

**UNIVERSITE DE DROIT, D'ECONOMIE et de SCIENCES SOCIALES  
de PARIS (PARIS II)**

---

**MODELISATION ET ARCHITECTURE DE  
MACHINES DE SYNTHÈSE D'IMAGES POUR LA  
REPRESENTATION ET LE RENDU D'OBJETS TRI-  
DIMENSIONNELS (3D) SUR ECRANS GRAPHIQUES A  
BALAYAGE**

**pour le doctorat de l'Université de Paris 2  
(arrêté du 23 novembre 1988)**

***SCIENCES DE L'INFORMATION  
MENTION IMAGES, MEDIAS, ET INFORMATIQUE DE LA  
COMMUNICATION  
(IMAC)***

**présentée et soutenue publiquement par:**

*Monsieur Pascal LERAY*

---

**Jury:**

**Directeur de thèse: Monsieur J. DONIO** Professeur à l'Université de Paris II

**Membres du Jury: Monsieur J.E. DUBOIS** Professeur à l'Université de Paris VII

**Madame J. POINSSAC-NIEL** Professeur à l'Université de Paris II,  
Directeur de l'Institut IMAC

---

Date de soutenance: Mardi 25 septembre 1990

L' UNIVERSITE DE DROIT, D'ECONOMIE et de SCIENCES SOCIALES de PARIS (PARIS II) n'entend donner aucune approbation ni improbation aux opinions émises dans les thèses, ces opinions devront être considérées comme propres à leurs auteurs

<b>RESUME de THESE:</b>
-------------------------

La synthèse d'images tri-dimensionnelle peut être définie comme "l'art de modéliser et de dessiner ce qui n'existe pas, ou pas encore". Elle nécessite la conjonction de ces deux fonctions essentielles que l'Homme possède naturellement grâce à l'ensemble oeil - cerveau - membres supérieurs et qu'aucune machine ne peut actuellement réaliser.

- La fonction de "modélisation" chez l'Homme coorespond en informatique graphique non seulement à la tâche simple de **saisie** numérique point à point de l'image visible, mais encore à une "**modélisation**", c'est à dire à l'élaboration d'une structure de données sémantiques contenant l'ensemble des paramètres de représentation et des propriétés attachés à chaque objet nouveau, associées à celles de l'ensemble déjà mémorisé par le système.
- La fonction de "dessin" correspond au **rendu** réaliste des objets, ou "synthèse d'image", ce terme induisant le sens d'"artificiel" et également le sens de "synthétique".

La réalisation matérielle de ces deux fonctions comporte actuellement deux problèmes théoriques majeurs:

- Les meilleures techniques de modélisation actuelles ne peuvent en aucun cas effectuer une analyse sémantique complète tri-dimensionnelle d'une collection d'objets et de leurs mouvements.
- Les techniques de rendu de qualité sont très lentes (plusieurs dizaines d'heures pour une image) ou de qualité très insuffisante lorsque l'on souhaite réduire les temps de calculs.

Après un historique de la synthèse d'image suivi d'un inventaire des méthodes théoriques et techniques actuellement utilisées en modélisation et en rendu 3D, l'objet de cette thèse consiste:

- A en dresser une analyse critique.
- A proposer pour ces deux problèmes des solutions théoriques et techniques nouvelles, ayant deux objectifs majeurs:

*Pour le premier*, permettre à l'utilisateur une ergonomie de travail véritablement 3D pour la modélisation des objets: L'idéal visé serait de retrouver le type d'ergonomie de la sculpture traditionnelle, pilotée grâce à des procédés électroniques.

