utiles lorsqu'il s'agira plus tard d'animer notre objet . la matrice de transformation qui permet de les realiser est la suivante:

$$(X' , Y' , Z' , 1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix} \times (X , Y , Z , 1) = (X + T_x , Y + T_y , Z + T_z , 1)$$

1.5.6.4 - LA SYMETRIE (EFFET MIROIR) :

Un objet en 3D se refere a trois plans , le plan forme par les axes X et Y , celui forme par les axes Y et Z , et enfin celui se rapportant aux axes X et Z , il serait interessant de pouvoir obtenir une image-miroir de cet objet de l'autre cote d'un de ces plans .

Ainsi , la r'flexion de l'objet \mathcal{O} travers le plan XY a tout simplement pour effet de changer le signe de toutes les coordonnees Z a l'interieur de la matrice qui depeint l'objet .

Ce qui , selon le plan traverse , nous donne les matrices de transformation suivantes :

Par rapport au plan XOY :

$$S_{xy} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Par rapport au plan XOZ :

$$S_{xz} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Par rapport au plan YOZ :

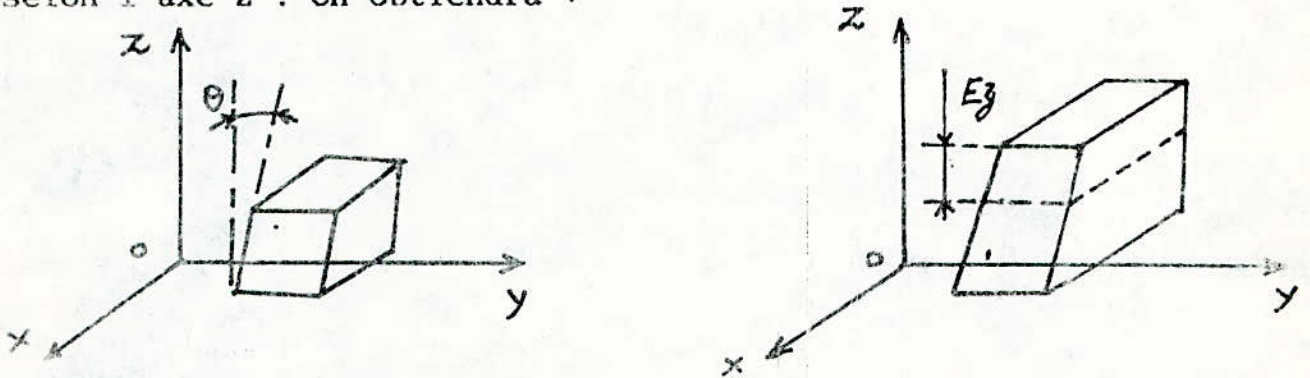
$$S_{yz} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1.5.7 - COMBINAISON DE TRANSFORMATIONS DE BASE :

Les transformations etant representees par des matrices , la combinaison de ces transformations se fera par le produit des matrices correspondantes .

Le produit de deux matrices n'est pas commutatif en general , l'ordre dans lequel celui-ci se fera influera donc sur le resultat . ceci peut etre aussi verifie dans la pratique .

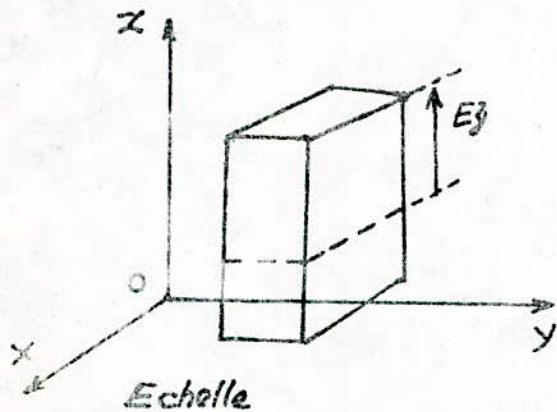
A titre d'exemple considerons un objet que l'on fait tourner d'un angle θ autour de l'axe Y, et auquel on fait subir un changement d'echelle selon l'axe Z. on obtiendra :



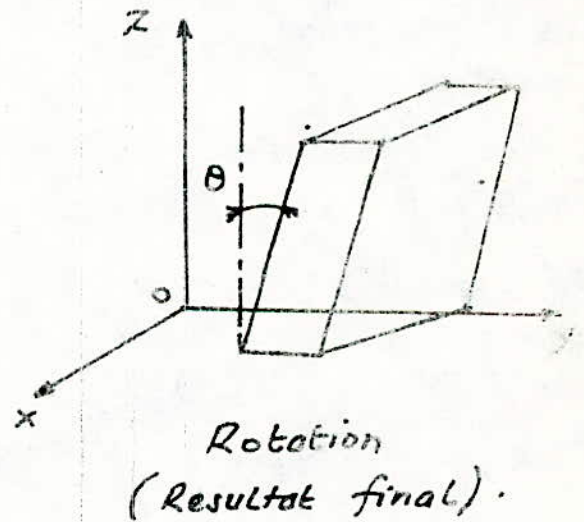
Cette operation etant faite dans le sens inverse, on obtiendra :
(figure B)

CONCLUSION :

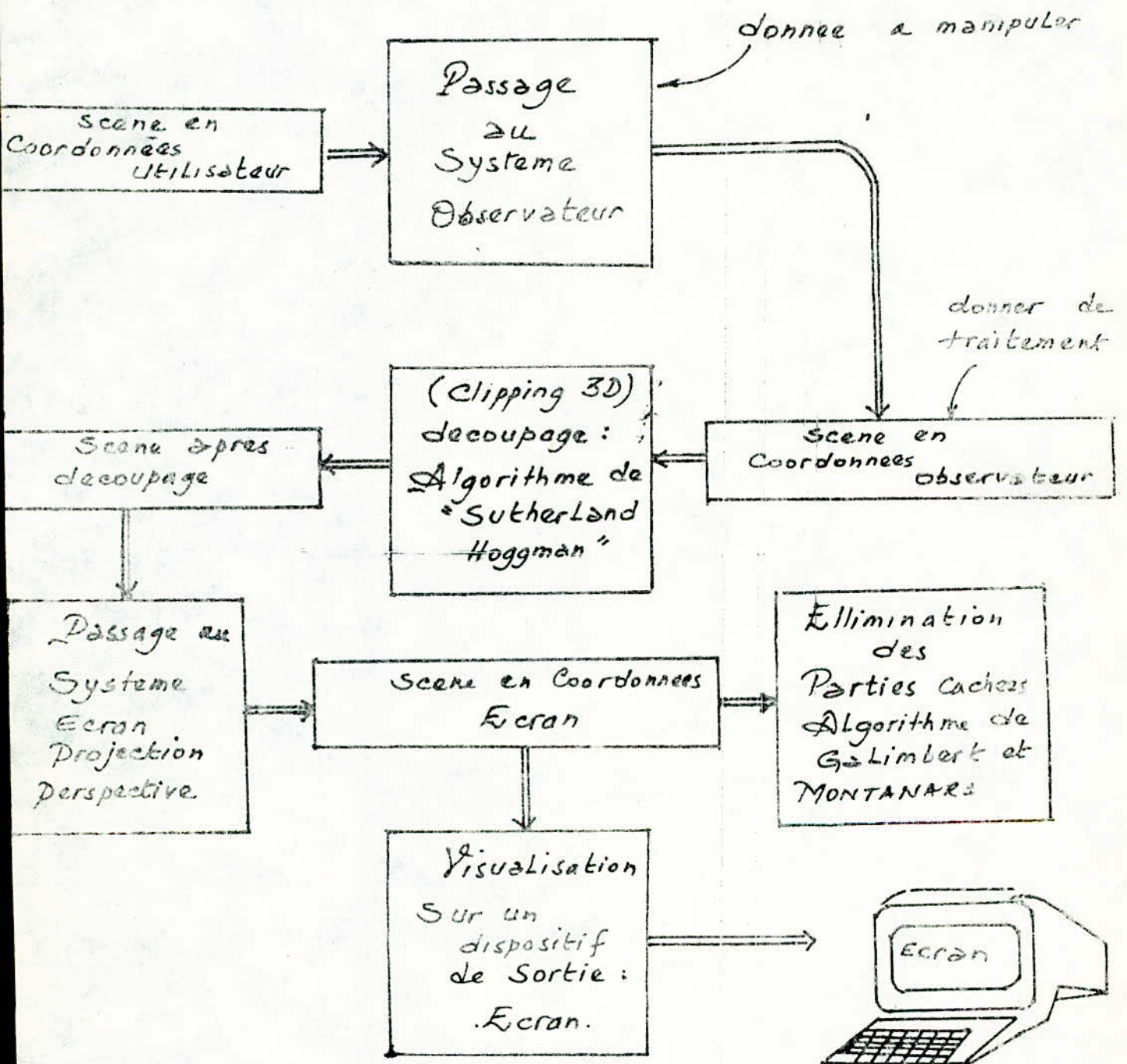
On constate donc que les deux resultats sont bien differents. l'ordre dont lequel ces transformations seront declenchees devra etre respecte.



(Figure B)



CHAINE DE VISUALISATION D'UNE SCENE 3D :



CHAPITRE II

TECHNIQUES DE LA VISION
ARTIFICIELLE
ET DE L'ANIMATION

II.1- TECHNIQUES DE LA VISION ARTIFICIELLE :

II.1.1 - INTRODUCTION :

La vision par ordinateur est un processus de traitements de l'information .

C'est un système où l'entrée est constituée par une séquence d'images , le système lui-même exploite ces informations contenues d'une image , de les traiter afin d'améliorer la qualité de l'image et de les rendre plus facilement interprétables par la machine afin d'identifier ou d'extraire des informations nécessaires , directement utilisables par le module de prise de décision . afin d'atteindre ces buts , le système utilise des stratégies bien définies , les deux mises en jeu sont :

- Les stratégies ascendantes:

Tendent de construire à partir de l'information sensorielle une représentation la plus abstraite possible (ex : un ensemble de primitives géométriques 3D) .

- Les stratégies descendantes:

Déduisent à partir de l'ensemble d'objets connus par le système une description compatible avec les primitives extraites de l'image . Il est alors possible de mettre en correspondance la représentation extraite de l'image avec les descriptions des objets afin de décrire les données sensorielles en termes de ces objets les connaissances mises en jeu sont de trois types :

- Physiques
- Géométriques
- Sémantiques

Les techniques utilisées pour atteindre ce niveau d'abstraction a

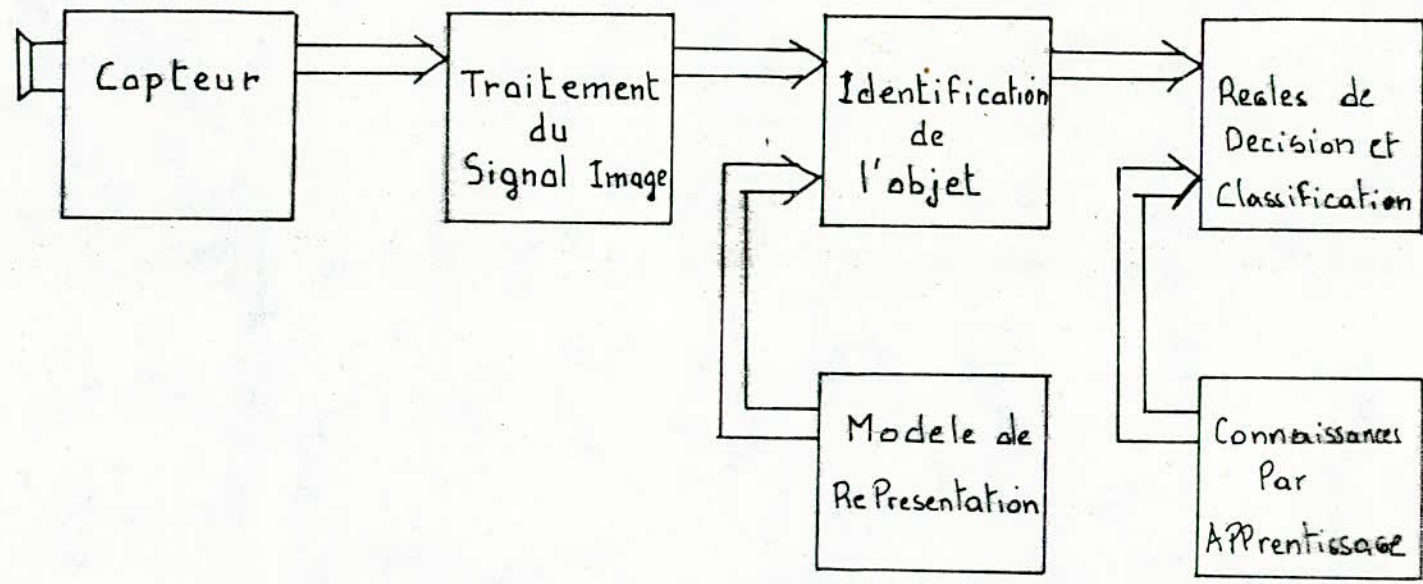
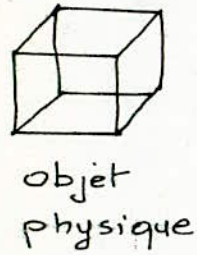


Fig: 2-1 : Systeme de vision artificielle

partir de proprietes concernant chacun des pixels d'une image; exigent un traitement mathematiques qui relevent essentiellement de trois domaines:

- La reconnaissance de formes.
- Traitement du signal 2D.
- Intelligence Artificielle.

Nous allons donner un apercu general des techniques de vision par ordinateur .

Un systeme de vision (Comme le montre la figure 2-4) met en oeuvre une ou plusieurs cameras qui captent les images et un ou plusieurs processeurs qui les interpretent. les applications de tels systemes sont; des a present illimitees.

_Outil de surveillance

_En bureautique ou il permet la saisie automatiques de messages.

_Inspection des produits pour y detecter les defauts ; il contribue a l'objectif de qualite totale.

Couplees a des equipements robotises, le systeme de vision permet de manipuler les objets et des outils pour fabriquer , assembler ou emballer.

Notre etude consiste la a presenter les techniques de l'analyse des images numeriques dont la maitrise permet de tirer le maximum des systemes de vision . c'est-a-dire le chemin qui nous guide de l'image a la decision.

II.1.2-CAPTEURS :

Divers types de capteurs sont utilises pour aquerir l'image d'une scene ; les plus repandus sont les cameras de television ; elles sont constituees d'un optique qui focalise l'image sur une couche photosensible ; a la sortie, Elles delivre un signal video proportionnel au

signal lumineux qu'elles recoit.

Il existe,deux types principaux de cameras:

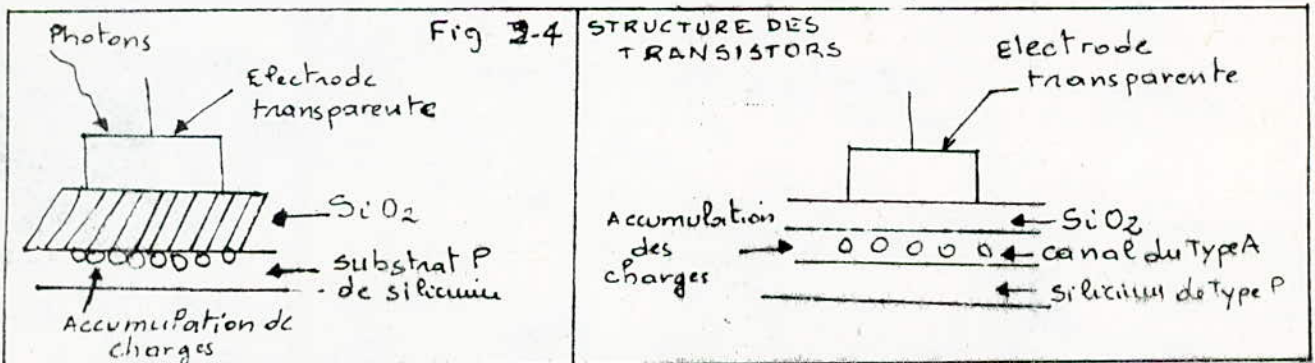
A-CAMERAS A TUBES : (Vidicon , etc ...)

Le type le plus repandu de camera utilise un tube de vue vidicon (voir fig 2-2) , un objectif forme l'image de la scene observee a l'interieur d'un tube a vide , sur un ecran ou cible photoconductrice qui prend en chacun de ces points une charge electrique proportionnelle a l'eclaircissement.

Un faisceau electronique convenable devie,explore successivement ces points , creant sur l'electrode de sortie un courant analogique variant avec la luminance du point analyse l'origine temporelle de chaque ligne est repere par une impulsion de synchronisation marquant la "fin de ligne" precedente.le debut d'une nouvelle trame est identiquement reperee par une impulsion particuliere de " fin de trame " (fig 2-3). Le courant au moyen d'une resistance de sortie permet l'obtention d'un signal video (fig 2-4).

B-CAMERAS A SEMI-CONDUCTEUR (C C D , C I D , etc...) :

Ce sont de nouvelles , dites solides,dont la surfaces sensible est composees d'une matrice d'element photosensible a technologie C.C.D (Charge Coupled Devices) ou C.I.D (Fig 2-4+5)



Le principe des cameras C.C.D est basee sur le concept de transfert de charges des semiconducteur.Les informations de provenance optique

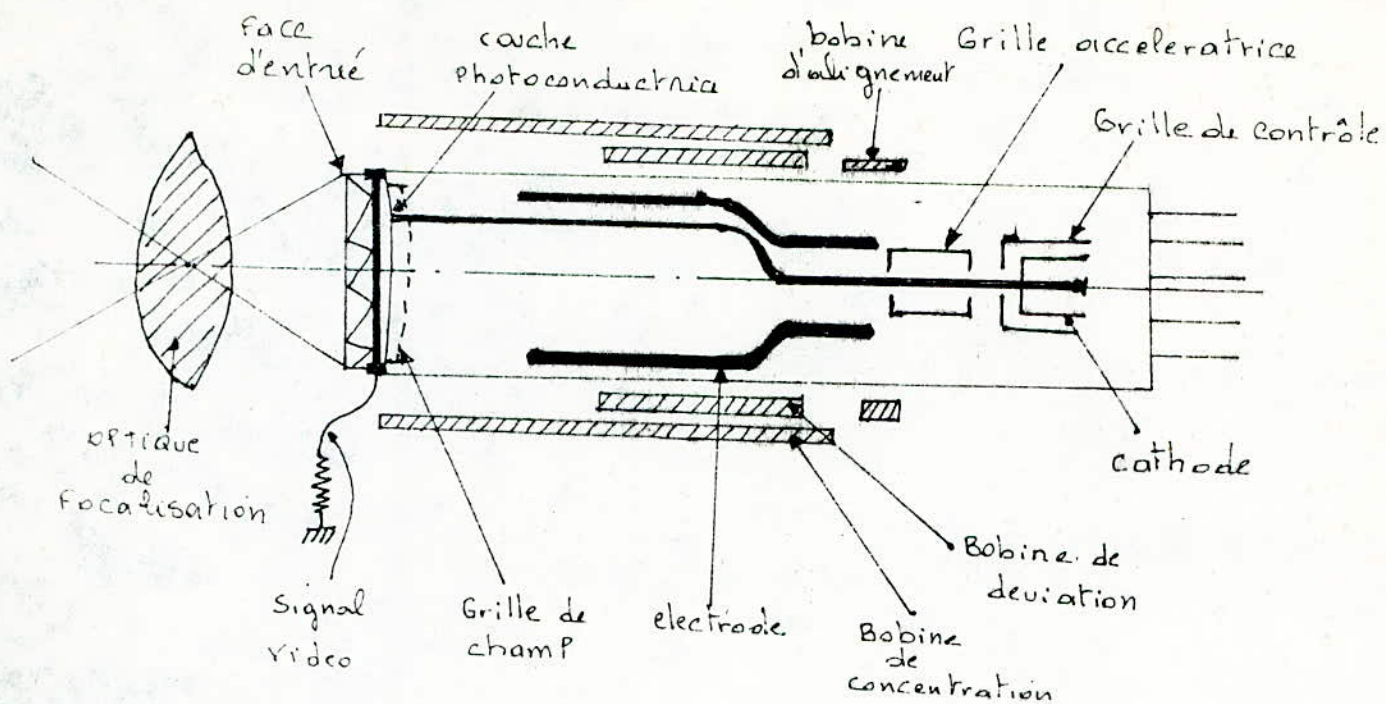


FIG 2-2 : schéma d'un tube vidicon

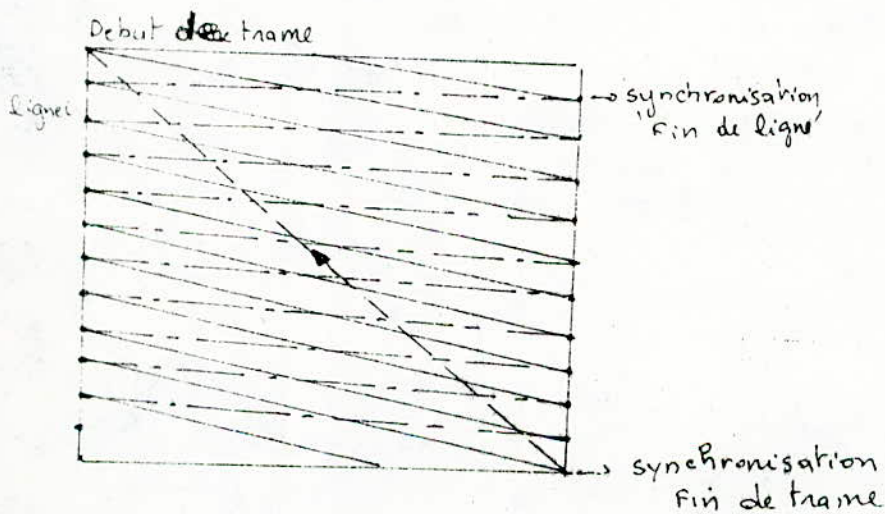


FIG 2-3: Boisoyage entrelacé d'une image

sont converties par effet photoelectronique au niveau de chaque grille en de paquets de charges electroniques . Le signal video en sortie est analogue a celui delivre par les cameras a tubes .

Les cameras de television a semiconducteur ont, sur la camera classique (a tube) , l'avantage d'etre beaucoup moins encombrantes et moins fragiles de plus leurs distortions geometriques sont pratiquement nulle . En effet les inconvenients majeurs des cameras a tubes sont:

- Le mauvais alignement du canon a electron .
- Les distortion geometriques dues au balayage de trame .
- La saturation due a un fort eclairage (effet BLOOMING) .

Malgres tous ceci , les cameras a tubes presentent une meilleure definition de l'image et un cout beaucoup plus faible .

C - LES CAMERAS INFRAROUGES :

C'est un dispositif qui permet de donner un signal video a partir d'un rayonnement electromagnetique emis par la matiere . sa construction necessite l'emploi de materiaux speciaux a par les elements optiques. l'objectif forme le plan image qu'exploire le dispositif opto-mecanique de balayage . le detecteur infrarouges recoit le rayonnement et fournit un signal traite ensuite par l'electronique de la camera. La camera etant equipee d'un seul detecteur , l'exploration de l'image infrarouge doit se faire par balayage opto-mecanique , type television ligne par ligne , puis trame par trame , avec un eventuel entrelacement pour diminuer le papillotement lors de la visualisation.

II.1.3-SYSTEME D'AQUISITION :

Dans cette parties , on s'interesse sur les techniques de num"risation de l'image recu par le capteur .

Cette operation consiste a convertir l'image video analogique issue du capteur en une image numerique sur laquelle un traitement par calculateur pourra etre effectue .

C'est-a-dire qu'une mise en forme du signal video est necessaire afin de l'introduire comme donnee dans un systeme de traitement cette etape comprend deux techniques :

A - L'ECHANTILLONNAGE :

Il permet de structure l'image video en cellules elementaires (ou "pixels") disposee selon un maillages approprie (carre , rectangle , hexagonal . . .) , en general le type de maillage utilise est carre ou rectangulaire , et ceci pour sa facilite de realisation .

Soit T la periode du balayage d'une ligne , un signal d'horloge synchronise sur une periode multiple de T , permet de figer la valeur du signal video , jusqu'a la prochaine impulsion d'horloge des compteurs de lignes et de colonnes associer a l'horloge, permettent de determiner

les coordonnees du point ainsi echantillonne (fig 2-6)



Signal video

Fig. 2-5-1
echantillonnage
de la
Ligne image

B - LA QUANTIFICATION DU SIGNAL VIDEO :

Un convertisseur analogique-numerique quantifie alors la valeur echantillonnee , selon une echelle lineaire ou non de niveaux de gris . Ce nombre varie entre 2 pour les images binaires et 256 pour les images multiniveaux .

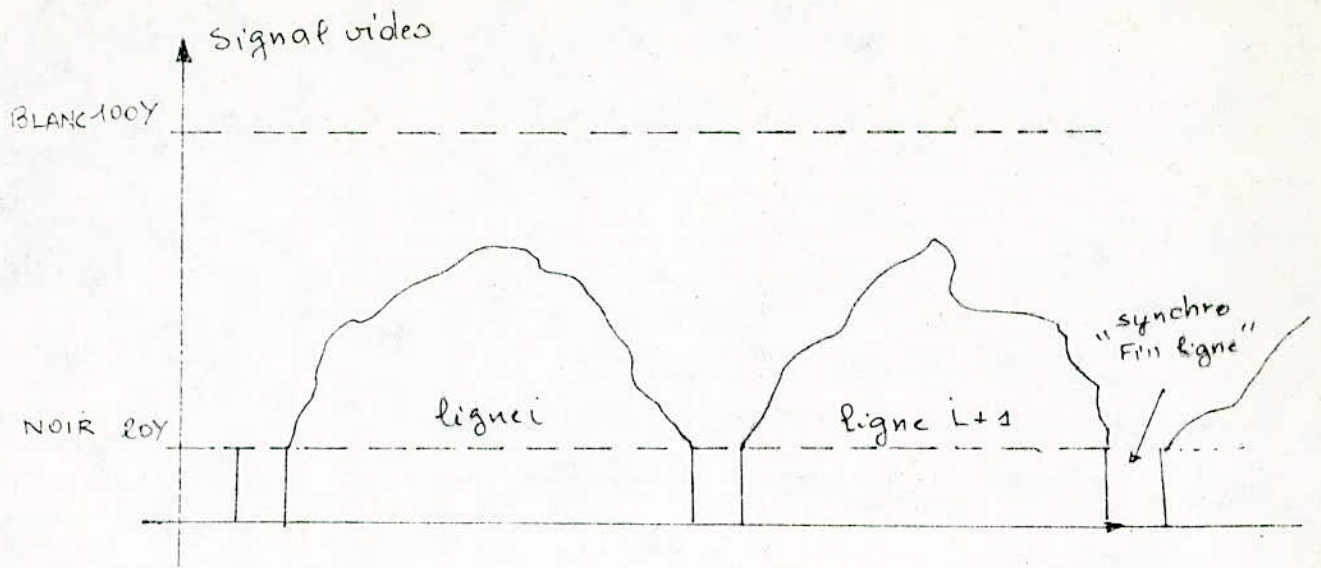


Fig 26-2: Evolution du signal video

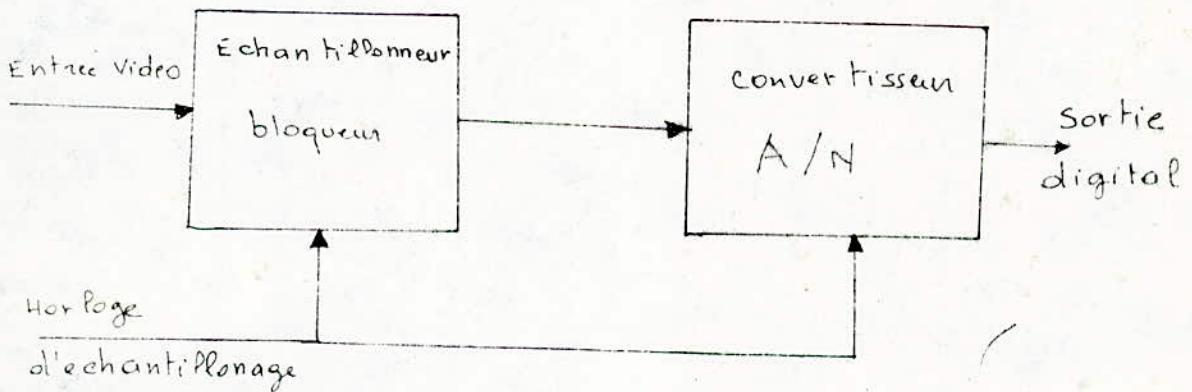


Fig 2-7 : chaîne de numérisation

Son choix depend principalement du traitement a effectuer et des possibilites des convertisseurs utilisees (fig 2.7) .

Une image ainsi numerisee se caracterise par :

- Ses dimensions M x N :

Ou M est le nombre de pixels sur une ligne et N est le nombre de pixels sur une colonne .

- Sa resolution :

C'est les dimensions " physiques " d'un element d'image " pixel " . On distingue la resolution horizontale liee a la frequence d'echantillonnage d'une ligne , et la resolution verticale liee a l'intervale entre deux lignes de balayage . Dans le cas ou celles-ci sont egales l'image est dite isotrope . Certains systemes de perception utilisent une resolution variable en fonction du niveau de precision requis par l'analyse de l'image .

- Sa definition :

C'est l'echelle du niveau de gris qui code l'intensite lumineuse des points de l'image . Plus fine est la definition , moins l'image sera sensible aux problemes d'eclairage , d'ombre ou d'eblouissement ... mais plus lourd sera le traitement (En robotique, la pr*cision requise necessite rarement plus de 32 niveaux de gris .

II.1.4 - TRAITEMENT D'IMAGES :

En traitement d'images , on utilise les techniques de traitement de signal qui sont les plus connues dans le domaine des signaux a une dimension (Les signaux temporels qu'etudient les electroniciens) parce que la nature de certains signaux et des operations que l'on effectue sur les signaux est plus facile dans le domaine frequenciel

que dans le domaine temporel, on préfère travailler non pas sur le signal d'origine lui-même mais sur sa transformée de Fourier (T.F) qui caractérise dans l'espace des fréquences .

Si $f(t)$ est le signal d'origine , sa T.F sera :

$$F(u) = \int_{-\infty}^{+\infty} f(t) e^{-j2\pi ut} dt$$

En traitement d'image le signal de départ n'est plus une fonction du temps $f(t)$ mais une fonction spatiale $f(x,y)$. x et y représente l'abscisse et l'ordonnée du point de l'image .

Sa transformée de Fourier à 2 dimensions est alors :

$$F(u, v) = \iint_{-\infty}^{+\infty} f(x,y) e^{-j2\pi(ux+vy)} dx dy$$

Dans le cas de signaux discrets , et c'est notre cas (l'image n'est en réalité définie qu'en un nombre fini de points " pixel ") , la T.F. devient transformée de Fourier discrète dont l'expression est :

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux+vy)/N}$$

Une autre notion importante est l'opération de convolution par définition le produit de convolution des deux fonctions $f(x,y)$ et $g(x,y)$ s'écrit $f * g(x, y)$ définie par :

$$f(x, y) * g(x, y) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m, n) * g(x-m, y-n)$$

Une partie des traitements que l'on effectue sur l'image consiste à remplacer la valeur d'un pixel donnée par une combinaison linéaire des valeurs de ce pixel et d'un certain nombre de ses voisins . On montre que

cette opération simple peut s'exprimer par le produit de convolution de la fonction $f(x,y)$ qui représente l'image avec une autre

fonction $h(x,y)$ et que l'effet de cette operation peut etre deduit du comportement de h dans le domaine frequenciel (c'est-a-dire sa T.F) .

L'explication reside dans le fait que la T.F du produit de convolution de 2 fonctions est simplement le produit des T.F de ces fonctions . Ainsi si l'on sait que l'image fournie par le systeme de captation est entachee d'un bruit a haute frequence , un moyen de s'en debarrasser est de supprimer toutes les composantes au-dela d'une certaine frequence en utilisant un filtre passe-bas dont la T.F est :

$$H(u,v) = \begin{cases} 1 & \text{si } D(u,v) \leq D_0 \\ 0 & \text{si } D(u,v) > D_0 \end{cases} \quad D(u,v) = \sqrt{u^2 + v^2}$$

Il y'a deux facon de realiser une telle operation,ou bien ; on retrouve la fonction spacial qui realise ce filtre et on fait le produit de convolution avec la fonction image $f(x,y)$ ou alors , on realise l'operation d'abord dans le domaine frequenciel , c'est-a-dire que l'on calcule la T.F de $f(x,y)$ (l'image) on l'a multiplie par $H(u,v)$ (la T.F du filtre) , et on obtient l'image finale en calculant la T.F inverse de ce produit.

Il y'a toujours avantage de travailler dans le domaine frequenciel car on dispose d'algorithmes tres perfoirmants pour realiser la T.F discrete (T F R :T.F rapide , ou la F F t: Fast Fourier Transform) .

1.4.1 - LE FILTRAGE :

Dans le processus de formation de l'image a partir de la scene,des degradations peuvent apparaitre (bruit , probleme d'echantillonnage , quantification de l'image , eclaireage , etc) .

L'operation la plus simple pour eliminer un bruit parasite consiste a moyenner l'information , c'est-a-dire realise l'operation suivante :

$$g(x,y) = \frac{1}{M} \sum_{(n,m) \in S} f(n,m) \quad s : \text{etant un voisinage du point } M$$

Cette operation effectuee uniformement sur toute l'image , a le defaut d'egalement adoucir les limites entre zones de differentes intensites ; il faut donc eviter de faire l'operation si un pixel est trop different de ces voisins , preuve qu'il appartient a une zone differente .

On obtient l'algorithme suivant :

$$g(x,y) = \begin{cases} \frac{1}{M} \sum_{(m,n) \in S} f(n,m) & \text{si } \left| f(x,y) - \frac{1}{M} \sum_{(m,n) \in S} f(n,m) \right| > T \\ f(x,y) & \text{sinon} \end{cases}$$

Une possibilite parmi les algorithmes de traitement d'image reduisent le bruit est d'utiliser le filtrage passe-bas (en frequence) . ce sont des filtres dont la T.F peut s'ecrire sous la forme suivante :

- FILTRE PASSE-BAS DE BUTHERWORTH :

$$H(u,v) = \frac{1}{1 + [D(u,v) / D_0]^{2n}}$$

- FILTRE PASSE-BAS EXPONENTIEL :

$$H(u,v) = e^{- [D(u,v) / D_0]^n}$$

- FILTRE PASSE-BAS TRAPEZOIDAL :

$$H(u,v) = \begin{cases} 1 & \text{si } D(u,v) < D_0 \\ \frac{D(u,v) - D_0}{D_0 - D_1} & \text{si } D_0 \leq D(u,v) < D_1 \\ 0 & \text{si } D(u,v) > D_1 \end{cases}$$

Si l'on veut mieux visualiser les limites entre les differentes zone de

l'image on a alors recours aux methodes de differentiations ou de filt

rage passe-haut .

Le gradient d'une fonction de deux variables $f(x,y)$ est un vecteur dont les deux composantes sont les derivees de f selon les directions X et Y et qui represente le degre de variation de cette fonction au point considere . Si la fonction est constante , le vecteur sera nul . Si la fonction change brusquement de valeurs , la norme de ce vecteur sera grande . La definition formelle est :

$$G [f(x,y)] = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

La norme est :

$$|G [f(x,y)]| = [(\partial f / \partial x)^2 + (\partial f / \partial y)^2]^{1/2}$$

le traitement est effectue par logiciel . cependant, il existe d'autres methodes , optiques ou materielles, qui necessitent des temps de calcul tres courts .

- La methode de filtrage par convolution de l'image par masques de taille $(M \times M)$, il en existe deux type :

A - Filtres lineaires :

Ce sont des filtres de la forme :

$$Y(i,j) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} H(k,l,i,j) \times f(k,l)$$

Avec :

- $Y(i,j)$ est la fonction image filtree au point (i,j)
- $H(k,l,i,j)$ est la reponse impulsionnelle du filtre au point (k,l) associe a une impulsion unitaire au point (i,j) .
- $f(k,l)$ est la fonction image initial du point (k,l) .

B - Filtres non-lineaires :

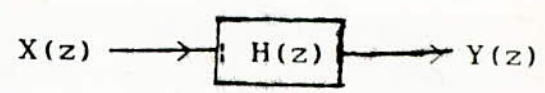
Ce sont des filtres qui , ainsi que les filtres lineaires servent a

la restauration des images en tenant compte d'une certaine connaissance a priori de l'image .

Par exemple un filtre lineaire, va traiter de facon identique les points de bas ou de haut contraste. Le resultat previsible est une deformation des zones de haut contraste , toutefois ce phenomene ne se produit pas si le traitement tient compte differement des zones homogenes et de fort contraste . Pour ce type de filtrage . le temps de calcul varie avec la taille du masque .

* Classification des filtres numerique :

La fonction de transfert s'ecrit :



$$M(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^N A_i \cdot z^{-i}}{\sum_{i=0}^M B_i \cdot z^{-i}}$$

avec $B_0 = 1 \iff y_n = \sum_{i=0}^N a_i x_{n-i} - \sum_{i=1}^M b_i y_{n-i}$

$z = \text{Exp} (j2\pi f t) = \text{Exp}(pT)$

T est la periode d'echantillonnage

Ou A_i et B_i sont les valeurs respectivement de X et Y au instant $i \cdot T$.

Il existe deux types de filtres numerique :

- Les filtres non-recursifs : (transversaux)

Ce sont des filtres dont tous les coefficients B_i sont nuls (sauf B_0) , la valeur de Y_n depend que des valeurs de X aux instants $n, n-1, \dots$

C'est-a-dire que la valeur de la sortie a l'instant n ne depend que des valeurs des entrees aux instants $n, n-1, \dots$

Ces filtres sont dits a memoire finie (Ou F.I.R : Filter.Finite.Impulse Reponse) . Sa reponse impulsionnelle est finie .

- Les filtres non-recursifs :

Ce sont des filtres dont un au moins des coefficients B_i est différent de zéro . La valeur de Y_n dépend des valeurs de Y aux instants précédents . C'est-à-dire que la valeur de la sortie à l'instant n dépend des valeurs d'entrées aux instants $nT, (n-1)T, \dots$, et aussi des valeurs des sorties aux instants $nT, (n-1)T, \dots$.

Ces filtres sont dits à mémoire infinie (Ou I.I.R.Filter : Infinite Impulse Reponse) , sa réponse impulsionnelle n'est pas forcément finie en temps .

1.4.2- AMELIORATION DU CONTRASTE :

C'est une discipline qui est liée à la subjectivité de l'observateur . Elle consiste à fournir une image , parfois différente de l'image initiale .

Pour des images peu contrastées, on utilise la technique d'amélioration de contraste dite par égalisation d'histogramme .

Supposons que la valeur des pixels (leur niveau de gris) ait été normalisée et soit comprise entre 0 et 1 (0 pour le noir et 1 pour le blanc).

Une image sombre comprendra de nombreux pixels proche de la valeur 0.

Contrairement la valeur d'une grande partie des pixels d'une image claire sera proche de 1 . (Fig)

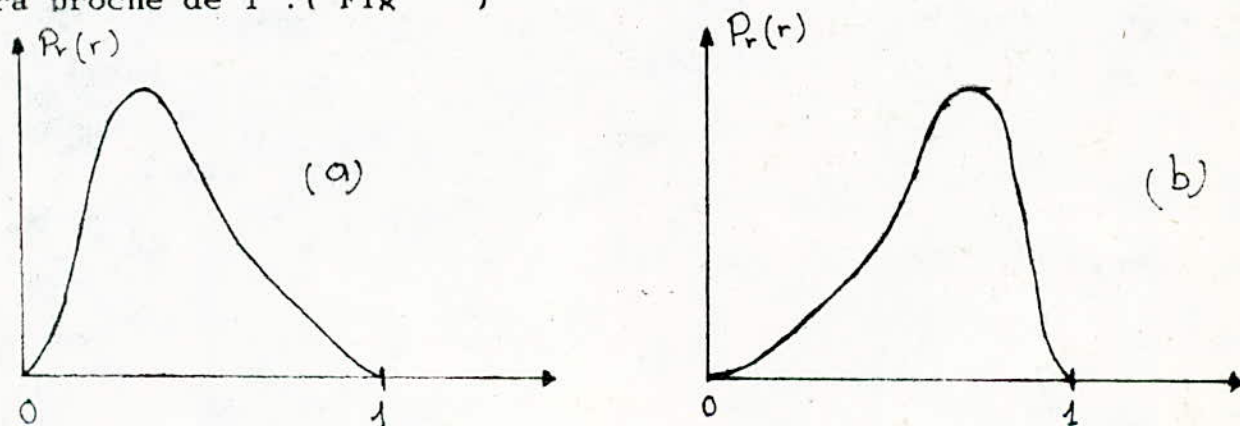


Fig : Histogramme (a) d'une image sombre
(b) d'une image claire

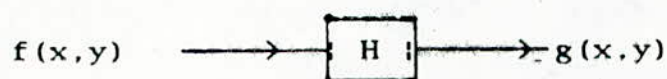
l'egalisation d'histogramme consiste a trouver une fonction $S=T(r)$, c'est-a-dire a chaque pixel d'intensite r une nouvelle valeur S , pour modifier cette repartition et obtenir une distribution relativement uniforme des intensites entre 0 et 1, cela afin d'augmenter la dynamique et donc d'ameliorer le contraste .

1.4.3- LA RESTAURATION :

Un probleme nouveau se pose lorsque l'image que l'on a obtenue des capteurs est pratiquement inutilisable parcequ'elle a ete deformee par un phenomene parasite qui n'est pas du bruit aleatoire , mais dont on connait les caracteristiques .

La restauration des images est l'ensemble des methodes developpees pour compenser les degradations , connues ou estimees , que subit l'image a cause d'un changement de support (reproduction , transmission , memorisation,...) ou d'un mauvais réglage de prise de vue .

Le modele de degradation peut se représenter ainsi :



Il s'agit de recuperer $f(x,y)$ a partir de $g(x,y)$, en ayant une idee de la perturbation H , on peut realiser l'operation inverse H et apprecier le resultat .

Cette operation se fait souvent d'une facon interactive , ce qui permet a l'utilisateur plusieurs tentatives jusqu'a l'obtention d'une image convenable .

Les techniques de restauration des images continues presentent l'avantage d'etre simples et de rendre effectif l'usage du domaine de fourier. Cependant, ces techniques connaissent plusieurs limitations importantes. Elles ne prevoient pas les effets des erreurs de recouvrement causees

par le sous-échantillonnage de l'image observée . Elles permettent la restauration pour les dégradations invariantes dans l'espace seulement . Il est difficile d'analyser les effets des erreurs numériques sur le processus de restauration et de développer des méthodes pour traiter de telles erreurs .

1.4.4- LE CODAGE ET LA COMPRESSION D'IMAGE :

Soit une image formée d'une partie carrée de $N \times N$ points et une quantification à B bits de chaque échantillon ; donc il faudrait $N \times B$ bits pour représenter l'image binaire .

Exemple :

Si $N = 256$ et $B = 8$ bits , on doit utiliser plus de 500 000 bits pour représenter l'image (appelée forme canonique de l'image numérique) . A cause du très grand nombre de bits utilisés ; il est préférable de chercher à réduire ce nombre de bits (à cause de la taille de la capacité mémoire ou canal de transmission qui ne permettent pas de mémoriser ou de transmettre l'intégralité des informations contenues dans une image) , cette opération s'appelle réduction de redondance ou compression d'images .

Cette technique utilise les méthodes de transformées , qui agissent sur le domaine transformé sur une transformée (linéaire ou non linéaire) de l'image originale .

Les transformées les plus utilisées sont celles de Fourier , Hadamard , et de Haar .

1.4.5- LA SEGMENTATION :

Segmenter une image , c'est effectuer une partition de cette image en région telle que chacune d'entre elle possède au moins une caractéristique que ne possède pas les régions voisines .

Le but de la segmentation est de fournir une description de l'image sous la forme d'une liste de régions caractérisées par des propriétés qui les différencient .

Les differents algorithmes employes pour faire la segmentation des images en regions se differentient par :

- Le type des proprietes recherchees pour les regions (parametre de segmentation) .

La maniere d'operer des regroupements de pixels pour former les region

On distingue generalement trois approches:

- La classification des pixels .
- La detection des contours .
- L'approche par regions .

1.5.1- CLASSIFICATION DES PIXELS :

La formation des pixels se fait sur la base d'une ressemblance entre pixels consideres individuellement , le principe general est : On definit un ensemble fini de classes de pixels et on cherche a affecter chaque pixel de l'image a une de ces classes . On forme les regions en regroupant les pixels connectes par la propriete d'appartenance a une meme classe . La maniere de definir les classes permet de distinguer deux groupes de methode de classification:

- Les methodes supervisees dans lesquelles les classes sont definies a priori .
- Les methodes non-supervisees pour lesquelles les classes ne sont pas connues a priori .

- CLASSIFICATION SUPERVISEE :

Dans ce cas , les classes doivent etre definies avant d'effectuer l'operation de classification des pixels de l'image , chaque classe etant definie par un ensemble d'echantillons constitues d'element representatifs de la classe consideres .

- CLASSIFICATION NON-SUPERVISEE :

Dans ce cas , on n'a aucune connaissance a priori sur les classes les classes seront definie apres examination de pixels . Parmi les methodes utilisees , citons les methodes permettant d'obtenir les regroupements de maniere iterative ,on optimisant un critere de similarite entre objet regroupes .

1.5.2- DETECTION DE CONTOURS :

La formation des regions va se faire en determinant leurs frontieres . nous appellerons :

- Points de contour :

tous points appartenant a la frontiere d'une region.

- Contour d'une region:

la courbe fermee telle que tous les pixels de la regions se trouvent d'un meme cote de cette courbe general , il peut y avoir un ensemble de courbes fermees obeissant a cette definition .

La recherche des contours se decompose en deux phases :

- Detection des points susceptible d'appartenir a un contour
- Formation du contour ,c'est-a-dire des sequences de points constituant

des courbes fermees verifiant la definition du point de contour.

Deux approche sont possible :

- Detection des points sur toute l'image , choix des points frontieres possibles,elimination des faux points de contour recherche de nouveaux points de contour qui n'avaient pas ete retenus lors de la phase de detection ;

- Detection des droites et des courbes : Dans ce cas , on recherche directement un ensemble de points de contour constituant droites ou courbes et detectant eventuellement direction ,concavite,longueur,etc.

1.5.2.1- DETECTION DE POINTS SUSCEPTIBLES D'APPARTENIR A UN CONTOUR :

- Detection des points sur toutes l'image:

Dans une image , un contour est determine par une variation rapide des niveaux de gris de pixels consecutifs . Cette variation peut etre detecte par le maximum d'une derivee premiere d'un point image ou par le passage par zero d'une derivee seconde .

Les operateurs les plus utilise dans ce domaine sont :

- Operateur gradient :

L'operateur gradient d'une image en chaque point est un vecteur dont la phase indique la direction de l'image dans laquelle le taux de changement du niveau de gris est le plus grand , et dont le module es une mesure de ce taux de changement maximal .

Les expressions du module et de la direction du gradient sont donnees par :

$$\text{Grad } L(i,j) = [(\partial L(i,j) / \partial i)^2 + (\partial L(i,j) / \partial j)^2]^{1/2}$$

$$\alpha(i,j) = \text{Arctg} \frac{\partial L(i,j) / \partial i}{\partial L(i,j) / \partial j}$$

- Operateur laplacien : C'est un operateur differentiel isotrope

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Une approximation discrete de cet operateur est :

$$L(x,y) = f(x,y) - \frac{1}{4} [f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y)]$$

1.5.2.1- DETECTION DE POINTS SUSCEPTIBLES D'APPARTENIR A UN CONTOUR :

- Detection des points sur toutes l'image:

Dans une image , un contour est determine par une variation rapide des niveaux de gris de pixels consecutifs . Cette variation peut etre detecte par le maximum d'une derivee premiere d'un point image ou par le passage par zero d'une derivee seconde .

Les operateurs les plus utilise dans ce domaine sont :

- Operateur gradient :

L'operateur gradient d'une image en chaque point est un vecteur dont la phase indique la direction de l'image dans laquelle le taux de changement du niveau de gris est le plus grand , et dont le module est une mesure de ce taux de changement maximal .