*Pour le second*, améliorer économiquement les temps de réponse des matériels, clé de voute du développement de l'image de synthèse. Celle-ci coûte actuellement très cher (de l'ordre de 10 à 20 fois le coût moyen d'une image "naturelle", c'est à dire issue de caméras de télévision )

Enfin l'apparition récente des techniques neuro-mimétiques apportera à la saisie et au rendu des images 3D des possibilités nouvelles susceptibles d'améliorer de façon sensible les techniques actuelles. Nous esquisserons enfin une solution consistant à faire coopérer -à

l'instar de la vision humaine et des aires cervicales - des réseaux neuronaux spécialisés avec boucle de retour, comparant en permanence la sortie dessinée sur l'écran à la mémoire acquise par l'exemple des objets à représenter. Chaque réseau étant adapté à un type d'analyse donné pour ce qui concerne la texture, le relief, la couleur et le mouvement. Un tel système permettrait de modéliser des objets directement à partir d'une caméra, d'en extraire une représentation interne 3D neuronale, enfin de restituer un rendu réaliste à la manière de l'objet original et des contraintes particulières demandées par l'opérateur.

Le théoricien et l'informaticien se doivent d'apporter des solutions à toutes les formes de représentation 3D, tout en restant vigilants sur ce qui est véritablement demandé à moyenne échéance, compte-tenu de la riche et passionnante complexité de l'image de synthèse. La solution neuronale constituant une voie d'avenir vers une informatique graphique sans programmation, reposant comme l'éducation humaine sur l'apprentissage par l'exemple et la corrélation permanente avec l'ensemble des connaissances concentrées dans le cerveau humain.

---

**REMERCIEMENTS:**

A Monsieur J. PONCIN, Directeur du CCETT et  
A Monsieur D. NASSE, chef du Groupement Traitement d'image Audio-Visuel  
qui depuis 1980 apprécièrent et me permirent de construire la substance qu'elle  
contient, j'exprime ma vive reconnaissance.

A Madame le Professeur J. POINSSAC-NIEL, Directrice de l'Institut IMAC,  
qui contribua activement au développement de l'enseignement du graphique et de la  
synthèse d'image en France, et qui par ses encouragements constants, m'a permis de  
mener à bien cet ouvrage.

A Monsieur le Professeur J. DONIO, qui par ses précieux conseils m'aida à la  
rédaction de cette thèse.

A Monsieur le Professeur J.E. DUBOIS, qui apporta à l'infographie 3D en  
France une inappréciable contribution, et qui ne ménagea pas ses nombreux et  
pertinents conseils pour la conduite et la finalisation de ce long travail.

A ma femme, Maggie, à mes enfants

Christian, Cécile et Jean-François

A mes Parents

**SOMMAIRE**

RESUME de THESE:.....3

1/LA SYNTHÈSE D'IMAGE ou L'ART DE VISUALISER LE NON MANIFESTE:.....11

1-1/L'HISTOIRE PRODIGIEUSE DE LA SYNTHÈSE D'IMAGE: ou L'ART DE VOIR LA SURREALITE.....12

1-2/INTRODUCTION:.....18

1-2-1/Le contexte général actuel de la synthèse d'images- Définitions techniques 18