Les expressions du module et de la direction du gradient sont donnees par :

$$\text{Grad } L(i,j) = [(\partial L(i,j) / \partial i)^2 + (\partial L(i,j) / \partial j)^2]^{1/2}$$
$$\alpha(i,j) = \text{Arctg} \frac{\partial L(i,j) / \partial i}{\partial L(i,j) / \partial j}$$

- Operateur laplacien : C'est un operateur differentiel isotrope

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Une approximation discrete de cet operateur est :

$$L(x,y) = f(x,y) - \frac{1}{4} [f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y)]$$

base . Cette partition est realisee en fonction d'un certain nombre d'hypotheses ou de connaissance a priori .

Puis , par etapes successives , ces regions sont remodelees suivant la nature de remodelage , on distingue 3 methodes :

- METHODE PAR DIVISION :

On divise l'image globale en grande region etant elle meme divisees et ainsi de suite jusqu'a ce que la region produite verifient un certain critere d'informite (moyenne et variance a un seuil donne) . Le choix de ces grandes regions peut etre fait arbitrairement .

- METHODE PAR FUSION :

On part de petites region constituees de quelques pixels adjacents, a chaque etape, on va fusionner entre elles certaines regions adjacentes .

- METHODE PAR DIVISION-FUSION :

On utilise successivement les methodes precedentes , par exemple , on commence par effectuer une division , puis on va essayer de regrouper les regions adjacentes .

CONCLUSION :

L'approche par les regions est d'un cote eleve en temps de calcule , en espace memoire .

Les parametres issus de l'etape de segmentation sont divisees en deux classes :

- Les parametres intrinseques : qui definissent une region .
- Les parametres relationnels : qui definissent les relations (ex:adjacence) entre ces regions .

A - Parametres intrinseques :

Ces parametres permettent de decrire les caracteristiques propres de regions . Il sont calcul*s en general dans les coordonnees du plan image . Outre les parametres issus de la segmentation tels que le perimetre et surface , on peut distinguer :

- Les moments d'ordre p,q :

$$M_{pq} = \iint_S x^p y^q dx dy$$

. (x^p, y^q) coordonnees x,y d'un pixel de la region de surface S .

. M_{00} represente la surface de l'objet

- Le barycentre X_g, Y_g :

$$X_g = M_{10} / M_{00} \quad ; \quad Y_g = M_{01} / M_{00}$$

- Distance minimales et maximales du barycentre aux points de contour.
- Les axes principaux d'inertie .

Ces parametres sont faciles a calculer est caracterise relativement bien les regions considere , les deux dernieres sont parmi les plus significatif pour la localisation d'un objet .

B - Parametres relationnels :

Ils expriment les relations existant entre les regions trouvees lors de la segmentation . L'une des relation les plus importantes est celle de connexite .

Deux pixels sont connexes au sens des quatres voisins s'ils ont la meme propriete caracteristique (ex : le meme niveau de gris) et

s'ils sont situés dans la direction horizontale ou verticale . (Fig 2-7)

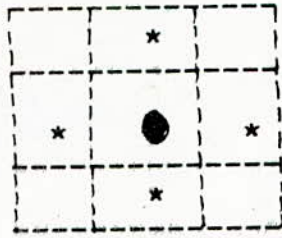


Fig 2-7

La connexité au sens des voisins est similaire à celle des quatre, mais permet la connexion dans toutes les directions . (Fig 2-8)

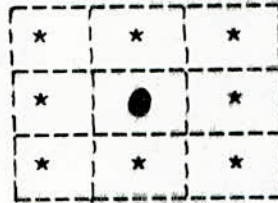


Fig 2-8

Elle traduit la propriété d'adjacence ou de non-adjacence . Tous les moyens sont bons pour caractériser une région et pouvoir la comparer efficacement avec des modèles de référence présents en mémoire .

On peut prendre en compte les propriétés topologiques (ex : Nombre d'Euler)

II.1.5- RECONNAISSANCE DES FORMES :

Dans cette partie, l'image a une description rationnelle et condensée de ces différentes composantes , que l'on a d'abord débarrassée des bruits et des déformations qui l'avait entachée et que l'on a ensuite représentée à un niveau conceptuel supérieur à celui du simple pixel . Le module de reconnaissance peut alors entrer en fonction .

Il ne faut perdre de vue que le but final est de comparer ce que le système de vision artificielle perçoit , avec un certain nombre de modèles qu'il a en mémoire et qu'il tente de retrouver dans la scène analysée .

La reconnaissance de forme est un domaine très riche et on dispose maintenant de théories et d'algorithmes puissants . Le problème de

base peut s'exprimer ainsi :

On doit classe un objet donne parmi un ensemble de classes de parametre qui caracterisent cet objet et d'une description des differentes classes .

On notera X le vecteur representant les differentes caracteristiques de l'objet (Moments geometriques par exemple) plusieurs cas se presentent selon la quantite d'information que l'on a , a priori , a sa disposition .

Si on connait deja plusieurs representants de chaque classes avec les parametres qui leurs sont associes .

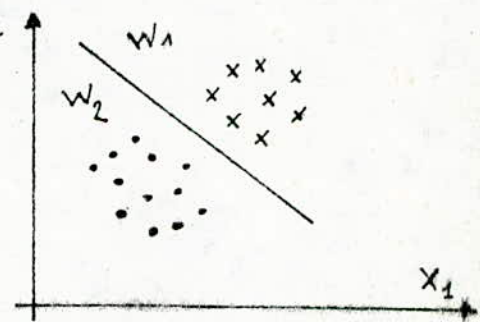
Le but c'est de tracer des frontieres entre les classes , de telle sorte que l'on pourra classe un objet uniquement par sa position relativement a ces frontieres . (fig 2-9)

Ces limites entre classes seront des lignes dans le cas elementaire ou le nombre de parametres ne depasse pas 2 , et plus generalement des hyperplans quand la dimension est superieur a 3 .

Dans le cas simple de la figure suivante :

$$d(X) = W_1 \times X_1 + W_2 \times X_2 + W_3 = 0$$

Fig 2-9



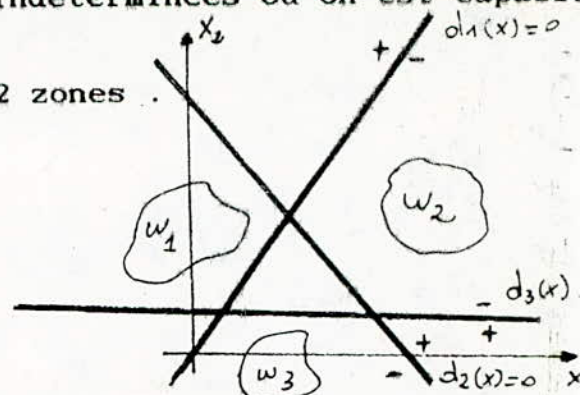
Il faut trouver un vecteur W (vecteur de ponderation tel que $d(X) = W \cdot X = 0$ represente l'equation de la droite frontiere et pour un objet quelconque , on puisse dire qu'il appartient a la classe W_1 si $d(X) > 0$ et a la classe W_2 si $d(X) < 0$.

La fonction $d(X)$ est une fonction de decision .

Dans le cas de 3 classes ou plus , on va tenter de les separe 2 a 2 et d'en deduire les regions de l'espace ou l'on pourra raisonnablement porter un jugement sur un objet inconnu . (fig 2-10)

Cette methode fait apparaitre des zones indeterminées ou on est capable de faire reellement la difference entre 2 zones .

Fig 2-10: Fonctions de decisions associées a 3 classes



Ce procede peut etre entierement automatique . Dans ce cas, on presente au systeme de vision une serie de tous ces objets qu'on veut plus tard lui faire reconnaitre sans meme lui dire quel objet , il est entrain d'observer .

Lui meme va calculer les parametres de chacun des objets et va tenter de les grouper en classes homogenes (Cluster Seeting) .

Si les differents types d'objets ne se ressemblent pas trop, le systeme parviendra a rassembler dans chaque classes les objets de meme type, le volume de la classe etant dimensionne par la dispersion que presente les valeurs des parametres des objets de meme type .

Il calculera ensuite lui-meme les fonctions de decision qu'il doit appliquer et pourra alors identifier un objet inconnu .

Il faut noter qu'il n'y a pas reellement de limite entre la phase d'apprentissage et la phase de reconnaissance .

Le processus est plus rapide si on indique chaque fois l'identite de l'objet a la machine . Le systeme sait alors combien de classes il faut constituer et a quelle classe appartient chaque objet . Il ne lui rest plus qu'a determiner les fonctions de decision .

Une solution intermediaire consiste a corriger la machine quand celle ci fait une classification fausse .

L'autre approche a la reconnaissance des formes est l'approche statistique , a partir de donnees concernant les objets d'une meme classe , une regle de classification est deduite pour minimiser la probabilite d'erreur .

Si l'on parvient à estimer un certain nombre de fonctions de probabilités propres aux objets que l'on étudie, alors on peut obtenir une classification optimale.

L'objet X appartient à la classe W_i qui maximise la quantité suivante:

$$d_i(X) = P(X / W_i) \cdot P(W_i)$$

où $P(W_i)$ est la probabilité a priori qu'un objet de la classe W_i apparaisse.

et $P(X / W_i)$ est la probabilité que l'objet ait les caractéristiques définies par le vecteur X sachant qu'il appartient à la classe W_i .

Tout le problème, consiste donc à estimer la quantité $P(X / W_i)$ ce qu'on peut tenter de faire au cours d'une phase d'apprentissage.

On a considéré jusqu'ici que l'ensemble de paramètres, regroupés en un

vecteur X était fourni par le module de traitement de l'image et qu'il

n'y avait aucune raison d'attacher plus d'importance à un paramètre plutôt qu'à un autre.

Dans la réalité, certains d'entre eux sont plus discriminants que d'autres. Il est alors intéressant de déterminer ces paramètres pour

mettre spécialement l'accent sur ceux-ci.

On utilise généralement, le concept d'entropie qui est une mesure statistique du degré d'incertitude de désordre définie comme suit :

$$H_i = - \int P_i(x) \cdot \log P_i(x) \cdot dx$$

Avec $P_i(x)$: Densité de probabilité dans la classe i .

On va donc choisir parmi les paramètres ceux qui réduisent le degré d'incertitude. Cela revient à diminuer la dispersion au sein d'une classe pour pouvoir conclure plus facilement quant à l'appartenance d'un objet donné à l'une des classes, puisque celles-ci auront été de quelque sorte mieux regroupées.

L'utilisation d'algorithmes numériques, vont décider qu'il suffit d'

jeu de variables reduit qu'il suffit de prendre en compte ,les autres ajoutant finalement peu d'informations supplementaires .

Les logiciels qui constituent la partie intelligente de la plus part des systemes de visions performants realisent une bonne partie des fonctions qu'on vient de decrire .

Un exemple est le logiciel Caiman d'ITMI ou ANIMA de CEA . En plus de ces operateurs dont le deroulement est relativement mecanique,certains de ces logiciels font appel a des regles pour l'identification des objets de la scene , ainsi le P.V.V d'ITMI est capable en cours de traitement d'emettre des hypotheses sur l'identite de l'objet et oriente le calcule pour verifier l'exactitude de ces hypotheses .

CONCLUSION :

Le traitement automatise d'images fait appel a des ^{outils} de plus en plus sophistiques soit pour approcher la vision humaine (ECLAIRAGE , PERSPECTIVE , ANIMATION ,COULEUR) ; soit pour simuler la perception visuelle (CONTOURS , RELIEF , RECONNAISSANCE) . Dans ce qui suit , nous allons voir les techniques utilisees pour l'animation .

II.2- NOTIONS ET PRINCIPE DE L'ANIMATION :

II.2.1- DEFINITION :

L'animation est le découpage dans le temps du mouvement que l'on veut recréer en un certain nombre de positions intermédiaire chacune représentée par un dessin fixe qui sera donc légèrement différent du précédent et du suivant .

Pour que ce mouvement semble continu lors du visionnage de ces séquences d'images , il faut que le temps qui les sépare soit suffisamment court .

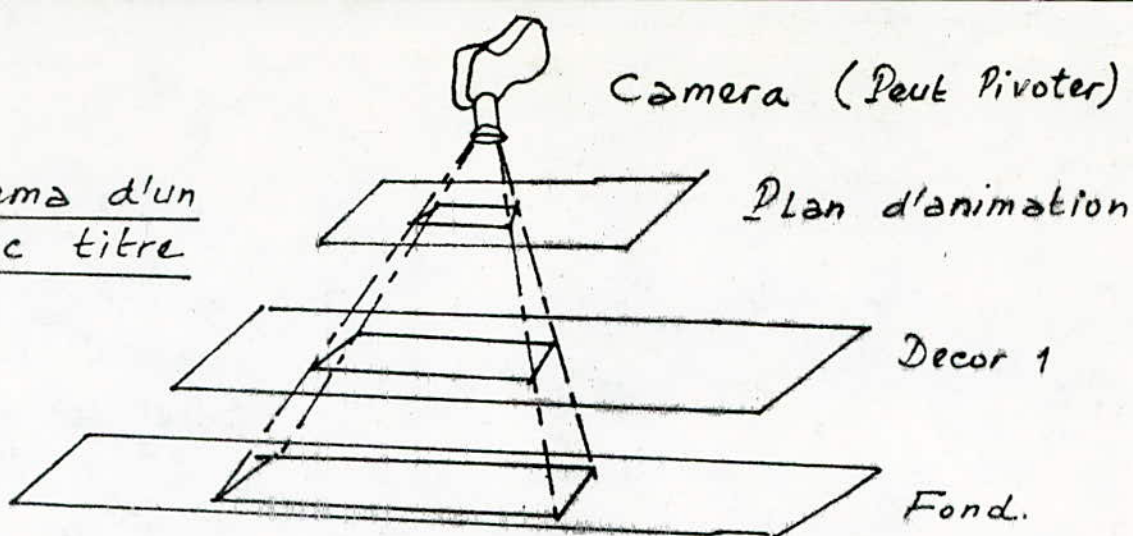
II.2.2 - L'ANIMATION TRADITIONNELLE :

Cette animation est généralement effectuée à l'aide d'un banc-titre : cet appareil est constitué d'un plateau d'animation généralement horizontal , comprenant plusieurs plans en cristal pouvant se déplacer horizontalement pour réaliser les mouvements, et verticalement, pour réaliser les travellings avant et arrière .

Dans cette animation , tout ce qui est fixe au moment de la prise de vue (personnages) apparaîtra comme mobile sur l'écran que tout ce qui est mis en mouvement (décors) apparaîtra comme point fixe au fond de l'écran .

Pour filmer , par exemple , une voiture longeant une route bordée d'arbres , il faudra faire reculer la route au rythme de la vitesse de la voiture . De même , pour décrire le mouvement de la caméra, celle-ci étant fixée au banc-titre , c'est le décor qui devra décrire un arc de cercle , avec changement de perspective .

Schema d'un
banc titre



II.2.3 - L'ANIMATION PAR ORDINATEUR :

Une animation traditionnelle peut se decomposer en trois grandes etapes :

- Elaboration d'une "story-board" et decoupage.
- Execution des dessins et decors sur celluloses et sur papier
- Tournage .

L'ordinateur peut facilement intervenir au niveau des etapes 1 et 3 : il s'agit du "motion control" ou controle des mouvements et celui-ci est dit intervalliste lorsqu'il intervient au niveau de l'etape 2 .

a - Le " motion control " :

Les déplacements des cellules , horizontalement et verticalement , et la simulation des rotations de la camera étant des operations souvent tres difficile a calculer , un ordinateur couple a un banc-titre pour aider l'operation et se charger de controler et de calculer les differents mouvements horizontaux en multiplan ,aussi que les zooms avant et arriere de la camera .

L'ordinateur assure ces operations en lançant les ordres correspondant a ces mouvements , lesquels sont retransmis a des organes de controle en liaison directe avec le moteurs du "banc-titre" qui pourront s'activer

selon l'ordre transmis . Il faut signaler que dans ce cas , l'ordinateur joue simplement le role de coordinateur des mouvements des parties du "banc-titre" et d'interface entre l'operateur et les moteurs .

b - Synthese d'image et l'ordinateur intervalliste :

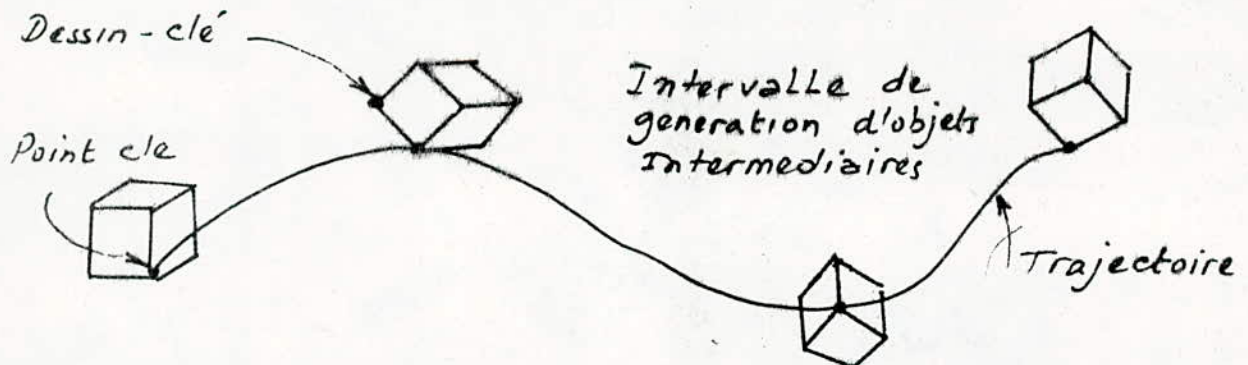
Dans ce cas , l'ordinateur intervient dans la deuxieme partie d'une animation (citee ci-dessus) et permet a l'utilisateur de creer des dessins , les visionner et va meme jusqu'a les animer .

Il existe en general , deux types d'animations par ordinateur :

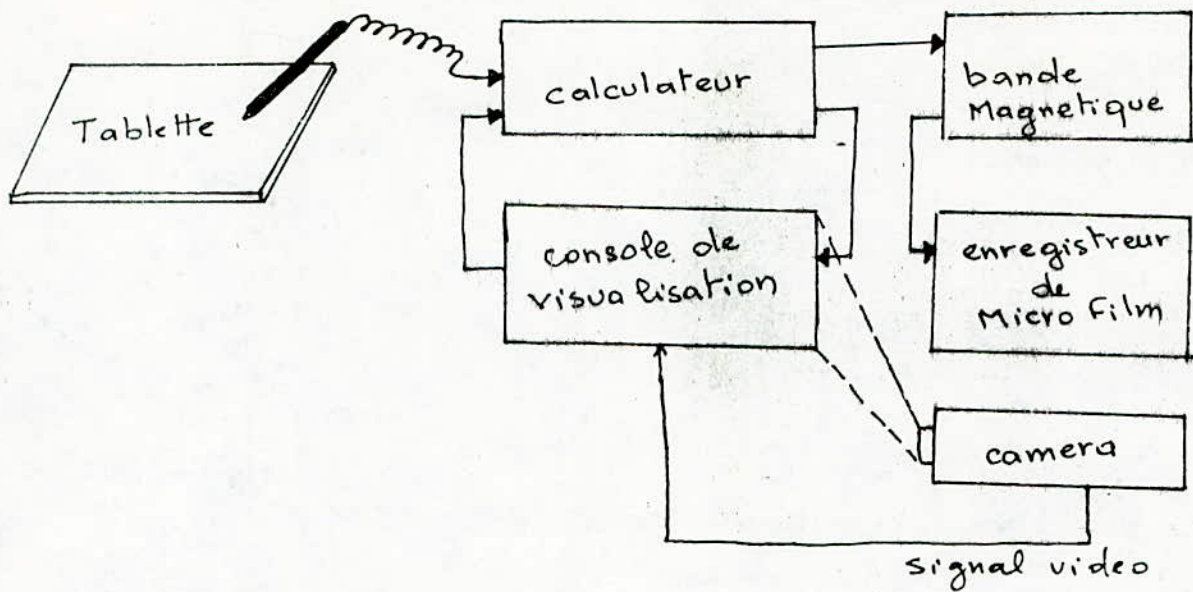
- Les animations a l'aide de dessins-cle :

Dans ce cas , l'animateur introduit ses dessins-cles a l'aide d'un organe d'entree (souvent une tablette graphique numerique equipe d'un stylet electronique) ou a partir d'un langage specifique qui lui permet de construire ses dessins .

L'ordinateur analyse alors le dessin et en fait une synthese . il ensuite les dessins-cles suivant et genere des dessins intermediaires , en faisant subir au premier dessin-cle un certain nombre de transformations qui lui permettront , en passant par des dessins intermediaires , d'arriver au deuxieme dessins (fig) , ces transformations sont des déplacements et des deformations .



- Les animations obeissant a des modeles mathematiques et contraintes precises :



Exemple de configuration associé
au traitement interactif

Ce type d'animation suppose une grande quantité d'informations sur l'aspect physiologique (par exemple pour déplacements, enchainements, etc...)

Ce type d'animation donne des résultats plus réalistes, mais demande une grande puissance de calcul et de stockage.

II.2.4- LES TECHNIQUES D'ANIMATIONS :

En animation , on distingue deux types de mouvements :

- Les déplacements
- Les déformations

A - Les déplacements :

Sont des mouvements qui n'influent pas sur la forme de l'objet . Ils influent par contre sur sa position et son orientation dans l'espace . Ce sont les translations , les rotations , les effets de miroir, etc..

B - Les déformations :

Peuvent modifier les attribus et la forme des objets sans pour autant , changer leurs positions relatives .

les déformations peuvent suivre des lois , c'est le cas des transformations homogenes : Projection , Similitude , etc . . .

Comme elles peuvent n'obeir a aucune loi precise , c'est le cas des metamorphoses .

II.2.5- LES DIFFERENTS TYPES DE PRODUCTION D'ANIMATIONS :

Dans le cinema traditionnel , l'illusion d'une image animee continue est creee grace a la diffusion de 25 images a la seconde . Cette vitesse necessite donc la production d'une image en 10ms au minimum . Pour cette raison , on distingue trois types de production d'animations :

1 - Animation temps reel :

Assure le calcul et l'affichage d'une image en moins de 10ms . Ce qui n'est possible que dans certains grands centres informatiques qui disposent de moyens de calcul considerables (ceci est souvent obtenu en reliant plusieurs ordinateurs) .

Ce type d'animations est utilise surtout pour les simulateurs .

2 - L'animation differe precalculee:

Dans laquelle les images sont produites au préalable et stockées ensuite dans des mémoires secondaires à accès rapide . il suffit donc, par la suite , d'utiliser ces 40ms pour l'affichage de l'image uniquement et non pas pour le calcul et l'affichage comme pour le premier type .

3 - L'animation differe vue par vue :

Dans ce cas , l'image produite sur l'écran de visualisation est enregistrée par l'intermédiaire d'une camera commandée par l'ordinateur lui-même . Ce qui permet d'avoir une indépendance vis-a-vis du temps de calcul des images et donc d'obtenir des images plus réalistes .

REMARQUE :

Dans le cas de l'animation en temps réel et vu le coût fabuleux de production de celle-ci, on se contente souvent de 12 images par seconde au lieu des 25 images par seconde du cinéma traditionnel .
Le tableau résume les différents cas d'utilisations des types de production d'animation .

console calcul	Modification d'une image < 40ms	Modification d'une image > 40ms
calcul d'une image < 40ms	Animation directe en temps réel	Animation differe vue par vue
calcul d'une image > 40ms	Animation differe precalculee	Animation differe vue par vue

EDITEUR 3D

1 - INTRODUCTION :

Pour construire des objets tridimensionnels , on dispose d'un éditeur , dans lequel ont été implantés des objets élémentaires prédéfinis qui constituent les éléments de base avec lesquels on pourra faire la construction cet éditeur permettra en plus de la création interactive de ces objets , de les visualiser , leur faire subir des transformations et mettra en disposition d'autres utilitaires , qui permettent de les manipuler .

2 - EXEMPLES D'ENTITES D'UN SYSTEME :

On distingue deux types d'objets :

- Les objets élémentaires prédéfinis .
- Les objets - utilisateurs .

2.1 - LES OBJETS ELEMENTAIRES PREDEFINIS :

Se sont des objets élémentaires mis à la disposition de l'utilisateur par introduction des paramètres nécessaires pour les obtenir.

a) LE POINT : Il est défini par ses coordonnées dans l'espace c'est à

dire :

$$\begin{cases} X: \text{ Pour les abscisses} \\ Y: \text{ Pour les coordonnées} \\ Z: \text{ Pour la profondeur} \end{cases}$$

b) L'ARRETE : Qui est un segment de droite dans l'espace , celle-ci est définie par la donnée des coordonnées de ces deux extrémités .

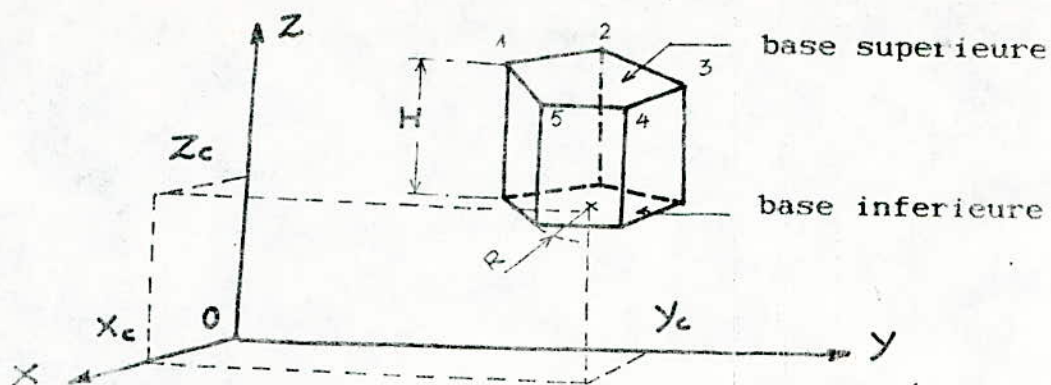
c) LES POLYEDRES REGULIERS :

Ils ont la caractéristique principale d'avoir leur base circonscrite dans un cercle . Les arêtes constituant cette base étant de même longueur .

Les paramètres permettant la construction de ces objets sont .

- Le centre du cercle (X , Y , Z)
- Le rayon du cercle (R)
- La hauteur du polyèdre (H)
- Le nombre de sommets de la base

Exemple : Pour un nombre de sommets egale a cinq (5) .



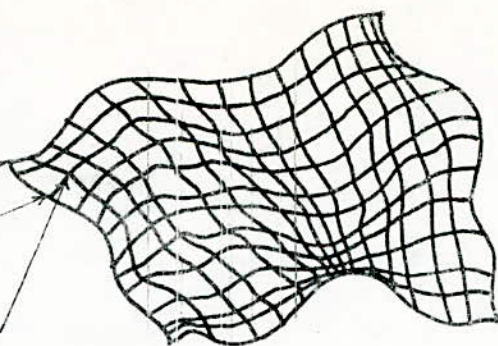
ces polyedres peuvent etre partages en trois types :

- * LES POLYEDRES REGULIERS DU TYPE 1 : Qui possedent deux bases identiques , et suivant le nombre de sommets nous distinguons :
 - Le cube : Nombre de sommets egale 5 et les arretes sont identiques .
 - Le parallelepipede : Nbre de sommets a 4 .
 - Le cylindre : Quand le nombre de sommets est assez grand et que la base approche un cercle .
- * LES POLYEDRES REGULIERS DU TYPE 2 : Se sont les polyedres dont l'une des bases est reduite a un point : Les pyramides , les cones (Quand la base approche un cercle) .
- * LES POLYEDRES REGULIERS DU TYPE 3 : Se sont les polyedres dont la hauteur (H) est nulle , ce qui donne des objets surfaciques reguliers : Triangles , Carres , cercles , etc ...
- * LES COURBES : Une courbe est construite par une methode d'interpolation , a partir des points de controles .
- * LES SURFACES GAUCHES : La modelisation de ces surfaces est relativement recente et a pour but de rendre possible , la realisation des formes les plus diverses .
Une surface est obtenue a partir du trace de deux familles de courbes , formant ainsi un fillet .

point de controle

1er point de controle

2eme point de controle



PRINCIPE :

L'obtention de telles surfaces , necessite la donnee des differents points de controle sous forme matricielle qui representera le modele et l'allure de la surface . Cette derniere sera fonction de deux parametres U et V (Appelles coordonnees curvilignes ou de GUAUSS) , qui sont pris dans l'intervalle [0 , 1] .

Cette surface sera decrite de la maniere suivante :

1er type de courbes : Garde U constant et fait varier V dans [0 , 1]

2eme type de courbes : garde V constant et fait varier U dans [0 , 1]

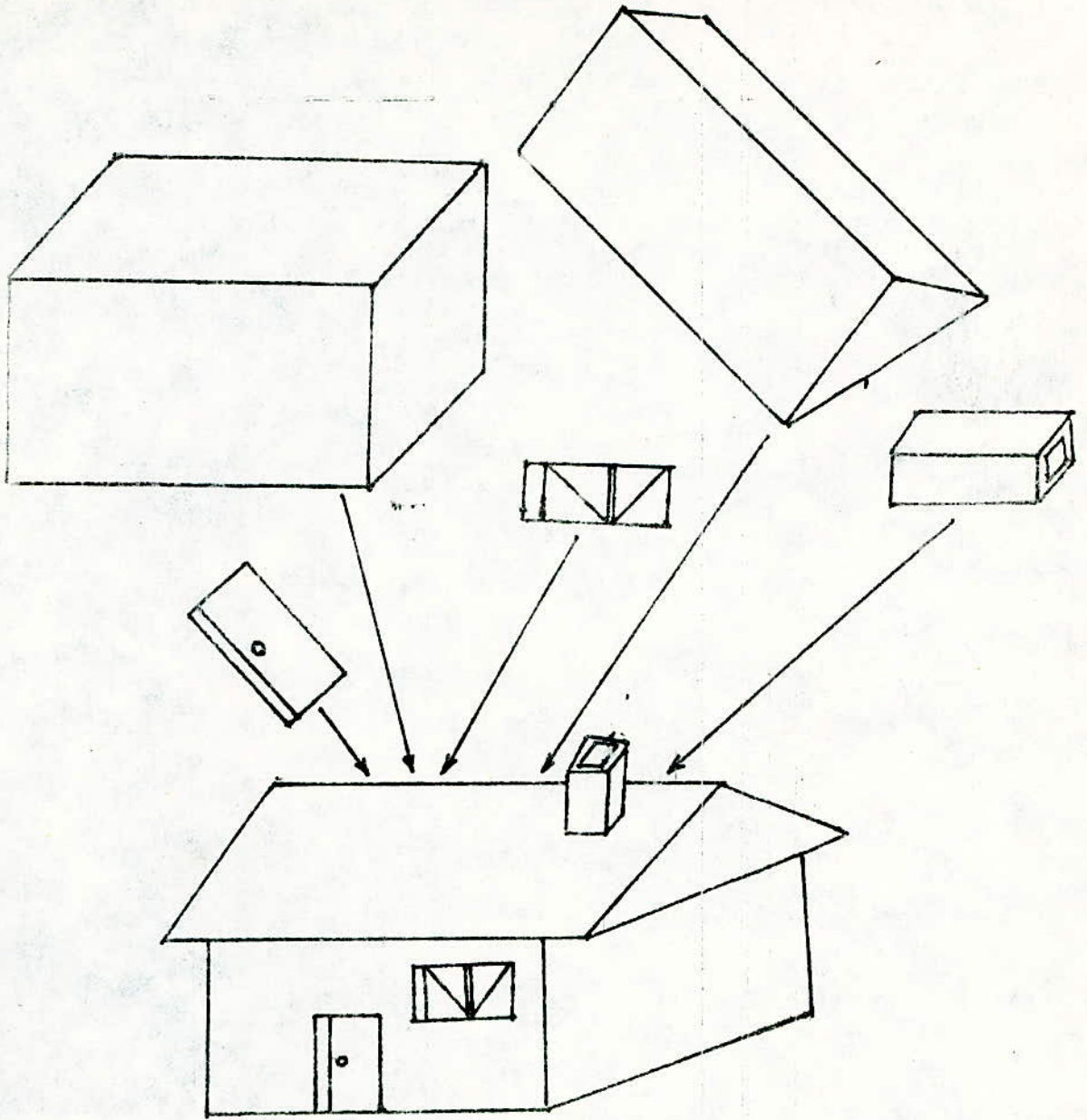
L'equation de la surface etant :

$$Q (U , V) = (X (U , V) , Y (U , V) , Z (U , V))$$

2.2 - LES OBJETS UTILISATEURS :

Ces objets sont construits a partir des objets predefinis (ou surfaces) , par exemple la construction des portes , des fenetres , des murs , etc ... , ensuite construire un objet maison constituee d'une reunion de tous ces objets .

Chaque objet utilisateur possede un nom , faisant partie de la " bibliotheque " et peut entrer dans la composition d'un autre objet utilisateur .



CHAPITRE III

PREPARATION A L'ANIMATION
ET MISE EN SCENE

III.1 - INTRODUCTION :

Cette étape constitue le noyau de l'animation d'images vu que l'expression d'un mouvement quelconque est décrite dans sa totalité à ce niveau.

L'expression d'un mouvement débute par la réalisation d'un certain nombre de croquis représentant les personnages animés dans leurs positions et attitudes qu'ils devront occuper à certains intervalles de temps, ces croquis (objets ou dessins) sont appelés dans la suite de ce rapport LES DESSINS CLES.

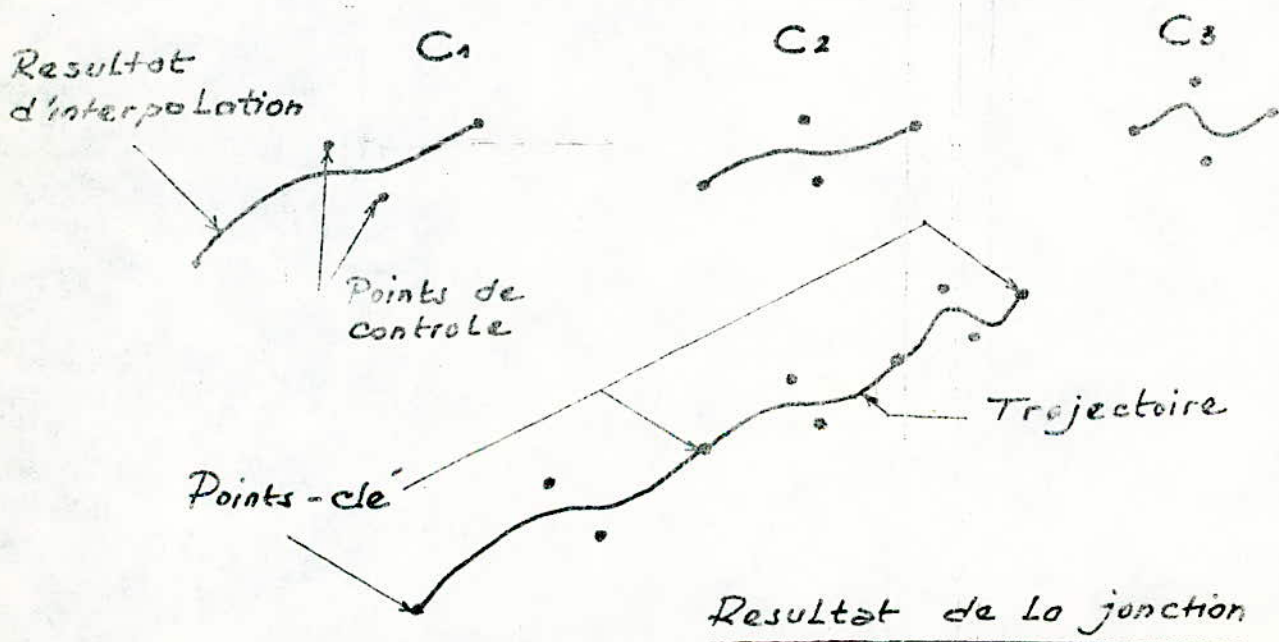
Toute la difficulté dans le cas de l'animation d'un mouvement que l'on veut naturel, réside dans le choix des dessins intermédiaires, c'est ce qui fait dire à MAC LAREN : "l'animation n'est pas l'art des dessins qui bougent, mais l'art des mouvements qui sont dessinés".

III.2 - PRINCIPE DE CONSTRUCTION DE TRAJECTOIRES :

3.2.1 - Définition :

Une trajectoire est construite par la jonction de deux ou moyennant les méthodes de BEZIER, B-SPLINE ou CATMUL-ROM-SPLINE, ces jonctions de courbes représentent des points particuliers appelés points-cles (ce sont les points sur lesquels seront les objets clés).

Exemple: soit trois courbes (C_1, C_2, C_3) définies séparément



Une question se pose :

Pourquoi joindre plusieurs courbes pour former une trajectoire et ne pas considerer simplement une courbe comme une trajectoire ?

vu que les methodes d'interpolation ne permettent pas le controle du " pas " de generation des points d'une courbe entre deux points de controle et que ces memes points de controle n'appartiennent pas obligatoirement a la courbe (Exemple : courbe B-SPLINE) ceux-ci ne peuvent etre consideres comme points cles de la trajectoire d'ou la necessite de decomposer cette trajectoire en plusieurs courbes , les extremités de ces courbes representent les points cles de la trajectoire obtenue (Voir annexe) .

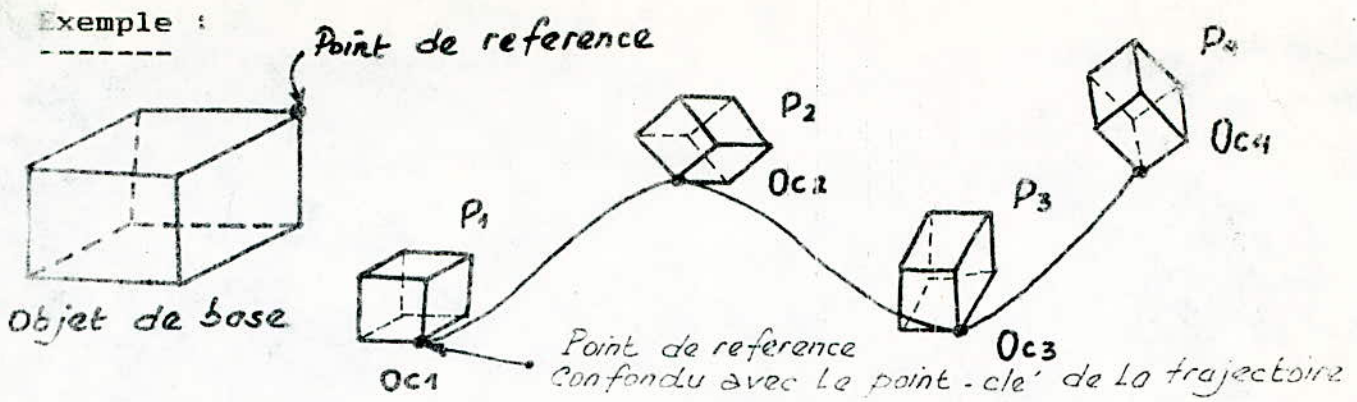
Remarques :

- Dans le cas de B-SPLINE la courbe obtenue ne passe pas obligatoirement par tous les points de controle .
- Les courbes C.R.S ne sont pas pratiques pour generer une trajectoire car les points obtenus ne sont pas disperses uniformement. Elles presentent ainsi des tous empechant une animation naturelle .
- Les courbes composant une trajectoire ne sont pas obligatoirement generees par la meme methode .
- Garder les memes coordonnees des points de controle donnees .

III.3 - LES OBJETS CLES :

3.3.1 - INTRODUCTION :

Apres avoir cree la trajectoire nous devons decrire l'evolution de notre objet sur cette trajectoire : des utilitaires permettant de definir des objets cles qui seront implantes respectivement a chaque point cle de la trajectoire . Ces objets representent l'etat dans lequel sera l'objet a animer, nous allons le clarifier dans un exemple dans les pages qui suivent .

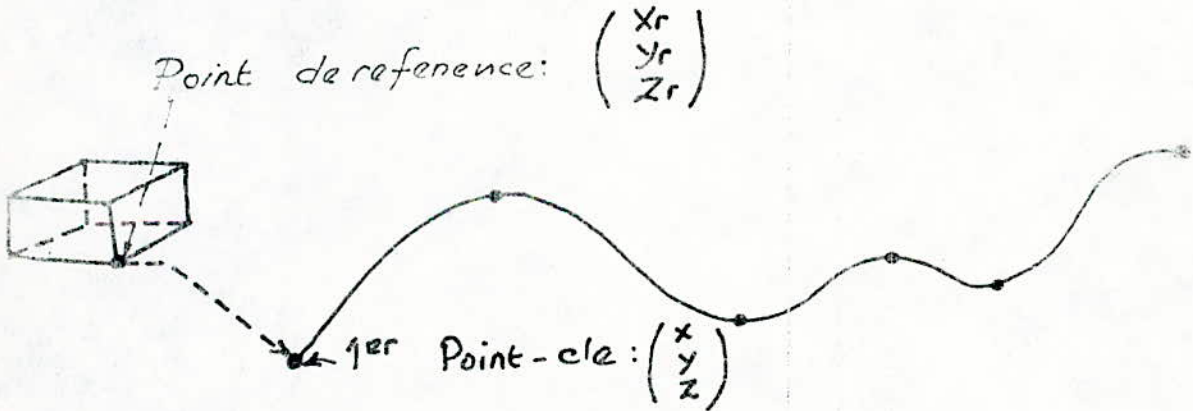


Ces points clés doivent être astucieusement choisis, afin qu'il y ait une corrélation entre les objets clés définissant la continuité du mouvement.

3.3.2 - MISE EN OEUVRE : -----

Nous choisissons l'objet à animer et la trajectoire sur laquelle il va évoluer. Les structures de données correspondant à ces deux entités seront chargées.

Sur l'écran nous verrons affichés la trajectoire et les points clés en évidence ainsi que la représentation du premier objet au niveau du premier point clé. Cet objet n'est autre que l'objet tel qu'il a été construit (au niveau de l'éditeur). Il sera implanté tel que son point de référence coïncide avec le point clé.



ALGORITHME GENERAL : -----

DEBUT

Selection de l'objet à animer

Choisir une trajectoire

\$vt : vecteur de translation \$:vt xc-xr
yc-yr
zc-zr

\$pr : point de reference \$

\$pcl: premier point cle \$

Calcul de vt : d'origine pr.d'extremite pcl

Tr : translation de l'objet (de vt)

Visualisation de l'objet-cle

Orientation de l'objet cle

SI : Existe un point-cle suivant

ALORS :

\$ pcc : point-cle courant \$

\$ pcs : point-cle suivant \$

Calcul de vt entre pcc et pcs : vt xcc-xcs

ycc-ycs

zcc-zcs

aller a tr

SINON

FIN

FINSI

FIN

III.4 - DETERMINATION DE LA VITESSE :

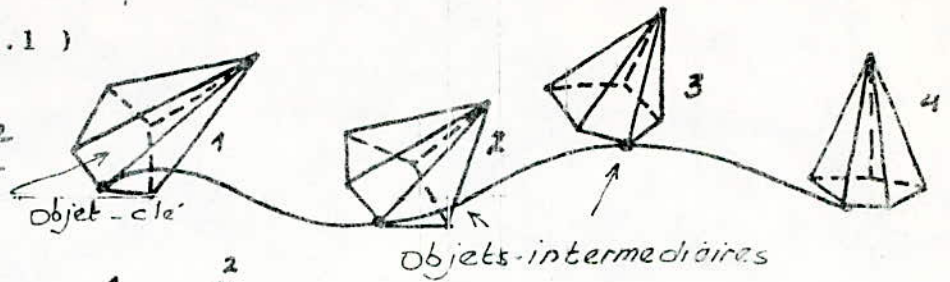
3.4.1 - DEFINITION :

L'illusion du mouvement est obtenue en faisant defiler un a un les objets intermediaires (objets-cle inclus) ce sont eux qui determinent au fait la vitesse du composant a animer .

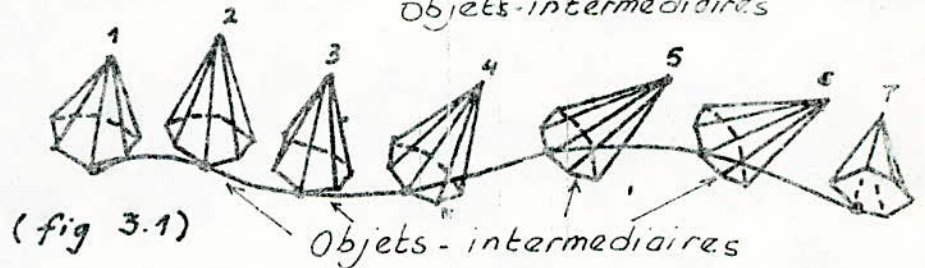
En effet si la distance entre deux dessins intermediaires est grande , la vitesse du composant dans cet intervalle est grande aussi . De meme si la distance est petite la vitesse le sera aussi . Autrement dit , si le nombre de dessin intermediaires est grand entre deux dessins-cle

la vitesse sera petite dans cet intervalle et inversement si le nombre est petit . (fig 3.1)

Mouvement Rapide



Mouvement Lent



(fig 3.1)

REMARQUES:

- * Le nombre de dessins intermediaires est inversement proportionnel a la vitesse du mouvement de l'objet anime entre deux points-cle .
- * La fixation de ce nombre ne semble pas interessante car tous les mouvements auront la meme vitesse .

Soit N_i le nombre d'objets intermediaires a generer entre l'objet-cle (i) et (i+1) ; ainsi le "Pasi" de generation de la tranche de courbe correspondante par un certain procede d'interpolation est egale a :

$$Pasi = 1 / Ni$$

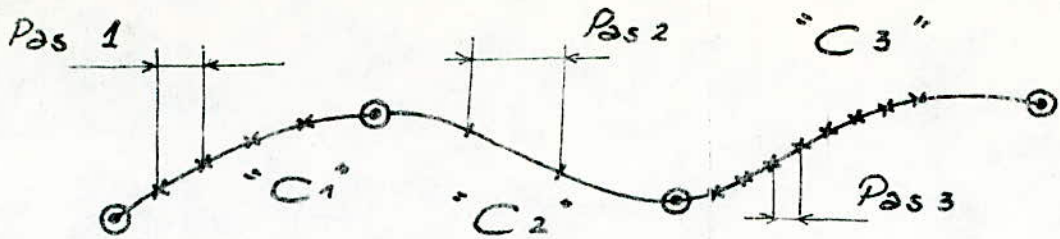
III.5 - VISUALISATION INDIVIDUELLE DU MOUVEMENT :

3.5.1 - GENERATIONS D'OBJETS INTERMEDIAIRES :

3.5.1.1 - INTRODUCTION :

Les objets intermediaires sont l'ensemble des objets generes automatiquement entre chaque paire de points-cle consecutifs .

En fait ce sont les points par lesquels passent les objets intermediaires qui sont les points generes par interpolation de chaque courbe de la trajectoire .



- A chaque point intermediaire , on fait correspondre un objet intermediaire .

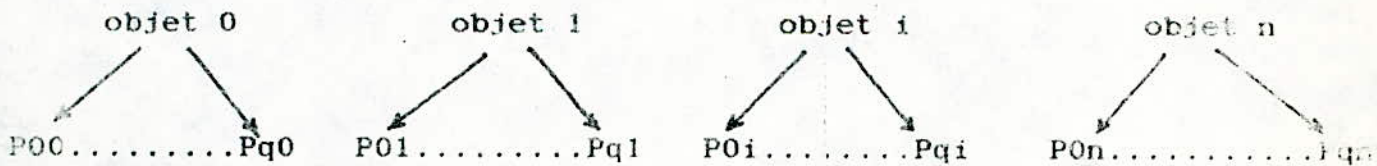
Plusieurs techniques ont ete developpees pour generer automatiquement les objets intermediaires , les approches les plus utilisees sont :

- Par interpolation des objets .
- Par concatenation des transformations geometriques de bases .

3.5.1.2 - METHODE ET CHOIX :

1ere Methode : Par interpolation des objets

Soient (n+1) objets-cle de (q+1) points chacun , la generation d'objets intermediaires entre eux revient a une serie d'interpolation entre les points du meme rang des objets .