1-2-2/ POSITION DU PROBLEME DE CHOIX COUT-QUALITE-TEMPS.....18

1-2-3/ LE PROBLEME ACTUEL DE LA SYNTHÈSE D'IMAGES.....20

2/LES DIFFÉRENTES METHODES DE MODELISATION: ETUDE COMPARATIVE DES SOLUTIONS ACTUELLES.....23

2-1/ LA MODELISATION PAR VECTEURS.....23

2-2/ LA MODELISATION PAR SURFACES.....24

2-2-1/ MODELISATION PAR SURFACES ELEMENTAIRES PLANES.....24

2-2-2/LES SURFACES MATHÉMATIQUES.....24

2-2-3/LES SURFACES d'HERMITE.....25

2-2-4/SURFACES DE BEZIER.....26

2-2-5/SURFACES B-SPLINE CUBIQUES.....27

2-2-6/ Comparaison critique de ces différentes formes de modélisation.....27

2-3/ LA MODELISATION PAR MAILLAGE VOLUMIQUE.....27

2-4/ LA MODELISATION PAR CONSTRUCTION SOLIDE (CSG).....28

2-5/Passage à d'autres MODES DE représentations.....29

2-6/PAR NUMÉRISATION DIRECTE D'OBJETS.....30

2-6-1/NUMÉRISATION MÉCANIQUE.....31

2-6-2/ NUMÉRISATION PAR MESURE DE LA POSITION d'un STYLET.....31

2-6-3/PROCÉDE PAR CAMERA INFRA-ROUGE:.....32

2-6-4/PAR SYSTÈME LASER.....32

2-6-6/CONCLUSION SUR LES SYSTÈMES DE NUMÉRISATION DIRECTE34

3/LE RENDU RÉALISTE DES IMAGES 3D (RENDERING) ETUDE COMPARATIVE DES SOLUTIONS ACTUELLES.....35

3-1/ LES PRINCIPALES TECHNIQUES ALGORITHMIQUES DE RENDU.....35

- 3-1-1/LES ALGORITHMES FONDAMENTAUX D'ELIMINATION DES PARTIES CACHEES:.....35
- 3-1-2/ LES ALGORITHMES D'AMELIORATION DE LA QUALITE DES IMAGES: .....36
- 3-2/ LES GRANDS TYPES D'ARCHITECTURES DE MACHINES DE RENDU:.....36
  - 3-2-1/MACHINES DE TYPE TAMPON EN Z (Z-BUFFER).....36
  - 3-2-2/MACHINES DE TYPE LANCER DE RAYONS.....39
  - 3-2-3/MACHINES MIXTES-SOLUTIONS RECENTES :.....41
- 3-3/ LES PROBLEMES DE LIEN ENTRE MODELISATION ET RENDU.....42
  - 3-3-1/LES CONTRAINTES DE REPRESENTATION INTERNE DES OBJETS POUR LA MODELISATION.....42
  - 3-3-2/LES CONTRAINTES TECHNIQUES DE REPRESENTATION DES OBJETS POUR LE RENDU:.....43
  - 3-3-3/LES CONTRAINTES ERGONOMIQUES DE LA REPRESENTATION DES OBJETS.....44

**4/ UNE SOLUTION NOUVELLE POUR LA MODELISATION ELECTRONIQUE DIRECTE D'OBJETS 3D.....46**

- 4- 1/ ANALYSE CRITIQUE DES DIFFERENTES METHODES DE MODELISATION 46
- 4- 2/ LA SOLUTION PROPOSEE.....46
  - 4-2-1/DOMAINE D'UTILISATION.....46
  - 4-2-2/DESCRIPTION THEORIQUE.....47
  - 4-2-3/IMPLANTATION TECHNIQUE.....48
  - 4-2-4/ETUDE DES PERFORMANCES.....50
  - 4-2-5/LIMITES ET AMELIORATIONS POSSIBLES.....50
- 4-3/ CONCLUSION, LIMITES ET AVANTAGES DE LA SOLUTION PROPOSEE.50
- 4-4/ UNE PROPOSITION NOUVELLE DE SYSTEME DE MODELISATION NEURO-MIMETIQUE:.....50

**5/UNE PROPOSITION d'ARCHITECTURE ORIGINALE DE MACHINE DE RENDU.....55**

- 5- 1/ ETUDE CRITIQUE DES SOLUTIONS ACTUELLES.....55
- 5- 2/ PROPOSITION D'ARCHITECTURE DE RENDU: CUBICAST.....55
  - 5-2-1/DOMAINE D'UTILISATION.....55
  - 5-2-2/DESCRIPTION THEORIQUE.....55
  - 5-2-3/IMPLANTATION TECHNIQUE.....56
  - 5-2-4/ ETUDE DU FONCTIONNEMENT ET CARACTERISTIQUES ORIGINALES DE CETTE ARCHITECTURE.....61
  - 5-2-5/ETUDE DES PERFORMANCES.....62
  - 5-2-6/LIMITES ET AMELIORATIONS POSSIBLES.....66
- 5-3/ CONCLUSION, LIMITES ET AVANTAGES DE LA SOLUTION PROPOSEE.66