L'interpolation entre objet 0 , objet 1 objet n revient a l'interpolation entre :

P00, P01, P0n
 P10, P11, P1n
 ⋮ ⋮ ⋮
 Pi0, Pi1, Pin
 ⋮ ⋮ ⋮
 Pq0, Pq1, Pqn

Associons a chaque objet i (objet i) un vecteur V_i de valeurs reelles pour chaque point les coordonnes (x, y, z) engendrer un objet intermediaire (objet k) revient a determiner un vecteur $V(k)$ en fonction de V_0, V_1, \dots, V_n sachant que :

Quelque soit k appartient a $\{0, 1, \dots, n\}$

$V(k) = V_k$; (Condition d'interpolation)

Avec $V(k) = C_0(k) \cdot V_0 + C_1(k) \cdot V_1 + \dots + C_n(k) \cdot V_n$ (1)

$C_i(k)$ est appele fonction d'influence associee au vecteur V_i :

elle exprime l'influence de V_i (objet i) sur l'objet intermediaire

(objet k) a generer . Elle doit verifier les conditions suivantes :

Quelque soit j appartenant a $\{0, 1, \dots, n\}$

$C_i(i) = 1$

$C_i(j) = 0 ; i \neq j ; C_i(k) = 1 ;$ quelque soit k appartenant a \mathbb{N}

2 eme Methode : Par transformation geometriques .

Cette methode permet la generation des objets intermediaires a l'aide des transformations geometriques .



* Soit M la matrice de transformation que l'on doit appliquer a l'objet cle O_1 au point P_1 pour obtenir l'objet cle O_2 au point P_2 .

* Soit X le nombre de points generes entre les deux points cle P_1 et P_2 . (ce nombre depend du pas de generation de la courbe qui depend a son tour de la vitesse de deplacement de l'objet entre les deux points P_1 et P_2) .

Chaque objet intermediaire O_i sera alors genere en appliquant a l'objet intermediaire precedent (on a l'objet cle , si c'est ce

premier objet intermediaire genere) la transformation M' , avec

$$\boxed{M' = M / X}$$

Cet objet intermediaire est implante a chaque fois au point intermediaire suivant . Il faudra donc rajouter a M' une translation dont le vecteur est calcule a partir des deux points intermediaires.

Pour passer d'un objet intermediaire O_i a l'objet intermediaire

O_{i+1} implantes respectivement aux points P_i et P_{i+1} on doit appliquer a l'objet O_i la translation T telle que :

$$\boxed{T = (M / X) + tr (P_i , P_{i+1})}$$

tr , est une translation $P_i \rightarrow P_{i+1}$.

Remarque : La matrice de transformation T n'est valable qu'entre deux points de
consecutifs car la matrice M change entre deux points de et X depend de la vitesse et peut aussi changer .

III.6 _ STRUCTURE LOGIQUE D'UN FILM :

La finalite de l'animation est bien entendu la production d'un film sur un support quelconque. Nous pouvons decomposer le film en deux parties :

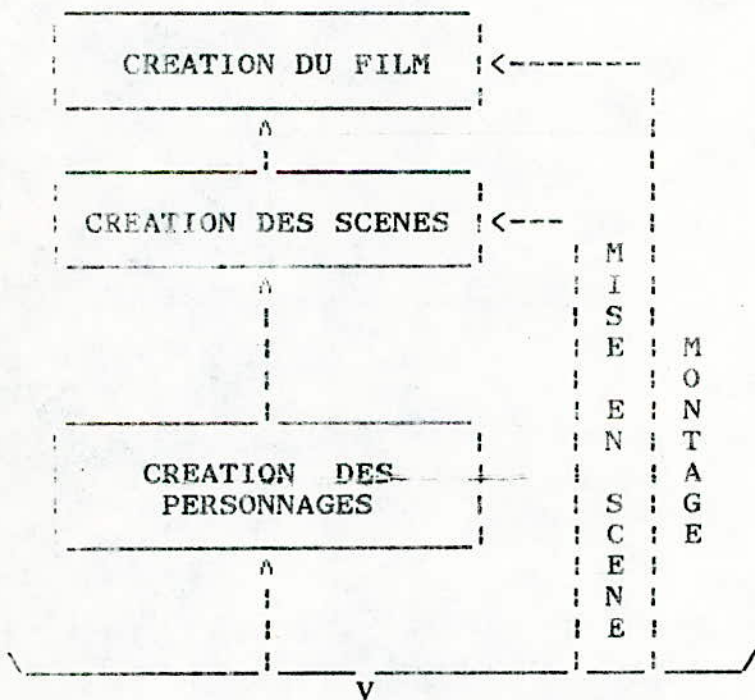
- _ LA DEFINITION DES DIFFERENTES SCENES UTILISEES.
- _ LA DEFINITION DE L'ENCHAINEMENT DE CELLES-CI (Montage).

La description de la scene se fait egalement en deux temps :

- _ DEFINITION DES PERSONNAGES PARTICIPANT A LA SCENE .
- _ DESCRIPTION DE LA MISE EN SCENE .

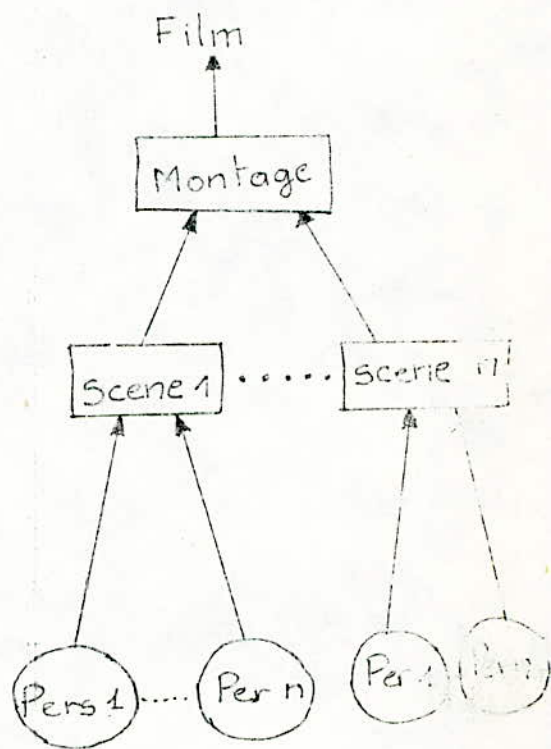
Ainsi la description logique d'un film a une structure arborescente de laquelle se deduisent ces trois principales parties constituant un systeme d'animation :(Voir fig 4.1 et fig 4.2)

- * CREATION DES PERSONNAGES
- * MISE EN SCENE
- * MONTAGE



STRUCTURE D'UN SYSTEME D'ANIMATION

Fig 4.2 .



STRUCTURE D'UN FILM

Fig 4.1 .

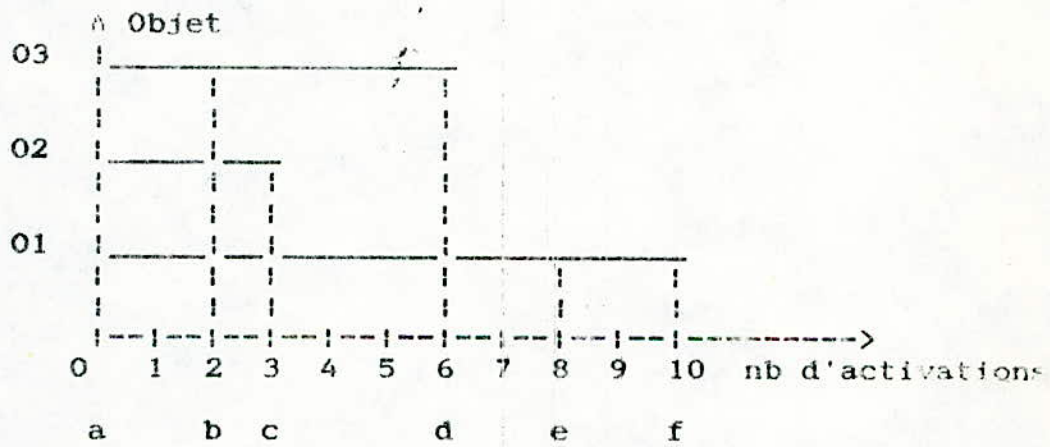
III.7 - CONSTITUTION DU SCENARIO :

Dans cette etape on constitue la scenario a partir des mouvements definis dans l'etape precedente et sauvegarde dans un support externe. On fera appel donc a chaque mouvement participant a la scene apres quoi , il y aura chargement des structures de donnees correspondantes et integration de celle-ci dans un environnement capable de les reconnaitre .

1 - SYNCRONISATION :

La synchronisation est la derniere etape avant l'animation effective . les differents mouvements etant charges , on devra donc specifier pour chacun de ces mouvement le moment de son activation et le nombre d'activations qu'il doit subir avant de s'arreter .

Exemple :



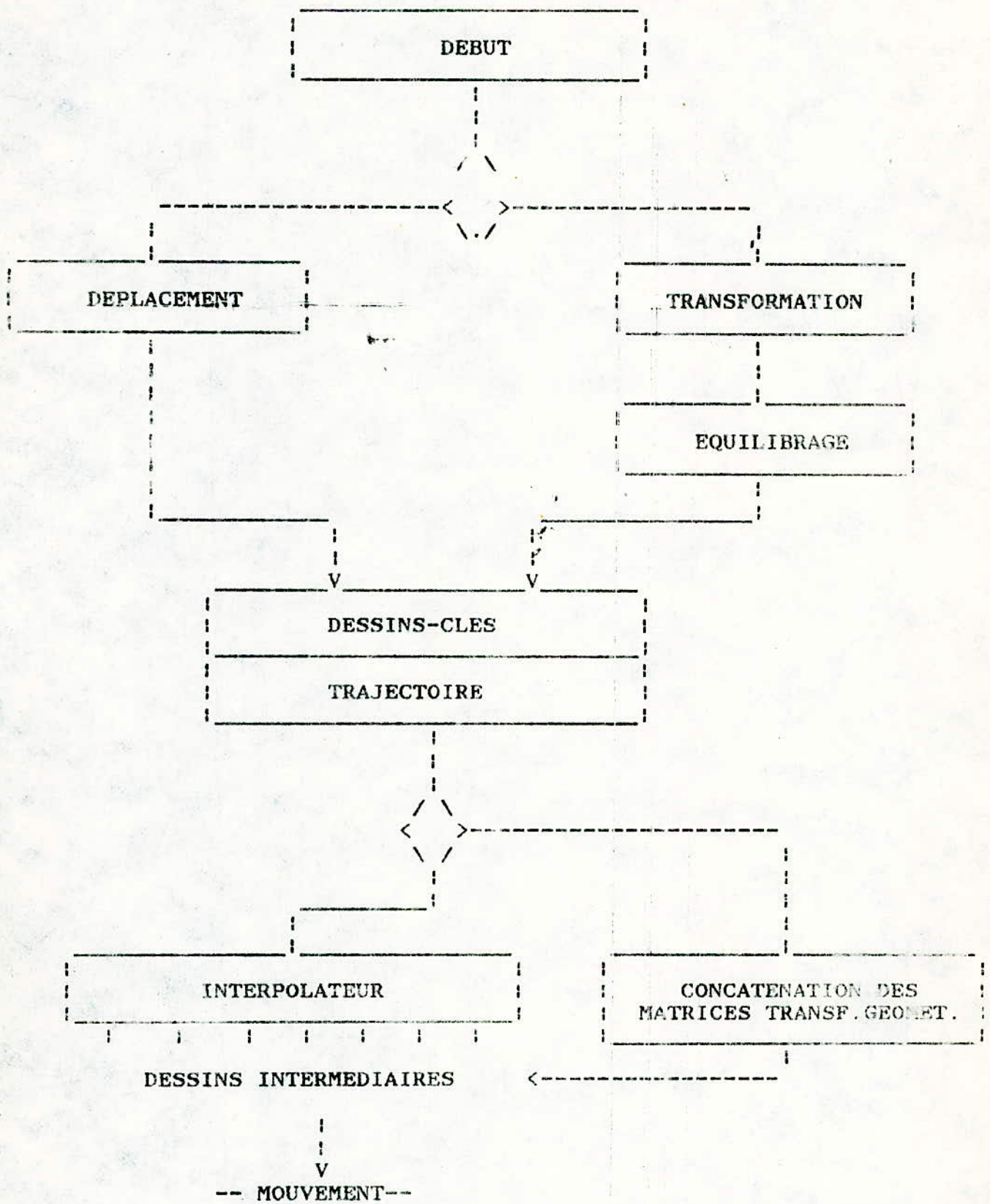
Interpretation :

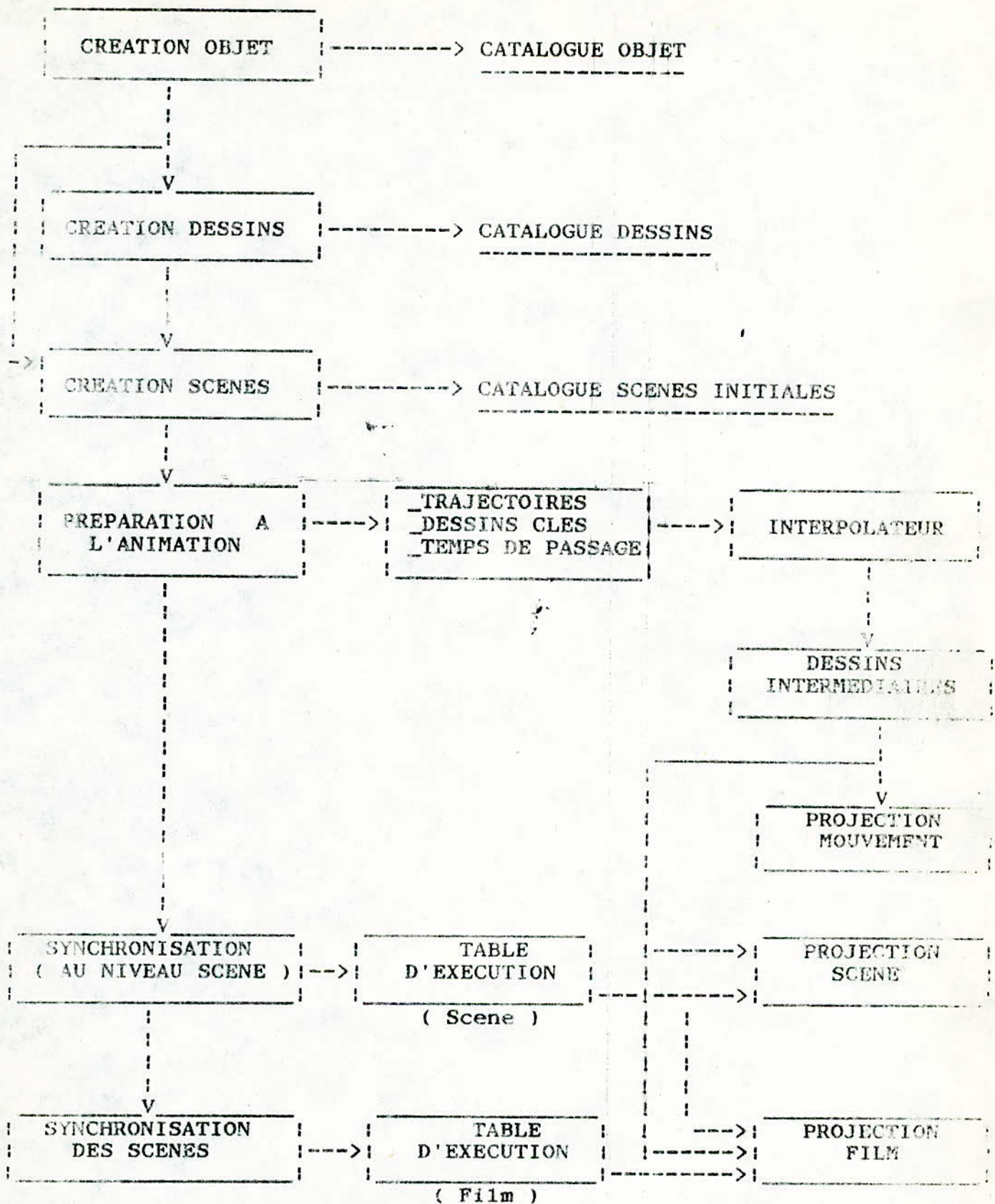
- a _ b : L'objet 02 démarre le premier ,pendant 2 activations .
- b _ c : 02 et 03 sont ensuite actives parallelement une seule fois .
- c _ d : 03 est continue ensuite pendant 4 activement .
- d _ e : Aucun objet n'est anime pendant cette activation (temps d'arret) .
- e _ f : L'objet 01 est anime pendant 2 activations

REMARQUE : Le meme principe est utilise pour la synchronisation des scenes pour la visualisation du " Scenario " .

SYNOPTIQUE GENERALE DU SCENARIO

SCHEMA GENERAL





SYNOPTIQUE GENERALE D'UN PROGRAMME DE MANIPULATION
ET D'ANIMATION D'OBJETS EN DEUX ET TROIS DIMENSIONS

RESULTATS
ET
INTERPRETATION

IV . Presentation des algorithmes et des resultats

ce chapitre est consacré à la description des algorithmes détaillés qui existe dans notre editeur

Ces algorithmes comportent essentiellement les Programmes de transformations .

les Programmes correspondant à quelques algorithmes seront donnés en annexe 3 (Programme de l'editeur .

On peut distinguer notamment :

- les rotations
- surfaces de Bezières
- et l'organigramme de l'editeur d'objet 3D

Le Programme de l'editeur objet 3D est donné en ANNEXE 3


```

80 * .....
85 *
90 * ALGORITHME POUR LA ROTATION TRI-DIMENSIONNELLE AUTOUR D'UN AXE
91 * QUELCONQUE DANS L'ESPACE
92 *
93 * .....
94 *
100 * ROUTINE DE PARAMETRES :P,X(),Y(),Z(),N1,N2,N3,T1
110 * p = nombre des triplets x,y,z
120 * x()= tableau contenant les coordonnees de x
130 * y()= tableau contenant les coordonnees de y
140 * z()= " " " " de z
150 * n1 = cosinus directeur de l'axe de rotation w.r.t. direction _ X
160 * n2 = " " " " " direction _ Y
170 * n3 = " " " " " direction _ Z
180 * t1 = angle de rotation en degres
181 *
190 DIM U(100,4),V(100,4) * 100 positions de vecteurs
200 MAT U=ZER(P,4) * initialisation
210 MAT V=ZER(P,4)
220 FOR I=1 TO P * construit des positions de
230 LET U(I,1)=X(I) * vecteurs homogenes
240 LET U(I,2)=Y(I)
250 LET U(I,3)=Z(I)
260 LET U(I,4)=1
270 NEXT I
280 MAT T=ZER(4,4) * redimensionne t rempli par des zeros
290 LET T2=T1/57.2957795# * convertit t1 en radians
300 LET T(4,4)=1 * creation de la matrice de transformation
310 LET T(1,1)=N1*N1+(1-N1*N1)*COS(T2)
320 LET T(1,2)=N1*N2*(1-COS(T2))+N3*SIN(T2)
330 LET T(1,3)=N1*N3*(1-COS(T2))-N2*SIN(T2)
340 LET T(2,1)=N1*N2*(1-COS(T2))-N3*SIN(T2)
350 LET T(2,2)=N2*N2+(1-N2*N2)*COS(T2)
360 LET T(2,3)=N2*N3*(1-COS(T2))+N1*SIN(T2)
370 LET T(3,1)=N1*N3*(1-COS(T2))+N2*SIN(T2)
380 LET T(3,2)=N2*N3*(1-COS(T2))-N1*SIN(T2)
390 LET T(3,3)=N3*N3+(1-N3*N3)*COS(T2)
400 MAT V=U*T * calcul des transformes de points
410 FOR I=1 TO P * calcul des coordonnees physiques
420 LET X(I)=V(I,1)
430 LET Y(I)=V(I,2)
440 LET Z(I)=V(I,3)
450 NEXT I
460 FIN ROUTINE

```



```

10 'Procédure de rotation en 3D sur l'axe des X
20 'Arguments:P,X(),Y(),Z(),t1
30 'X():tableaux contenant les coordonnées des X
40 'Y(): " " " " des Y
50 'Z(): " " " " des Z
60 'T1:angle de rotation en degré
70 DIM U(100,4),V(100,4)
80 MAT U=ZER(P,4)
90 MAT V=ZER(P,4)
100 FOR I=1 TO P
110 LET U(I,1)=X(I)
120 LET U(I,2)=Y(I)
130 LET U(I,3)=Z(I)
140 LET U(I,4)=1
150 NEXT I
160 LET T2=T1/57.2957795#
170 MAT T=ZER(4,4)
180 LET T(1,1)=T(3,3)=COS(T2)
190 LET T(3,1)=SIN(T2)
200 LET T(1,3)=-T(3,1)
210 LET T(2,2)=T(4,4)=1
220 MAT V=U*T
230 FOR I=1 TO P
240 LET X(I)=V(I,1)
250 LET Y(I)=V(I,2)
260 LET Z(I)=V(I,3)
270 NEXT I
280 FIN ROUTINE

```



```

10 *Procédure de rotation en 3D sur l'axe des Y
20 *Arguments:P,X(),Y(),Z(),t1
30 *X():tableaux contenant les coordonnées des X
40 *Y(): " " " " des Y
50 *Z(): " " " " des Z
60 *T1:angle de rotation en degré
70 DIM U(100,4),V(100,4)
80 MAT U=ZER(P,4)
90 MAT V=ZER(P,4)
100 FOR I=1 TO P
110 LET U(I,1)=X(I)
120 LET U(I,2)=Y(I)
130 LET U(I,3)=Z(I)
140 LET U(I,4)=1
150 NEXT I
160 LET T2=T1/57.2957795#
170 MAT T=ZER(4,4)
180 LET T(1,1)=T(2,2)=COS(T2)
190 LET T(1,2)=-SIN(T2)
200 LET T(2,1)=-T(1,2)
210 LET T(3,3)=T(4,4)=1
220 MAT V=U*T
230 FOR I=1 TO P
240 LET X(I)=V(I,1)
250 LET Y(I)=V(I,2)
260 LET Z(I)=V(I,3)
270 NEXT I
280 FIN DE ROUTINE

```

```

10 *Procédure de rotation en 3D sur l'axe des X
20 *Arguments:P,X(),Y(),Z(),t1
30 *X():tableaux contenant les coordonnées des X
40 *Y(): " " " " des Y
50 *Z(): " " " " des Z
60 *T1:angle de rotation en degré
70 DIM U(100,4),V(100,4)
80 MAT U=ZER(P,4)
90 MAT V=ZER(P,4)
100 FOR I=1 TO P
110 LET U(I,1)=X(I)
120 LET U(I,2)=Y(I)
130 LET U(I,3)=Z(I)
140 LET U(I,4)=1
150 NEXT I
160 LET T2=T1/57.2957795#
170 MAT T=ZER(4,4)
180 LET T(1,1)=T(3,3)=COS(T2)
190 LET T(3,1)=SIN(T2)
200 LET T(1,3)=-T(3,1)
210 LET T(2,2)=T(4,4)=1
220 MAT V=U*T
230 FOR I=1 TO P
240 LET X(I)=V(I,1)
250 LET Y(I)=V(I,2)
260 LET Z(I)=V(I,3)
270 NEXT I
280 FIN DE ROUTINE