5-4/ L'EMERGENCE et le lien avec LES SYSTEMES NEURO-MIMETIQUES.....69

6/ CONCLUSION GENERALE et CONSEQUENCES DES PROPOSITIONS PRECEDENTES SUR L'INDUSTRIE DE L'IMAGE.....71

ANNEXE 1: LE TRANSPUTER.....74

ANNEXE 2: SCHEMA SYNOPTIQUE DU TRANSPUTER.....75

ANNEXE 3 :TECHNIQUES DES PRIORITES.....76

ANNEXE 4: INTERSECTIONS DE VOLUMES et MAILLAGES.....77

ANNEXE 5: IMAGES DE SCULPTURE ELECTRONIQUE SUR ECRAN.....81

ANNEXE 6 :ETUDES ET DEVELOPPEMENTS PERSONNELS SUR L'INFOGRAPHIE.82

PRESENTATION CHRONOLOGIQUE.....82

7-1/ ETUDE SUR LES CALCULATEURS SYNCHRONES DANS LE CADRE DE LA SOCIETE RADIOTECHNIQUE EN 1969:.....85

7-2/ ETUDES et DEVELOPPEMENTS GRAPHIQUES REALISES AU CELAR.....85

7-2-1/Les ordres graphiques.....85

7-2-2/Les ordres de gestion des structures.....86

7-2-3/Les ordres d'interface avec le calculateur hôte.....88

7-2-4/CONCLUSION.....90

7-3/ MES RECHERCHES EN SYNTHESE D'IMAGE AU CELAR.....90

7-4/ MES RECHERCHES AU CCETT.....93

ANNEXE 7: Formulation mathématique de l'équation économique des machines graphiques98

ANNEXE 8: LES TECNHNiques NEURO-MIMETIQUES POUR LA SAISIE et la SYNTHESE DES FORMES 3D et des MOUVEMENTS.....99



ANNEXE 9: UTILISATION DES RESEAUX NEURO-MIMETIQUES POUR L'ANALYSE et LA SYNTHÈSE DES IMAGES.....	102
9-1/DEFINITION DES SYTEMES NEURO-MIMETIQUES.....	102
9-2/Extrait du texte de l'exposé présenté par J. HERAULT (HER 89) aux Journées d'électronique 89 de l'Ecole Polytechnique Fédérale de LAUSANNE:.....	104
ANNEXE 10: VISION PROSPECTIVE DE LA SAISIE et de la SYNTHÈSE D'IMAGE EN l'AN 2000.....	106
ANNEXE 11:NOTES CONTRIBUTIVES sur le NEURONAL.....	108
ANNEXE 12: ANALYSE STATISTIQUE D'IMAGE PAR FONCTIONS PROBABILISTES ITERATIVES FRACTALES.....	110
12-1/ RAPPEL SUR LA THEORIE DU CHAOS ET DES FRACTALES-LIEN AVEC LES RESEAUX NEURONAUX.....	110
12-2/ LES FONCTIONS ITERATIVES.....	110
LISTING APL DE LA FONCTION IFS FRACTALE ITERATIVE et PROBABILISTE POUR LA FEUILLE DE FOUGERE.....	113
LISTING DU LOGICIEL NEURONAL EN APL POUR L'ANALYSE ET LA RECONNAISSANCE DE FORMES.....	114
EXEMPLES D'IMAGES DE SYNTHÈSE.....	121

**TABLE DES FIGURES**

Fig 1: Exemple de représentation par surfaces limites ou Boundary Representation (B-Rep).....25