```

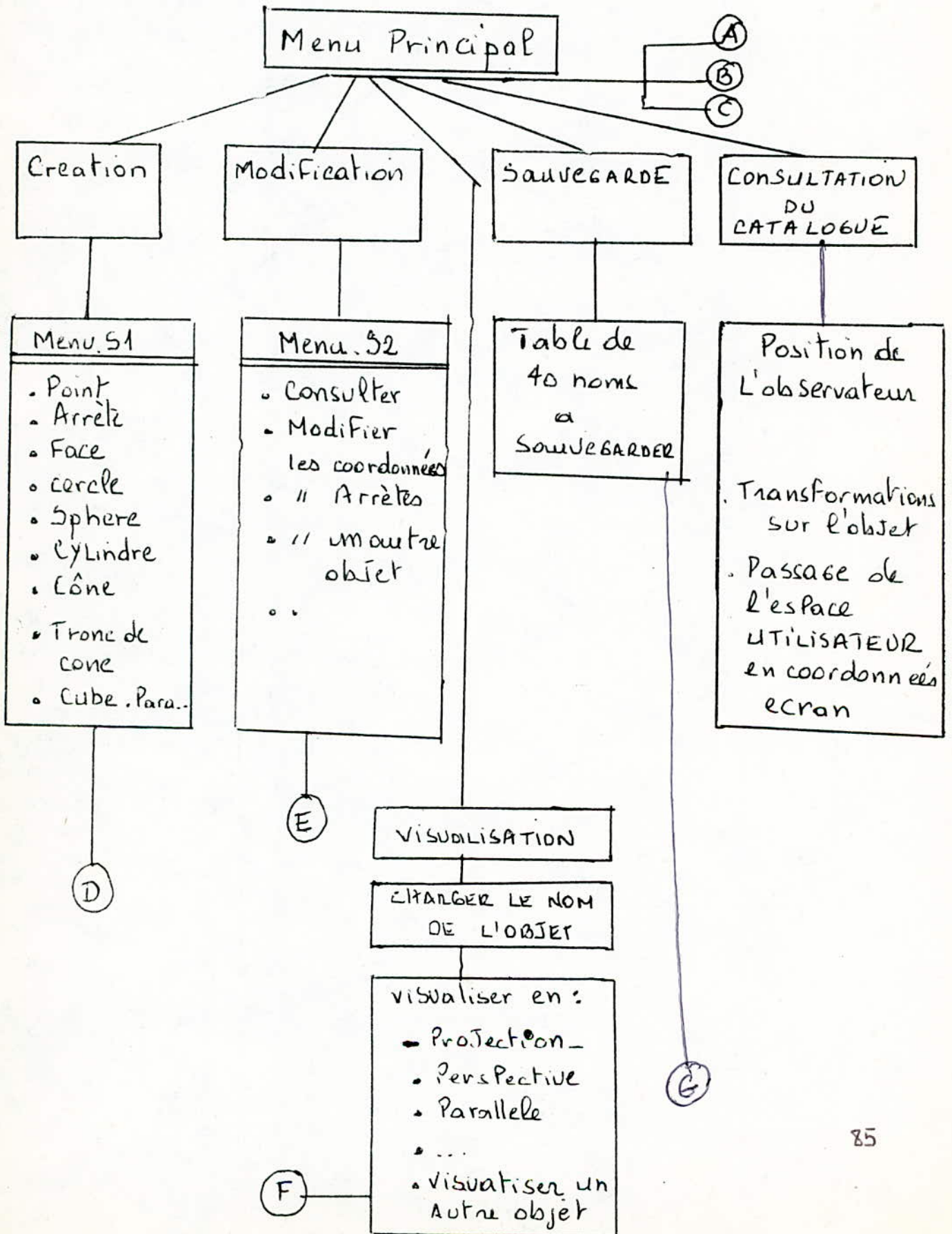


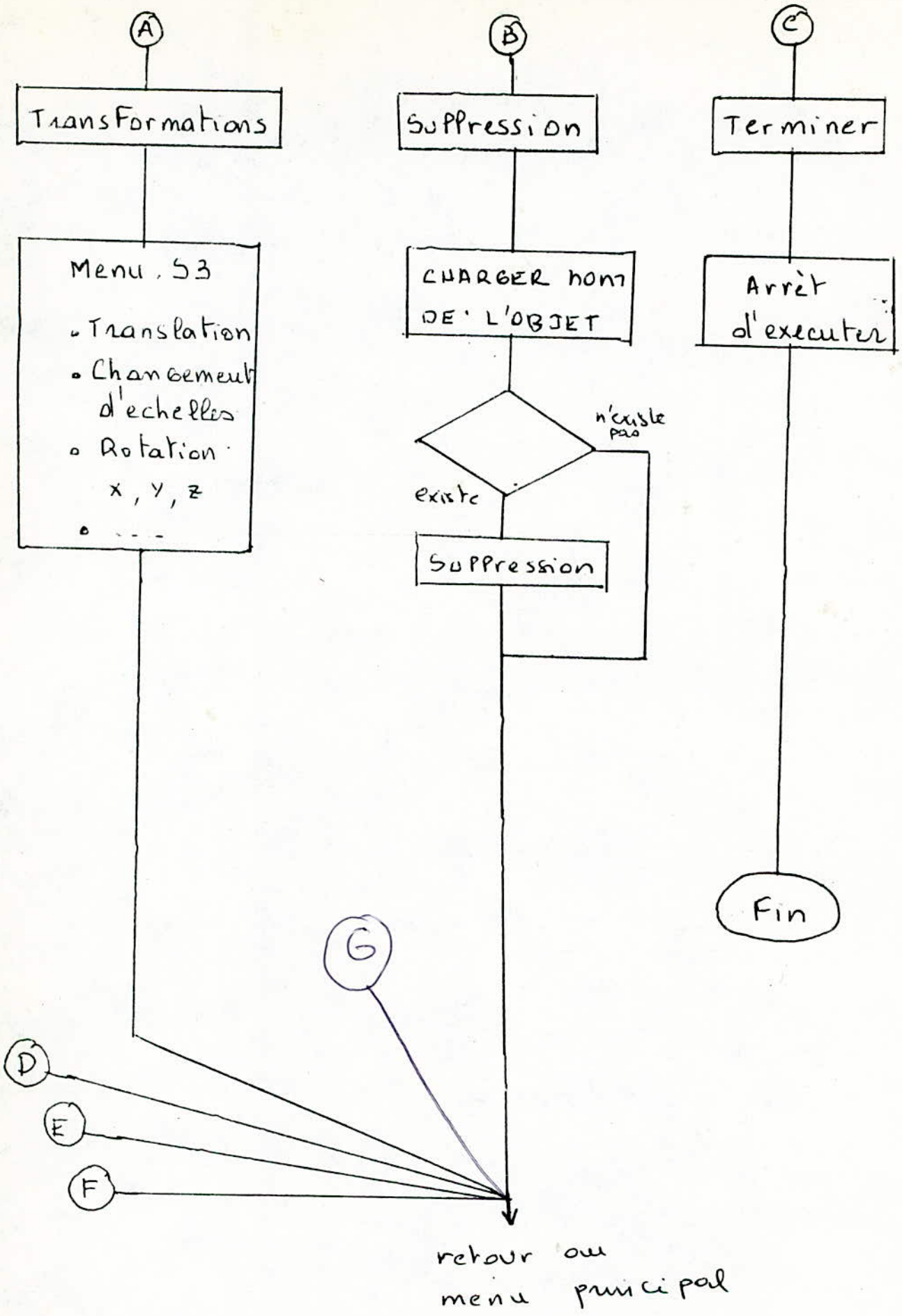
```

1 * *****
2 * GENERATION DE COURBES PAR LA METHODE D'INTERPOLATION DE BEZIER *
3 * *****
4 *
10 * ROUTINE DEPARAMETRES :N1,S,X(,),Z(,),F,R(.)
20 * N1=nombre de sommets dans le polygone bezier
30 * S=variable de controle 2=courbe plane, 3=courbe dans l'espace
40 * X(,1)=tableau contenant les composants de X pour les sommets du polygone
50 * Y(,1)=tableau contenant les composants de Y pour les sommets du polygone
60 * Z(,1)=tableau contenant les composants de Z pour les sommets du polygone
70 * P=le nombre de points le long de la courbe bezier
80 * R(,)=contenant les points le long de la courbe bezier
90 * R(1,)=les composants de X
100 * R(2,)=les composants de Y
110 * R(3,)=les composants de Z
120 * initialisation et dimension des matrices
130 * assume le maximum des 10 sommets du polygone
140 MAT J=ZER(1,N1)
150 MAT C=ZER(1,1)
160 MAT D=ZER(1,1)
170 MAT E=ZER(1,1)
180 LET N=N1-1
190 *definition d'une fonction pour l'evaluation de l'expansion binomiale
200 DEF FNF(X)
210 IF X=0 THEN 270
220 LET Y=1
230 LET Y=Y*X
240 LET X=X-1
250 IF X=0 THEN 290
260 GOTO 230
270 LET FNF=1
280 GOTO 300
290 LET FNF=Y
300 FNEND
310 LET K=1
320 FOR T=0 TO 1 STEP 1/(P-1) *genere les fonctions de base
330 FOR I=0 TO N
340 LET J(1,I+1)=(FNF(N)/(FNF(I)*FNF(N-I)))*T^I*(1-T)^(N-I)
350 NEXT I
360 MAT C=J*X *genere les points le long de A
370 MAT D=J*Y *courbe bezier 2-D ou 3-D
380 LET R(1,K)=C(1,1) *resultats du point cree
390 LET R(2,K)=D(1,1)
400 IF S=2 THEN 430
410 MAT E=J*Z
420 LET R(3,K)=E(1,1)
430 LET K=K+1
440 NEXT T
450 FIN ROUTINE

```


Principales fonctions
de
L'editeur Graphique (3D)





CONCLUSION

CONCLUSION

Mettre sur pied un systeme complet de generation et d'animation d'objets tridimensionnels n'est evidemment pas une affaire d'une journee , ni l'oeuvre d'une seule personne . Il s'agit la d'un travail purement pedagogique susceptible bien entendu de modifications et d'ameliorations , car ceci necessite une maitrise de l'infographie , une connaissance des techniques cinematographiques et enfin un materiel adequat . La recherche bibliographique a ete l'un de nos principaux soucis . Le lecteur trouvera dans ce memoire suffisamment de referances pouvant servir de base pour des taches ulterieures , sans perte de temps . Tout en etant q'une ebauche , notre etude permet l'ouverture d'une large voie de recherche sur l'animation et la vision artificielle au sein du departement d'electronique .

Au cours de cette etude , nous avons essaye d'appliquer certaines methodes mathematiques comme les transformations geometriques , interpolations et enfin certains mouvements sont obtenus a partir de dessins faits sur du papier graphique qui seront ensuite charges point par point (methode cinematographique) , celle ci necessite une importante occupation memoire .

La methode de transformations geometriques nous permet de controler le mouvement et l'accelerer ou de le ralentir entre chaque paire de points et tout en jouant sur le pas de generation des intermediaires ce qui permet d'attenuer l'influence du temps qui joue un role determinant dans la mise en oeuvre de l'animation .

Le probleme des surfaces cachees a ete resolu en en oeuvre l'importance de la distance du point d'observation par rapport a l'ecran de visualisation ainsi que la representation perspective avec "point de fuite" .

On peut d'ores et déjà entrevoir quelques extensions entre autres :

- Introduire les effets qu'engendre le mouvement de la camera : (zoom Panoramique , Travelling , etc ...)
- L'amélioration du réalisme de l'image (Ombrage , Transparence , etc ...)
- Evolution des acteurs dans des domaines dynamiques et leur attribuer des contraintes dans leur mouvement .

BIBLIOGRAPHIE

B I B L I O G R A P H I E

A- O U V R A G E S :

- 1- J . P . BLANCER .
Modeles d'expressions graphiques
P.S.I 1983
- 2- R . DONY .
Graphisme scientifique sur microordinateur de 2 et 3
dimension
Edition Eyrolles 1985
- 3- S . HARINGTON .
Computer graphics: a programming approach .
International Student Edition 1985
- 4- D . HEARN & M . PAULINE . BAKER
Graphisme sur votre microordinateur.
Intermicro 1984
- 5- F . MARTINEZ .
La synthese d'image:
Concepts, Materiels et Logiciels
Edition tests 1984.
- 6- P . MORVAN & M . LUCAS .
Image et Ordinateur:
Introduction a l'Infographie interactive.
Larousse 1976.
- 7- J . L . VULDY .
Graphisme sur votre microordinateur.
Edition Eyrolles 1985.
- 8- G . HEGRON .
Synthese d'images :
Algorithmes elementaires.
Edition Eyrolles 1986

9- D . F . ROGERS.

Procedural elements for computer graphics.

Edition Mc Graw Hill.

10- D . F . ROGERS et J . A . ADAMS.

Mathematical elements for computer graphics

Edition Mc Graw Hill Books Company

11- W . J . GORDON & R . F . REISENFELD .

Bernstein-Bezier Methods for the computer-aided design of free
form curves and surfaces .

J.ACM, vol. 21, pp 293-310, 1974

12- J . G . POSTAIRE .

De l'image a la decision:

Analyse des images numeriques et theorie de la decision.

Edition DUNOD informatique.

13- S . KIJNER . Tome 1 et 2

V.A.O : La vision assistee par ordinateur: le traitement d'images

Tome 1 : Echantillonnage , Codage , Restauration d'une image

Micro-Systemes-pp 114-135-Novembre 1983.

Tome 2 : Extraction de l'information et reconnaissance de formes

Micro-Systemes-pp 125-142 Decembre 1984.

14- La vision artificielle.

Micro-systemes- Janvier 1987-pp 69-82.

15- La vision artificielle :

Algorithme de traitement d'image et architectures.

Electronique application N 51-Aout-Septembre:pp27-31

16- Y . SMARA

Conception et realisation du systeme de traitement numerique
d'images A.R.T.I.S 84.

Traitement et visualisation d'images numeriques .

U.S.T.H.B Alger Juin 85

17- S . ABBED & A . MEZEREG .

Retrospective sur le traitement d'images .

E.N.P Alger Juin 86

18- A . BIJAOUI .

Image et information.

Introduction au traitement numerique des images.

Edition MASSON 1984

19- M . KUNT .

traitement numerique des signaux.

Edition DUNOD 1984

B- M E M O I R E S :

20- T . TAMAZOUZT & K . SIMOHAND .

Etude et realisation d'un logiciel graphique pour la manipulation

des dessins en deux dimensions ." S.I.G.M.A "

I.N.I 1986

21- R . OUADI

Animation assistee par ordinateur

RAPPORT DEA . TOULOUSE 1984

22- A . BENZOUAK & A . BOURELAM

Conception d'un logiciel graphique de creation et d'animation

d'objets 3D ."S.I.C.A 3D"

I.N.I 1987

23- BOUZEFRANE

Etude comparative d'algorithmes de traitement de surfaces gauches

elimination des parties cachees par subdivision .

Renne 1 1984

ANNEXE 1

contient le decor et P2 le dessin a animer; la voiture . Quand celle-ci se deplacera , on realisera les operations d'effacement et d'affichage sur P2 et le decor ne sera ^{pas} altere sur l'ecran .

2 - QUELQUES PROCEDE D'INTERPOLATION :

2.1 - Definition :

Etant donne une fonction definie discreteement sur un ensemble de points $X_i (i=0,1,\dots,n)$. L'operation d'interpolation consiste a calculer la valeur de la fonction pour une valeur X n'appartenant pas a l'ensemble de definition de la fonction (ensemble des X_i) .

- Interpolation de degre n : Formule de Lagrange :

On calcule un polynome $P_n(x)$ definis sur $(n+1)$ points (X_0, X_1, \dots, X_n) . La valeur interpolee , qui est la valeur prise par le polynome $P_n(x)$ pour $X = X'$, s'obtient par l'intermediaire de la formule de lagrange suivante :

$$P_n(X) = \sum_{i=0}^n L_i(X') \times Y_i$$

$$\text{Avec } L_i(X') = \frac{\prod_{\substack{j=0 \\ j \neq i}}^n (X_j - X')}{\prod_{\substack{l=0 \\ l \neq i}}^n (X_j - X_l)} ; \quad \begin{cases} L_i(X_j) = 0 ; i \neq j \\ L_i(X_i) = 1 \end{cases}$$

- Avantages : L'avantages du polynome de LAGRANGE est qu'il est exact sur les polynomes de degre n ; la courbe passe par tous les points et la possibilites du controle nombre des intermediaires.

- Inconvenient : Il ne tient pas compte des variations de courbure et des pente .

Remarque :

Ce procede offre un controle local . le changement de position d'un point n'influe pas sur le comportement de la courbe au dela du voisinage de ce point .

ALGORITHME D'INTERPOLATION

```
DEBUT
  J = 0
  TANT QUE J < N
    FAIRE
      APPEL GEN - INT(J)
      CAS DE J ;
        < 1 ; J = J + PAS1
        < 2 ; J = J + PAS2
        < 3 ; J = J + PAS3
        .
        .
        .
        < N ; J = J + PASN
      FIN CAS
    FIN FAIRE
  FIN .

* PROCEDURE GEN() - INT(J) *
  DEBUT
    POUR R = 0 JUSQU'A Q
      FAIRE
        X' = 0 ; Y' = 0 ; Z' = 0
        POUR I = 0 JUSQU'A N
          FAIRE
            CALCUL DE L(I,J)
            X' = X' + L(I,J) x X(I,K)
            Y' = Y' + L(I,J) x Y(I,K)
            Z' = Z' + L(I,J) x Z(I,K)
          FIN FAIRE
        FIN FAIRE
      FIN
  FIN
```

3- COURBE DE BEZIER :

Pour definir la courbe de bezier ; on definit d'abord un polynome
Ce dernier est defini par (n + 1) points P_i .

3.1- Definition mathematique :

Les P_i sont les $(n + 1)$ points de la courbe $n =$ ordre de la courbe ; $(n - 1) =$ degre de la courbe .

$$P(t) = \sum_{i=0}^n P_i \times B_{i,n}(t)$$

ou ;

$$B_{i,n}(t) = C_n^i t^i (1-t)^{n-i}$$

avec ;

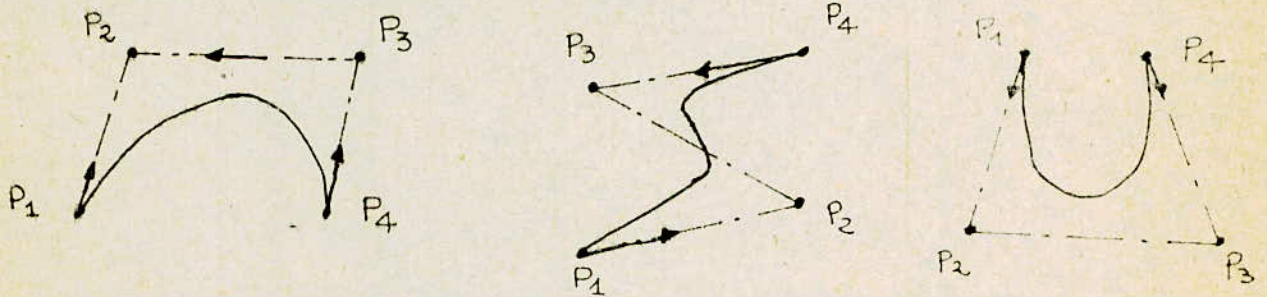
$$C_n^i = \frac{n!}{i! (n-i)!}$$

et

$$P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} ; P(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}$$

En faisant varier t entre les valeur 0 et 1 d'un increment faible , calcule avec cette modelisation la courbe approchant le polynome .

Exemple de configuration avec 4 points :



Les extremités $P_1 . P_4$ appartiennent a la courbe et les vecteurs $P_1 P_2$ et $P_3 P_4$ sont tangents a la courbe , cette remarque est valable pour $(n + 1)$ points .

Ces courbes peuvent etre ecrites sous la forme matricielle suivante

$$P(t) \longrightarrow C(t) = \begin{bmatrix} (1-t)^3 & 3.t.(1-t)^2 & 3.t^2.(1-t) & t^3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

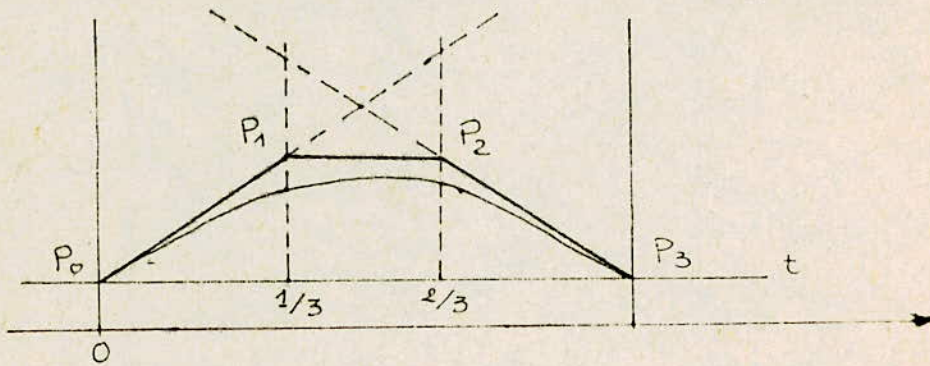
$$= (t^3, t^2, t, 1) \cdot MB$$

- P0
- P1
- P2
- P3

Avec MB =

$$\begin{vmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}$$

Avec $t \in [0, 1]$, si $t = 0$ la courbe est sur P0 si $t = 1$ la courbe est sur P3. P1 et P2 ont le role de vecteurs tangents.



- Tangente en P0 et P3 .
- La longueur du vecteur tangent en P0 est egale a $3 \times (P1 - P0)$.
- La longueur du vecteur tangent en P3 est egale a $3 \times (P3 - P2)$.

Passage des equations matricielles a la methode de la courbe en 3D .

- Chaque point P_i est defini par ses coordonnees (X , Y , Z) la multiplication matricielle de l'equation precedente donne une equation polynomiale pour chacune des coordonnees, par exemple, les coordonnee X de la courbe de Bezier sont donnees par :

$$X(t) = (-t^3 + 3t^2 - 3t + 1) P_0^x + (3t^3 - 6t^2 + 3t) P_1^x + (-3t^3 + 3t^2) P_2^x + t^3 P_3^x$$

N B : La notation P_i^x represente la coordonnee X point P_i

REMARQUE :

Les equations pour les coordonnees Y et Z sont les memes que pour Y

sauf que P_i^x est substituée par P_i^y et P_i^z . Le nombre de points de contrôle peut-être supérieur à quatre mais ceci exige un calcul de fonctions polynomiales à degrés élevés, ce qui demande un temps de calcul considérable.

4 - LES SURFACES :

Nous présenterons maintenant la même approche vue précédemment dans le cas des courbes et nous verrons leur adaptation pour la génération des surfaces gauches.

4.1 - LES SURFACES DE BEZIER :

Une surface de Bézier est définie par :

$$Q(u,v) = \sum_{i=0}^n \sum_{j=0}^m P_{i+1,j+1} J_{n,i}(u) K_{m,j}(v)$$

sont les polynômes de Bernstein et m, n les nombres de sommets respectivement suivant U et V . Les calculs sont plus faciles si $m < 4$ et $n < 4$.

REMARQUE : Pour m et n égaux à 4, une surface sera définie par 16 points de contrôle sous la forme matricielle.

$$Q(U,V) = \begin{bmatrix} (1-U)^3 & 3U(1-U)^2 & 3U^2(1-U) & U^3 \end{bmatrix} \cdot B \cdot \begin{bmatrix} (1-V) \\ (1-V) \cdot 3V \\ (1-V) \cdot 3V \\ V \end{bmatrix}$$

Avec $B = \begin{bmatrix} P_{11} & \dots & P_{14} \\ \vdots & & \vdots \\ P_{41} & \dots & P_{44} \end{bmatrix}$

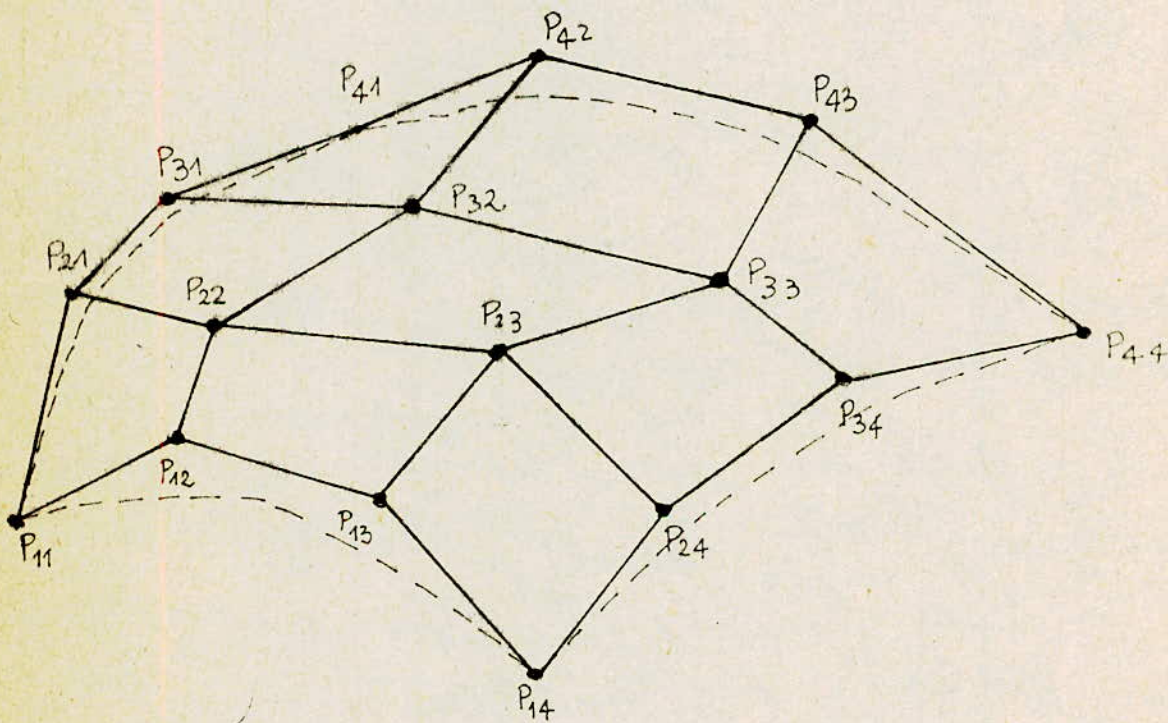
Ou bien

$$Q(U,V) = \begin{bmatrix} U^3 & U^2 & U & 1 \end{bmatrix} \cdot M_b \cdot B \cdot M_b \begin{bmatrix} 3 \\ V \\ 2 \\ V \\ V \\ 1 \end{bmatrix}$$

CONCATENATION : Pour joindre deux surfaces de Bezier , il suffit de faire coïncider les quatres points de controle d'un cote de l'une avec les quatres points de controle d'un cote de l'autre surface .

Les surfaces de Bezier sont compose des positions des vecteurs qui definissent les points du polygone (fig)

$$\text{si } B = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix}$$



ANNEXE 2

- ALGORITHME DE SUTHERLAND-HOGMAN :

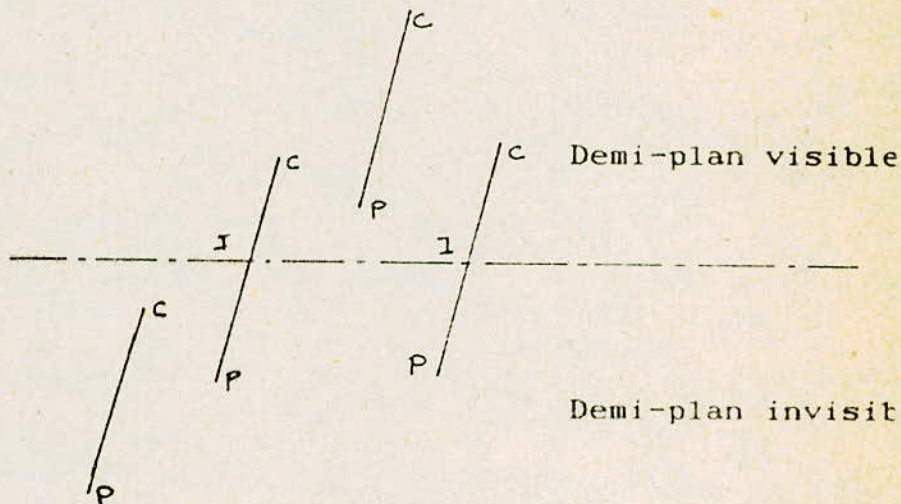
On parcourt le polygone arete par arete consecutivement . L'etude d chacune d'elle est faite par rapport a un bord de la fenetre .La droi supportant le bord de la fenetre fait apparaitre un demi-plan visible et un autre invisible ; la figure suivante fait ressortir les quatres

cas possible :

P : Precedent

C : Courant

I : Intersection



CAS 1 : Deux sommets sont dans la region visible ; on recupere le poi courant (C est ajoute a la liste de sortie)

CAS 2 : Le point sort de la region visible ; on ajoute I a la liste d sortie .

CAS 3 : Les points P et C sont hors de la region visible (on passe a une autre etape) la liste de sortie reste inchangee .

CAS 4 : Le point C entre dans la region visible ; on ajoute les point I et C a notre collection .

Dans tous les cas , on affecte a C sa valeur de visibilite (visible non) . La liste de sortie -si non vide- va etre reetudie par rapport au bord suivant de la fenetre jusqu'a parcourir entierement cette deriere , ou bien aboutir a une liste de sortie vide . Notons que lors d passage d'une arete a une autre le C devient P et l'autre extremite devient C .

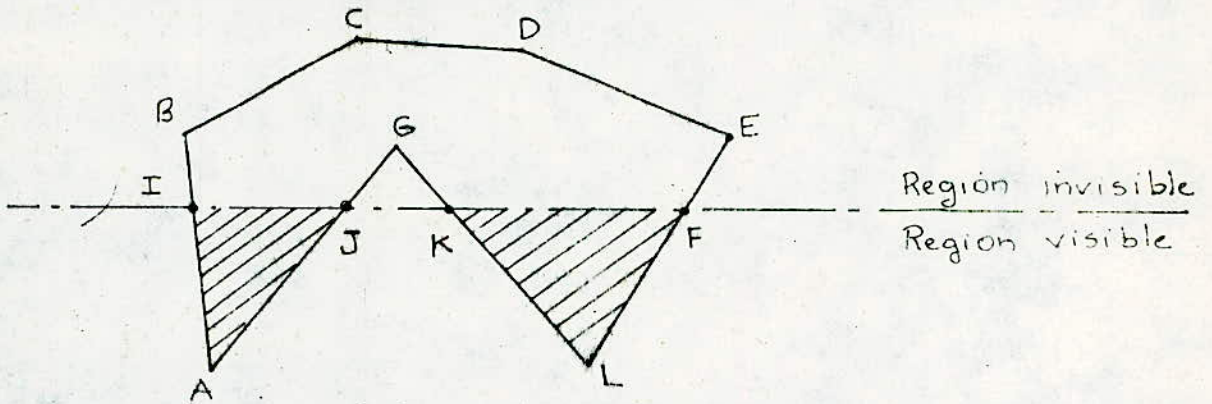
AVANTAGES : * La fenetre n'est pas forcement rectangulaire .

* On aboutit directement a un polygone .

INCONVENIENT :

* Erreur de decoupage dans le cas de polygones non conve

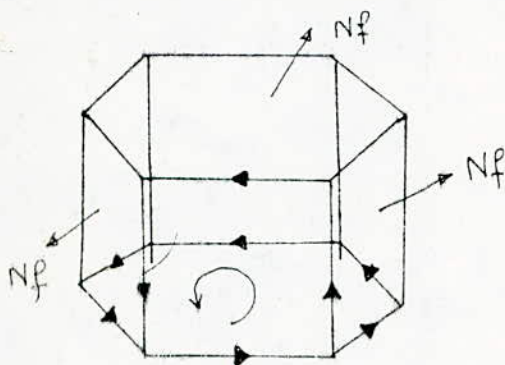
(voir figure ci-dessous)



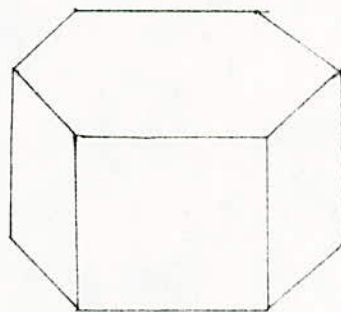
Le resultat restitué par l'algorithme de coupage contient un element parasite . Ainsi au lieu d'avoir deux polygone A.I.J et K.F.L , on au un seul polygone A.J.K.F.L.I

ALGORITHME DE GALIMBERTI ET MONTANARI :

Le present algorithme traite des polyedres tel que chacune de leur aretes appartient a deux faces et deux seulement ; la contrainte de convexite des faces ne se passe pas , cependant celles-ci doivent etre orientees de maniere a ce qu'une arete soit parcourue dans un sens po une face et dans l'autre sens pour la seconde .



Polyedre a etudier



Premiere phase et fin

Cette orientation nous permet egalement de calculer la normale a la face dirigee vers l'extremite du polyedre

$$\vec{N}_f = 2 \cdot U \cdot |A(F)| = \sum_{i=2}^{i=P-1} \vec{S}_i \cdot \vec{S}_{i+1} \cdot \vec{S}_1 \cdot \vec{S}_{i+1}$$

A : Aire de la face f .

U : Vecteur unitaire oriente vers l'exterieur du polyedre .

P : Degre de la face (le degre d'une face est egal au nombre de cotes du polygone formant cette face)

S_1 : Premier sommet de la face f et S_p le dernier .

Comme nous l'avons deja precise , pour cet algorithme ses traitements dans l'espace objet et procede comme suit :

La derniere phase consiste a reduire le nombre d'element a etudier en eliminant toute arete appartenant a deux faces cachees . Une face est cachee , lorsque sa normale a le meme sens que la direction de visee . Elle est susceptible d'etre visible quand sa normale est tournee vers

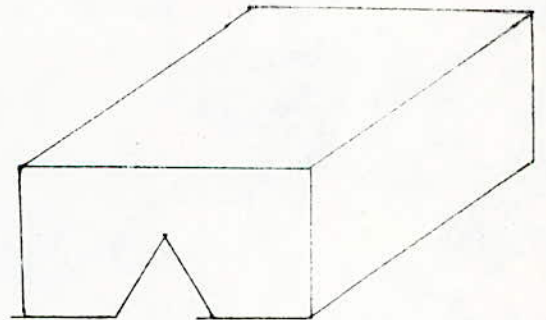
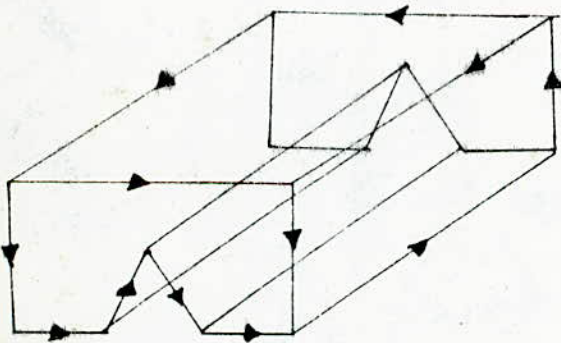
l'observateur .

Notons que lorsque la scene est constituee de polyedres convexes l'etude s'acheve a ce niveau .

L'idee de base de la seconde etape consiste a etudier point par point les aretes restantes par-rapport a toutes les faces susceptible d'etre vues . On dira dans ce cas qu'un point M est cache par la face f , s' verifie l'une des conditions qui vont suivre , il sera dit visible da le cas contraire .

- Le segment joignant le point M au point d'observateur O , perce le plan de face F .
- La projection du point M appartient a l'interieur de la projection de la face F sur le plan de vue .

Cet algorithme est d'une grande precision , il s'adapte essentielleme aux dessins avec traits , il est tres couteux en temps de traitement



Objet non convexe traite par la methode

de Galimberti-montanari

ANNEXE 3

```

1      '#####'
2      '#
3      '#
4      '#          EDITEUR GRAPHIQUE
5      '#
6      '#          3-DIMENSSION
7      '# REALISE PAR : HACENE ABDELRRANI
8      '#          &
9      '#          BENSALAM FERHAT
10     '#
13     '#          ANNEE 88/89
15     '#
16     '#####'
17     '#####'
20     '#
20     '#          programme principale
20     '#
20     '#####'
20 DIM TPT(150,3),TBA(200),OBJ$(40)
20 REM -----initialisation-----
20 CR=0:MPT=150:MOB=40:OX=255:OY=105:DDD=0
20 REM -----chargement de la table tabobj-----
20 ZZZ=0
20 OPEN "fcat" FOR INPUT AS #1
20 IF EOF(1) THEN CLOSE:GOTO 130
20 ZZZ=ZZZ+1:INPUT #1,OBJ$(ZZZ)
20 GOTO 100
20 SCREEN 0
20 WIDTH 80:CLS:GOSUB 820
20 LOCATE 25,3:PRINT "VISUALISATION DES OBJETS EN 3D"
20 LOCATE 25,4:PRINT "
20 LOCATE 8,6:PRINT "          1:CREER"
20 LOCATE 8,7:PRINT "          2:MODIFIER"
20 LOCATE 8,8:PRINT "          3:SAUVEGARDER"
20 LOCATE 8,9:PRINT "          4:VISUALISER"
20 LOCATE 8,10:PRINT "          5:CONSULTER LE CATALOGUE"
20 LOCATE 8,11:PRINT "          6:TRANSFORMATION"
20 LOCATE 8,12:PRINT "          7:SUPP UN OBJET"
20 LOCATE 8,13:PRINT "          8:TERMINER"
20 LOCATE 8,16:PRINT "          VOTRE CHOIX EST : "
20 A$=INKEY$:IF LEN(A$)=0 THEN GOTO 270
20 IF A$="1" THEN GOSUB 440:GOTO 140
20 IF A$="2" THEN GOSUB 2690:GOTO 140
20 IF A$="3" THEN GOSUB 1260:GOTO 140
20 IF A$="4" THEN GOSUB 2690:GOTO 140
20 IF A$="5" THEN GOSUB 1660:GOTO 140
20 IF A$="6" THEN GOSUB 1710:GOTO 140
20 IF A$="7" THEN GOSUB 2550:GOTO 140
20 IF A$="8" THEN 380
20 GOTO 270
20 CLS:PRINT "merci de votre attention"
20 GOSUB 2650:WIDTH 80:CLOSE
20 '#####'
20 '#
20 '#          fin de programme principale
20 '#####'
20 END
20 '#####'
20 '#
20 '#          sous-prog-creation
20 '#####'
20 CLS:GOSUB 820

```



```

480 LOCATE 8,3:PRINT "VOUS POUVEZ CREER UN(E) "
490 LOCATE 8,5:PRINT "          1:POINT"
500 LOCATE 8,6:PRINT "          2:ARETE"
510 LOCATE 8,7:PRINT "          3:FACE"
520 LOCATE 8,8:PRINT "          4:CERCLE"
530 LOCATE 8,9:PRINT "          5:SPHERE"
540 LOCATE 8,10:PRINT "         6:CYLINDRE"
550 LOCATE 8,11:PRINT "         7:CONC"
560 LOCATE 8,12:PRINT "         8:TRONC DE CONE"
565 LOCATE 8,13:PRINT "         9:CUBE"
566 LOCATE 8,14:PRINT "        10:PARALLEPIPEDE"
570 LOCATE 8,15:PRINT "        11:OBJET COMPLEXE"
580 LOCATE 8,16:PRINT "        12:OBJET A SURFACE DE REVOLUTION"
581 LOCATE 8,17:PRINT "        13:RETOURNER AU MENU "
590 LOCATE 8,19:INPUT "          VOTRE CHOIX EST : ";N2
600 IF N2<1 OR N2>13 THEN GOTO 470
610 IF N2=13 THEN 640
620 ON N2 GOSUB 860,920,1110,1910,2030,2140,2270,2400,4000,4100,1190,10000
630 DDD=1
640 RETURN
650          '#####
660          '###      sous-prog-rec-nom      ###
670          '#####
680 INPUT "donner le nom de 1 objet":NOM$
690 RETURN
700          '#####
710          '###      sous-prog-rec-point    ###
720          '#####
730 INPUT "donnez le nombre de point":NP
740 IF NP<0 OR NP>MPT THEN PRINT "erruer":GOTO 730
750 PRINT "introduire les pts un par un "
760 I=1
770 IF I>NP THEN 810
780 PRINT "point ";I;
790 INPUT "  x,y,z  "; TPT(I,1),TPT(I,2),TPT(I,3)
800 I=I+1:GOTO 770
810 RETURN
820 PRINT "#####
#####"
840 LOCATE 1,20:PRINT "#####
#####"
850 RETURN
860          '#####
870          '###      sous-prog-creation-point  ###
880          '#####
890 CLS:CR=1:TYP=1:GOSUB 650:GOSUB 700:NBF=0
900 IF NP=0 THEN CR=0:PRINT "erruer"
910 RETURN
920          '#####
930          '###      sous-prog-creation-arete  ###
940          '#####
950 CLS:CR=1:TYP=2:GOSUB 650:GOSUB 700
960 IF NP=1 OR NP=0 THEN PRINT "erruer":CR=0:GOTO 1010
970 IF NP<>2 THEN GOTO 1000
980 NBF=1:TBA(1)=1:TBA(2)=2:TBA(3)=-1:TBA(4)=-2
990 GOTO 1010
1000 NBF=1:VAR=1:GOSUB 1020
1010 RETURN
1020          '#####
1030          '###      sous-prog-rec-ensemble-aretes  ###
1040          '#####

```



```

1050 VAR=1
1060 PRINT "donnez les ensembles d aretes separees par (-1) a la fin tapez (
1070 INPUT P:IF (P<>-2) AND (P<>-1) AND (P<1 OR P>NP) THEN PRINT "erruer":G
070
1080 TBA(VAR)=P:VAR=VAR+1
1090 IF P<>-2 THEN 1070
1100 RETURN
1110      '#####
1120      '###      sous-prog-creation-face      ###
1130      '#####
1140 CLS:CR=1:TYP=3:GOSUB 650:GOSUB 700
1150 IF NP<3 THEN CR=0:PRINT "erruer":GOTO 1180
1160 NBF=1:VAR=1:GOSUB 1020
1170 IF TBA(1) <> TBA(VAR-2) THEN TBA(VAR-1)=TBA(1):TBA(VAR)=-1:TBA(VAR+1)=
1180 RETURN
1190      '#####
1200      '###      sous-prog-creation-objet complexe      ###
1210      '#####
1220 CLS:CR=1:TYP=11:GOSUB 650:GOSUB 700:NF=0
1230 IF NP<4 THEN PRINT "erruer":CR=0:GOTO 1250
1240 GOSUB 1020:NBF=1
1250 RETURN
1260      '##### ss-prog-save #####
1270 CLS
1280 IF NOM$="" THEN PRINT"pas d objet a sauvegarder":GOTO 1450
1290 I=1
1300 IF I>ZZZ THEN 1320
1310 IF OBJ$(I)=NOM$ THEN CR=0:GOTO 1340 ELSE I=I+1:GOTO 1300
1320 IF ZZZ=MOB THEN PRINT "sauvegard impossible(la table est saturee)":GOT
1330 CR=0:ZZZ=ZZZ+1:OBJ$(ZZZ)=NOM$:I=ZZZ
1340 OPEN OBJ$(I) FOR OUTPUT AS #1
1345 PRINT "sauvegard de 1 objet ...":PRINT:PRINT"          PATIENTEZ UN INSTAN
VOUS PLAIT"
1350 PRINT #1,NOM$:PRINT #1,TYP:PRINT #1,NP
1360 I=1
1370 IF I<= NP THEN PRINT #1,TPT(I,1),TPT(I,2),TPT(I,3):I=I+1:GOTO 1370
1380 PRINT #1,NBF
1390 IF NBF=0 THEN 1440
1400 I=1
1410 PRINT #1,TBA(I)
1420 IF TBA(I)=-2 THEN 1440
1430 I=I+1:GOTO 1410
1440 PRINT "objet sauvegarder ,tapez une touche ":CLOSE
1450 IF INKEY$="" THEN 1450
1460 RETURN
1470 REM ##### ss-pg-load #####
1480 CLS:INPUT"donnez le nom de 1 objet a transformer":NOM$
1490 I=1
1500 IF I>ZZZ THEN 1530
1510 IF OBJ$(I)=NOM$ THEN 1540 ELSE I=I+1:GOTO 1500
1520 A$=INKEY$:IF A$="" THEN 1520
1530 PRINT "objet inexistant":GOTO 1640
1540 PRINT "chargement de 1 objet ...":PRINT:PRINT"PATIENTEZ UN INSTANT S I
PLAIT"
1550 CR=1:DDD=1:OPEN NOM$ FOR INPUT AS #1
1560 INPUT #1,NOM$:INPUT #1,TYP:INPUT #1,NP
1570 I=1
1580 IF I<= NP THEN INPUT #1,TPT(I,1),TPT(I,2),TPT(I,3):I=I+1:GOTO 1580
1590 INPUT #1,NBF
1600 IF NBF=0 THEN CLOSE:GOTO 1630
1610 I=0
1620 IF EOF(1) THEN CLOSE ELSE I=I+1:INPUT #1,TBA(I):GOTO 1620
1630 IF A$="6" THEN 1650 ELSE PRINT "objet charge (tapez une touche)"

```



```

1640 IF INKEY$="" THEN 1640
1650 RETURN
1660 '##### ss-prog-catalogue #####
1670 CLS:I=1
1680 IF I>ZZZ THEN 1690 ELSE PRINT OBJ$(I):I=I+1:GOTO 1680
1690 IF INKEY$="" THEN 1690
1700 RETURN
1710 '##### ss-prog-transformation #####
1715 GOSUB 1470
1720 CLS:IF DDD=0 THEN 1900
1730 GOSUB 820
1740 LOCATE 20,3:PRINT "vous pouvez faire"
1750 LOCATE 8,6:PRINT "          1:TRANSLATION"
1760 LOCATE 8,7:PRINT "          2:CHANGEMENT D ECHELLE"
1770 LOCATE 8,8:PRINT "          3:ROTATION AUTOUR DE L AXE Z"
1780 LOCATE 8,9:PRINT "          4:ROTATION AUTOUR DE L AXE Y"
1790 LOCATE 8,10:PRINT "          5:ROTATION AUTOUR DE L AXE X"
1800 LOCATE 8,11:PRINT "          6:RETOURNER AU MENU"
1810 LOCATE 8,13:PRINT "          VOTRE CHOIX : "
1820 A$=INKEY$:IF A$="" THEN 1820
1830 IF A$="1" THEN GOSUB 2770:GOTO 1720
1840 IF A$="2" THEN GOSUB 2870:GOTO 1720
1850 IF A$="3" THEN GOSUB 3000:GOTO 1720
1860 IF A$="4" THEN GOSUB 3140:GOTO 1720
1870 IF A$="5" THEN GOSUB 3280:GOTO 1720
1880 IF A$="6" THEN 1900
1890 GOTO 1820
1900 RETURN
1910 '### ss-pg-cr-cercle ### '
1920 CLS:CR=1:TYP=4:GOSUB 650
1940 NP=0
1970 NP=NP+1:INPUT "donnez le centre du cercle":TPT(NP,1),TPT(NP,2),TPT(NP,3)
1980 NP=NP+1:INPUT "donnez le rayon du cercle":TPT(NP,1)
1990 NP=NP+1:INPUT "donnez la normale au cercle":TPT(NP,1),TPT(NP,2),TPT(NP,3)
2020 RETURN
2030 '### ss-pg-cr-sphere ### '
2040 CLS:CR=1:TYP=5:GOSUB 650
2060 NP=0
2090 NP=NP+1:INPUT "donnez le centre de la sphere":TPT(NP,1),TPT(NP,2),TPT(NP,3)
2100 NP=NP+1:INPUT "donnez le rayon de la sphere":TPT(NP,1)
2130 RETURN
2140 '### ss-pg-cr-cylindre ### '
2150 CLS:CR=1:TYP=6:GOSUB 650
2170 NP=0
2200 NP=NP+1:INPUT "donnez le centre":TPT(NP,1),TPT(NP,2),TPT(NP,3)
2210 NP=NP+1:INPUT "donnez le rayon":TPT(NP,1)
2220 NP=NP+1:INPUT "donnez la normale":TPT(NP,1),TPT(NP,2),TPT(NP,3)
2230 NP=NP+1:INPUT "donnez la hauteur":TPT(NP,1)
2260 RETURN
2270 '### ss-pg-cr-cone ### '
2280 CLS:CR=1:TYP=7:GOSUB 650
2300 NP=0
2330 NP=NP+1:INPUT "donnez le centre":TPT(NP,1),TPT(NP,2),TPT(NP,3)
2340 NP=NP+1:INPUT "donnez le rayon":TPT(NP,1)
2350 NP=NP+1:INPUT "donnez la normale":TPT(NP,1),TPT(NP,2),TPT(NP,3)
2360 NP=NP+1:INPUT "donnez le point":TPT(NP,1),TPT(NP,2),TPT(NP,3)
2390 RETURN
2400 '### ss-pg-cr-tronc-cone ### '
2410 CLS:CR=1:TYP=8:GOSUB 650
2430 NP=0

```



```

0 NP=NP+1:INPUT "donnez le centre ";TPT(NP,1),TPT(NP,2),TPT(NP,3)
0 NP=NP+1:INPUT "donnez le rayon1";TPT(NP,1)
0 NP=NP+1:INPUT "donnez la normale1";TPT(NP,1),TPT(NP,2),TPT(NP,3)
0 NP=NP+1:INPUT "donnez le centre2";TPT(NP,1),TPT(NP,2),TPT(NP,3)
0 NP=NP+1:INPUT "donnez le rayon2 ";TPT(NP,1)
0 NP=NP+1:INPUT "donnez la normale2";TPT(NP,1),TPT(NP,2),TPT(NP,3)
0 RETURN
0 '### ss-pg-sup ###'
0 CLS:INPUT"donnez le nom de l objet a sup";N$
0 I=1
0 IF I>ZZZ THEN PRINT "objet innexistant":GOTO 2630
0 IF OBJ$(I) <> N$ THEN I=I+1:GOTO 2580
0 IF I=ZZZ THEN 2620
0 OBJ$(I)=OBJ$(I+1):I=I+1:GOTO 2600
0 ZZZ=ZZZ-1:KILL N$:PRINT "objet sup"
0 IF INKEY$="" THEN 2630
0 RETURN
0 '##### ss-prog-save-tobj #####
0 I=1:OPEN "fcatt" FOR OUTPUT AS #1
0 IF I<=ZZZ THEN PRINT #1,OBJ$(I):I=I+1:GOTO 2670 ELSE CLOSE
0 RETURN
0 '##### ss-prog-avant-visu,mod #####
0 IF CR=1 THEN 2720
0 GOSUB 2650:IF A$="2" THEN LOAD"prog2",R ELSE LOAD"prog3",R
0 CLS:PRINT " objet ";NOM$;"non sauvegarder"
0 PRINT " tapez (r) pour sauvegarder"
0 E$=INKEY$:IF E$="" THEN 2740
0 IF E$<>"r" THEN 2710
0 RETURN
0 '##### ss-prog-translation #####
0 X0=0:Y0=0:Z0=0:CR=1:LOCATE 2:INPUT "dOnner le vecteur de translation (x.y
":X0,Y0,Z0
0 PRINT:LOCATE 6:PRINT"PATIENTEZ UN INSTANT S IL VOUS PLAIT"
0 ON TYP GOTO 2800,2800,2800,2830,2830,2830,2840,2850,2830,2830,2800,2852
0 I=1
0 IF I<=NP THEN TPT(I,1)=TPT(I,1)+X0:TPT(I,2)=TPT(I,2)+Y0:TPT(I,3)=TPT(I,3)
I+1:GOTO 2810
0 GOTO 2860
0 TPT(1,1)=TPT(1,1)+X0:TPT(1,2)=TPT(1,2)+Y0:TPT(1,3)=TPT(1,3)+Z0:GOTO 2860
0 TPT(1,1)=TPT(1,1)+X0:TPT(1,2)=TPT(1,2)+Y0:TPT(1,3)=TPT(1,3)+Z0:TPT(4,1)=T
)+X0:TPT(4,2)=TPT(4,2)+Y0:TPT(4,3)=TPT(4,3)+Z0:GOTO 2860
0 TPT(1,1)=TPT(1,1)+X0:TPT(1,2)=TPT(1,2)+Y0:TPT(1,3)=TPT(1,3)+Z0:TPT(6,1)=T
)+X0:TPT(6,2)=TPT(6,2)+Y0:TPT(6,3)=TPT(6,3)+Z0:GOTO 2860
0 TPT(NP,1)=TPT(NP,1)+X0:TPT(NP,2)=TPT(NP,2)+Y0:TPT(NP,3)=TPT(NP,3)+Z0
0 RETURN
0 '##### ss-prog-changement d echelle #####
0 X0=0:Y0=0:Z0=0:CR=1:LOCATE 2:INPUT "dOnner le facteur d echelle (x.y.z) :
Y0,Z0
0 PRINT:LOCATE 6:PRINT"PATIENTEZ UN INSTANT S IL VOUS PLAIT"
0 ON TYP GOTO 2900,2900,2900,2930,2930,2940,2950,2970,2950,2950,2900,2982
0 I=1
0 IF I<=NP THEN TPT(I,1)=TPT(I,1)*X0:TPT(I,2)=TPT(I,2)*Y0:TPT(I,3)=TPT(I,3)
I+1:GOTO 2910
0 GOTO 2990
0 TPT(2,1)=TPT(2,1)*X0:TPT(1,1)=TPT(1,1)*X0:TPT(1,2)=TPT(1,2)*X0:TPT(1,3)=T
)*X0:GOTO 2990
0 TPT(2,1)=TPT(2,1)*X0:TPT(4,1)=TPT(4,1)*X0:TPT(1,1)=TPT(1,1)*X0:TPT(1,2)=T
)*X0:TPT(1,3)=TPT(1,3)*X0:GOTO 2990
0 TPT(1,1)=TPT(1,1)*X0:TPT(1,2)=TPT(1,2)*X0:TPT(1,3)=TPT(1,3)*X0
0 TPT(2,1)=TPT(2,1)*X0:TPT(4,1)=TPT(4,1)*X0:TPT(4,2)=TPT(4,2)*X0:TPT(4,3)=T
)*X0:GOTO 2990
0 TPT(1,1)=TPT(1,1)*X0:TPT(1,2)=TPT(1,2)*X0:TPT(1,3)=TPT(1,3)*X0
0 TPT(4,1)=TPT(4,1)*X0:TPT(4,2)=TPT(4,2)*X0:TPT(4,3)=TPT(4,3)*X0:TPT(2,1)=T
)*X0:TPT(5,1)=TPT(5,1)*X0:GOTO 2990

```



```

982 I=1
983 IF I<=NP-2 THEN TPT(I,1)=TPT(I,1)*X0:TPT(I,2)=TPT(I,2)*Y0:TPT(I,3)=TPT(I,3)*Z0:I=I+1:GOTO 2983
990 RETURN
000 '##### ss-prog-rotation-z #####
010 X0=0:LOCATE 2:INPUT"donner 1 angle de rotation : ";X0
015 PRINT:LOCATE 6:PRINT"PATIENTEZ UN INSTANT S IL VOUS PLAIT"
982 I=1
983 IF I<=NP-2 THEN TPT(I,1)=TPT(I,1)*X0:TPT(I,2)=TPT(I,2)*Y0:TPT(I,3)=TPT(I,3)*Z0:I=I+1:GOTO 2983
990 RETURN
000 '##### ss-prog-rotation-z #####
010 X0=0:LOCATE 2:INPUT"donner 1 angle de rotation : ";X0
015 PRINT:LOCATE 6:PRINT"PATIENTEZ UN INSTANT S IL VOUS PLAIT"
020 I=1:X0=(3.14*X0)/180
030 ON TYP GOTO 3040,3040,3040,3070,3070,3070,3070,3070,3070,3070,3040,3122
040 IF I>NP THEN 3130
050 XV=TPT(I,1):YV=TPT(I,2)
060 TPT(I,1)=XV*COS(X0)-YV*SIN(X0):TPT(I,2)=XV*SIN(X0)+YV*COS(X0):I=I+1:GOTO 070
070 XV=TPT(1,1):YV=TPT(1,2):TPT(1,1)=XV*COS(X0)-YV*SIN(X0):TPT(1,2)=XV*SIN(X0)+YV*COS(X0)
080 XV=TPT(3,1):YV=TPT(3,2):TPT(3,1)=XV*COS(X0)-YV*SIN(X0):TPT(3,2)=XV*SIN(X0)+YV*COS(X0)
090 IF TYP=4 OR TYP=5 OR TYP=6 OR TYP=9 OR TYP=10 THEN 3130
100 IF TYP=7 THEN XV=TPT(4,1):YV=TPT(4,2):TPT(4,1)=XV*COS(X0)-YV*SIN(X0):TPT(4,2)=XV*SIN(X0)+YV*COS(X0):GOTO 3130
110 XV=TPT(4,1):YV=TPT(4,2):TPT(4,1)=XV*COS(X0)-YV*SIN(X0):TPT(4,2)=XV*SIN(X0)+YV*COS(X0)
120 XV=TPT(6,1):YV=TPT(6,2):TPT(6,1)=XV*COS(X0)-YV*SIN(X0):TPT(6,2)=XV*SIN(X0)+YV*COS(X0):GOTO 3130
122 XV=TPT(NP-1,1):YV=TPT(NP-1,2):TPT(NP-1,1)=XV*COS(X0)-YV*SIN(X0):TPT(NP-1,2)=XV*SIN(X0)+YV*COS(X0)
130 RETURN
40 '##### ss-prog-rotation-y #####
50 X0=0:LOCATE 2:INPUT"donner 1 angle de rotation : ";X0
55 PRINT:LOCATE 6:PRINT"PATIENTEZ UN INSTANT S IL VOUS PLAIT"
60 I=1:X0=(3.14*X0)/180
70 ON TYP GOTO 3180,3180,3180,3210,3210,3210,3210,3210,3210,3210,3180,3262
80 IF I>NP THEN 3270
90 XV=TPT(I,1):ZV=TPT(I,3)
00 TPT(I,1)=XV*COS(X0)+ZV*SIN(X0):TPT(I,3)=-XV*SIN(X0)+ZV*COS(X0):I=I+1:GOTO 010
010 XV=TPT(1,1):ZV=TPT(1,3):TPT(1,1)=XV*COS(X0)+ZV*SIN(X0):TPT(1,3)=-XV*SIN(X0)+ZV*COS(X0)
020 XV=TPT(3,1):ZV=TPT(3,3):TPT(3,1)=XV*COS(X0)+ZV*SIN(X0):TPT(3,3)=-XV*SIN(X0)+ZV*COS(X0)
030 IF TYP=4 OR TYP=5 OR TYP=6 OR TYP=9 OR TYP=10 THEN 3270
040 IF TYP=7 THEN XV=TPT(4,1):ZV=TPT(4,3):TPT(4,1)=XV*COS(X0)+ZV*SIN(X0):TPT(4,3)=-XV*SIN(X0)+ZV*COS(X0):GOTO 3270
050 XV=TPT(4,1):ZV=TPT(4,3):TPT(4,1)=XV*COS(X0)+ZV*SIN(X0):TPT(4,3)=-XV*SIN(X0)+ZV*COS(X0)
060 XV=TPT(6,1):ZV=TPT(6,3):TPT(6,1)=XV*COS(X0)+ZV*SIN(X0):TPT(6,3)=-XV*SIN(X0)+ZV*COS(X0):GOTO 3270
062 XV=TPT(NP-1,1):ZV=TPT(NP-1,3):TPT(NP-1,1)=XV*COS(X0)+ZV*SIN(X0):TPT(NP-1,3)=-XV*SIN(X0)+ZV*COS(X0):GOTO 3270
070 RETURN
80 '##### ss-prog-rotation-z #####
90 X0=0:LOCATE 2:INPUT"donner 1 angle de rotation : ";X0
95 PRINT:LOCATE 6:PRINT"PATIENTEZ UN INSTANT S IL VOUS PLAIT"
00 I=1:X0=(3.14*X0)/180
00 ON TYP GOTO 3320,3320,3320,3350,3350,3350,3350,3350,3350,3350,3320,3402
00 IF I>NP THEN 3410

```



```

30 YV=TPT(I,2):ZV=TPT(I,3)
40 TPT(I,2)=YV*COS(X0)-ZV*SIN(X0):TPT(I,3)=YV*SIN(X0)+ZV*COS(X0):I=I+1:GOTO
50 YV=TPT(1,2):ZV=TPT(1,3):TPT(1,2)=YV*COS(X0)-ZV*SIN(X0):TPT(1,3)=YV*SIN(X0)
+COS(X0)
60 YV=TPT(3,2):ZV=TPT(3,3):TPT(3,2)=YV*COS(X0)-ZV*SIN(X0):TPT(3,3)=YV*SIN(X0)
+COS(X0)
70 IF TYP=4 OR TYP=5 OR TYP=6 OR TYP=9 OR TYP=10 THEN 3410
80 IF TYP=7 THEN YV=TPT(4,2):ZV=TPT(4,3):TPT(4,2)=YV*COS(X0)-ZV*SIN(X0):TPT
=YV*SIN(X0)+ZV*COS(X0):GOTO 3410
90 YV=TPT(4,2):ZV=TPT(4,3):TPT(4,2)=YV*COS(X0)-ZV*SIN(X0):TPT(4,3)=YV*SIN(X0)
+COS(X0)
00 YV=TPT(6,2):ZV=TPT(6,3):TPT(6,2)=YV*COS(X0)-ZV*SIN(X0):TPT(6,3)=YV*SIN(X0)
+COS(X0):GOTO 3410
02 YV=TPT(NP-1,2):ZV=TPT(NP-1,3):TPT(NP-1,2)=YV*COS(X0)-ZV*SIN(X0):TPT(NP-1,
V*SIN(X0)+ZV*COS(X0):GOTO 3410
10 RETURN
00 '### ss-pg-cr-cube ###
10 CLS:CR=1:TYP=9:GOSUB 650
20 INPUT "donnez le centre du cube":TPT(1,1),TPT(1,2),TPT(1,3)
30 INPUT "donnez la normale":TPT(3,1),TPT(3,2),TPT(3,3)
40 INPUT "donnez la long d un cote":TPT(2,1)
50 NP=3:NBF=0:RETURN
00 '#### ss-pg-cr-parallelepipede ###
10 CLS :CR=1:TYP=10:GOSUB 650
20 INPUT "donnez le centre du cube":TPT(1,1),TPT(1,2),TPT(1,3)
30 INPUT "donnez la normale":TPT(3,1),TPT(3,2),TPT(3,3)
40 INPUT "donnez la long d un cote":TPT(2,1)
50 NP=3:NBF=0:RETURN
00 '#### ss-pg-cr-parallelepipede ###
10 CLS :CR=1:TYP=10:GOSUB 650
20 INPUT "donnez le centre du parallelipede":TPT(1,1),TPT(1,2),TPT(1,3)
30 INPUT "donnez la normale ":TPT(3,1),TPT(3,2),TPT(3,3)
40 INPUT "donnez la long du cotel":TPT(2,1)
50 INPUT "donnez la long du cote2":TPT(2,2)
60 NP=3 :NBF=0:RETURN
00 CLS:CR=1:TYP=12:GOSUB 650:PRINT "donnez les points de l objet (30 points
maximum)":GOSUB 700:NF=0
10 NP=NP+1:INPUT"donnez le vecteur directeur (x,y,z):":TPT(NP,1),TPT(NP,2),
P,3)
11 NP=NP+1:INPUT"donnez un point de reference (x,y,z):":TPT(NP,1),TPT(NP,2),
NP,3)
20 RETURN

```



```

10 '##### prog2-modification #####
20 DIM TPT(150,3),TBA(200),OBJ$(40)
30 MPT=150
40 GOSUB 1090
50 GOSUB 1320
60 IF CR=1 THEN 120
70 PRINT "tapez (c) pour continuer,(r) pour quitter"
80 A$=INKEY$:IF A$=""THEN 80
90 IF A$="c" THEN 40
100 IF A$="r" THEN LOAD"prog1",R
110 GOTO 80
120 CLS:GOSUB 420:LOCATE 10,3
130 PRINT "VOUS POUVEZ"
140 LOCATE 15,7:PRINT "1:CONSULTER"
150 LOCATE 15,8:PRINT "2:MODIFIER LES COORDONNEES"
160 LOCATE 15,9:PRINT "3:MODIFIER LES ARETES"
170 LOCATE 15,10:PRINT "4:MODIFIER UN AUTRE OBJET"
180 LOCATE 15,11:PRINT "5:RETOURNER AU MENU"
190 LOCATE 15,14:PRINT " VOTRE CHOIX : _"
200 A$=INKEY$:IF A$="" THEN 200
210 IF A$="1" THEN GOSUB 550:GOTO 120
220 IF A$="2" THEN GOSUB 460:GOTO 120
230 IF A$="3" THEN GOSUB 340:GOTO 120
240 IF A$="4" THEN GOSUB 1520:GOTO 50
250 IF A$="5" THEN GOSUB 270
260 GOTO 200
270 '##### RETOURNER AU MENU #####
280 IF MD=0 THEN 320
290 CLS:PRINT "TAPEZ (s) POUR SAUVEGARDER,AUTRE SINON"
300 A$=INKEY$:IF A$="" THEN 300
310 IF A$="s" THEN GOSUB 1160
320 GOSUB 1480:LOAD"PROG1",R
330 RETURN
340 '##### MOD-AR #####
350 IF NBF=0 THEN 410 ELSE MD=1
360 PRINT "DONNER LES ENSEMBLE D ARETES SEPARES PAR (-1),A LA FIN TAPEZ (-2)
370 VAR=1
380 INPUT P:IF (P<>-2) AND (P<>-1) AND (P<1 OR P>NP) THEN PRINT "ERREUR":GO
390 TBA(VAR)=P:VAR=VAR+1
400 IF P<>-2 THEN 380
410 RETURN
420 REM
430 PRINT "#####
#####"
440 LOCATE 1,20:PRINT"#####
#####"
450 RETURN
460 REM mod-point
470 CLS
480 INPUT "donner le numero du point a mod ,(-1) sinon":N
490 IF N=-1 THEN 540 ELSE MD=1
500 IF N<0 OR N>NP+2 THEN PRINT "ERREUR":GOTO 480
510 INPUT "(X,Y,Z)":TPT(N,1),TPT(N,2),TPT(N,3)
520 IF N=NP+1 THEN NP=NP+1
530 GOTO 480
540 RETURN
550 REM consulter
560 CLS
570 INPUT "nom de l'objet : ";NOMS$
580 PRINT "type de l'objet : ";typ
590 ON TYP GOTO 600,610,620,630,640,650,660,670,680,690,700,710

```



```

600 PRINT "point ":GOTO 720
610 PRINT "arete ":GOTO 720
620 PRINT "face ":GOTO 720
630 PRINT "cercle ":GOTO 720
640 PRINT "sphere":GOTO 720
650 PRINT "cylindre":GOTO 720
660 PRINT "cone":GOTO 720
670 PRINT "tronc de cone":GOTO 720
680 PRINT "cube":GOTO 720
690 PRINT "parallepipede":GOTO 720
700 PRINT "objet complexe":GOTO 720
710 PRINT "objet a surface de revolution":GOTO 720
720 ON TYP GOTO 730,730,730,810,840,860,900,940,1000,1030,730,1062
730 PRINT "nombre de points : ";NP
740 I=1
750 IF I<= NP THEN PRINT"point :";I,TPT(I,1);TPT(I,2);TPT(I,3):I=I+1:GOTO 7
760 IF NBF=0 THEN 1070
770 I=1
780 IF TBA(I)=-2 THEN 1070
790 IF TBA(I)=-1 THEN PRINT TBA(I):PRINT :ELSE PRINT TBA(I);
800 I=I+1:GOTO 780
810 PRINT "centre du cercle (1) : ";TPT(1,1);TPT(1,2);TPT(1,3)
820 PRINT "rayon (2) : ";TPT(2,1)
830 PRINT "la normale (3) : ";TPT(3,1);TPT(3,2);TPT(3,3):GOTO 1070
840 PRINT "centre du sphere (1) : ";TPT(1,1);TPT(1,2);TPT(1,3)
850 PRINT "rayon (2) : ";TPT(2,1):GOTO 1070
860 PRINT "centre du cercle (1) : ";TPT(1,1);TPT(1,2);TPT(1,3)
870 PRINT "rayon (2) : ";TPT(2,1)
880 PRINT "la normale (3) : ";TPT(3,1);TPT(3,2);TPT(3,3):GOTO 1070
890 PRINT "la hauteur (4) : ";TPT(4,1):GOTO 1070
900 PRINT "centre du cercle (1) : ";TPT(1,1);TPT(1,2);TPT(1,3)
910 PRINT "rayon (2) : ";TPT(2,1)
920 PRINT "la normale (3) : ";TPT(3,1);TPT(3,2);TPT(3,3)
930 PRINT "le sommet (4) : ";TPT(4,1);TPT(4,2);TPT(4,3):GOTO 1070
940 PRINT "centre du 1er cercle (1) : ";TPT(1,1);TPT(1,2);TPT(1,3)
950 PRINT "rayon (2) : ";TPT(2,1)
960 PRINT "la normale (4) : ";TPT(4,1);TPT(4,2);TPT(4,3)
970 PRINT "centre du 2emme cercle (4) : ";TPT(4,1);TPT(4,2);TPT(4,3)
980 PRINT "rayon (5) : ";TPT(5,1)
990 PRINT "la normale (6) : ";TPT(6,1);TPT(6,2);TPT(6,3)
1000 PRINT "centre de cube (1) : ";TPT(1,1);TPT(1,2);TPT(1,3)
1010 PRINT "la normale (3) : ";TPT(3,1);TPT(3,2);TPT(3,3)
1020 PRINT "la long d un cote (2) : ";TPT(2,1):GOTO 1070
1030 PRINT "centre de parallepipede (1) : ";TPT(1,1);TPT(1,2);TPT(1,3)
1040 PRINT "la normale (3) : ";TPT(3,1);TPT(3,2);TPT(3,3)
1050 PRINT "la long de cote 1 (2,1) : ";TPT(2,1)
1060 PRINT "la long de cote 2 (2,2) : ";TPT(2,2):GOTO 1070
1062 I=1
1063 IF I<=NP-2 THEN PRINT "point :";I,TPT(I,1),TPT(I,2),TPT(I,3):I=I+1:GOTO
1064 PRINT "le vecteur directeur (";NP-1;) : ",TPT(NP-1,1);TPT(NP-1,2);TPT(
1065 PRINT "le point de reference (";NP;) : ",TPT(NP,1);TPT(NP,2);TPT(NP,3)
1070 IF INKEY$="" THEN 1070
1080 RETURN
1090 REM -----chargement de la table tabobj-----
1100 ZZZ=0
1110 OPEN "fcat" FOR INPUT AS #1
1120 IF EOF(1) THEN CLOSE:GOTO 1150
1130 ZZZ=ZZZ+1:INPUT #1,OBJ$(ZZZ)
1140 GOTO 1120
1150 RETURN

```



```

1160 '##### ss-prog-save #####
1170 CLS:PRINT "sauvegard de 1 objet ...":PRINT:PRINT " PATIENTEZ UN INS
S IL VOUS PLAIT"
1180 IF NOM$="" THEN PRINT"pas d objet a sauvegarder":GOTO 1300
1190 CR=0
1200 OPEN NOM$ FOR OUTPUT AS #1
1210 PRINT #1,NOM$:PRINT #1,TYP:PRINT #1,NP
1220 I=1
1230 IF I<= NP THEN PRINT #1,TPT(I,1),TPT(I,2),TPT(I,3):I=I+1:GOTO 1230
1240 PRINT #1,NBF
1250 IF NBF=0 THEN 1300
1260 I=1
1270 PRINT #1,TBA(I)
1280 IF TBA(I)=-2 THEN 1300
1290 I=I+1:GOTO 1270
1300 CLOSE
1310 RETURN
1320 '##### ss-prog-load #####
1330 CLS:INPUT"donnez le nom de 1 objet a modifier : ";NOM$
1340 I=1:MD=0
1350 IF I>ZZZ THEN 1380
1360 IF OBJ$(I)=NOM$ THEN 1390 ELSE I=I+1:GOTO 1350
1370 A$=INKEY$:IF A$="" THEN 1370
1380 PRINT "objet inexistant":GOTO 1470
1390 IND=I:CR=1:DDD=1:OPEN NOM$ FOR INPUT AS #1
1400 INPUT #1,NOM$:INPUT #1,TYP:INPUT #1,NP
1410 I=1
1420 IF I<= NP THEN INPUT #1,TPT(I,1),TPT(I,2),TPT(I,3):I=I+1:GOTO 1420
1430 INPUT #1,NBF
1440 IF NBF=0 THEN CLOSE:GOTO 1470
1450 I=0
1460 IF EOF(1) THEN CLOSE ELSE I=I+1:INPUT #1,TBA(I):GOTO 1460
1470 RETURN
1480 '##### ss-prog-save-tobj #####
1490 I=1:OPEN "fcats" FOR OUTPUT AS #1
1500 IF I<=ZZZ THEN PRINT #1,OBJ$(I):I=I+1:GOTO 1500 ELSE CLOSE
1510 RETURN
1520 REM
1530 IF MD=0 THEN 1570
1540 CLS:PRINT "tapez (s) pour sauvegarder,autre sinon"
1550 A$=INKEY$:IF A$="" THEN 1550
1560 IF A$="s" THEN GOSUB 1160
1570 RETURN

```