Fig 2: Solide à modéliser.....29

Fig 3: Exemple d'objet modélisé en CSG avec son graphe correspondant.....29

Fig 4: Exemple de passage de la CSG à d'autres modes de représentation.....30

Fig 4: Principe de l'algorithme du tampon en Z.....38

Fig 5: Principe de l'algorithme du lancer de rayons.....40

Fig 7 Poste de sculpture 3D sur écran.....47

Fig 8 Détail de l'écran: outil virtuel de sculpture 3D sur écran.....48

Fig 9: Schéma synoptique de Cubicast.....61

Fig 10: Machine d'analyse - restitution neuronale.....69

Fig 11: TRANSPUTER à Architecture RISC (Reduced Instruction Set Computer).....75

Fig 13: Différence de 2 volumes polygonaux.....78

Fig 14: Intersection de 2 volumes polygonaux.....78

Fig 15: Structure de donnée de GRAFOR.....87

Fig 16: Modélisation Neuronale d'objets 3D.....105

Fig 17: Exemple de fonction probabiliste itérative IFS:.....112

Modélisation d'une feuille de fougère (voir listing ci-dessous).....112

## 1/LA SYNTHÈSE D'IMAGE ou L'ART DE VISUALISER LE NON MANIFESTE:

"Voir ce qui n'existe pas, ou pas encore", c'est à dire le sans-forme, le non encore comme le déjà manifesté, telle pourrait être une définition préliminaire de la synthèse d'image. Cette définition porte en elle la dimension exceptionnelle d'une technique qui touche à l'heure actuelle bien plus qu'une simple branche de l'industrie informatique:

A l'aspect théorique complexe qu'elle sous tend, et pour lequel cette thèse est consacrée, s'ajoute en effet une dimension culturelle et métaphysique:

*-Culturelle* tout d'abord, puisque par son caractère spectaculaire de nouveau moyen d'expression, elle touche le grand public grâce à sa diffusion par l'intermédiaire de la Télévision ou du Cinéma. Elle sort donc du cadre classique des laboratoires et des bureaux d'étude, hôtes habituels des technologies modernes. Depuis la plus haute antiquité, la maîtrise de l'image a été vue de façon plus ou moins nette par les élites et par le pouvoir religieux comme l'instrument privilégié de contrôle des esprits humains de leurs contemporains.

Rappelons par exemple la réglementation de la fabrication des images dans beaucoup de religions, en Afrique l'interdiction faite aux profanes de confectionner des statues à caractère religieux, le sorcier seul ayant ce droit; Citons également l'importance de l'image peinte ou sculptée dans les religions, les statues géantes de divinités païennes par exemple. L'expérience des siècles passés a en effet démontré l'importance de la Représentation d'une forme (sculptée ou peinte) sur ceux à qui elle est adressée. La part *exotérique* des cérémonies culturelles passées ou actuelles est en fait constituée d'une série de présentations animées ou fixes de tableaux, d'objets, de sculptures, chacun symbolisant concrètement un principe, une idée, ou une divinité, souvent accompagnés d'une symbolique musicale (orgue, chant), complétant la symbolique visuelle.

Il est rare d'observer un culte religieux dénué de tout support matériel visuel ou sonore. <sup>(1)</sup>

*Métaphysique* ensuite, puisqu'à l'heure actuelle des physiciens tels que Hawkins (HAW 89) en Angleterre ou A. Aspect en France, et Michaël Talbot (TAL 89) aux USA découvrent l'existence de la multi-dimensionnalité de l'univers, débouchant sur les notions controversées de l'inexistence du temps, de l'équivalence Information- énergie- matérialisation de la matière, donc en définitive à la synthèse d'image de l'univers lui-même, tel que perçu par notre rétine et modélisé par notre cerveau:

---

<sup>(1)</sup> Le ZEN pourrait infirmer cette règle; Cependant les séances de méditation du ZEN ou "ZAZEN" nécessitent un certain nombre d'éléments matériels indispensables: un local, le coussin de méditation ou ZAFOU et des repères visuels sur les murs. Le vide total n'existe pas, même dans l'espace intersidéral.

Michael TALBOT en particulier, n'hésite pas à avancer l'hypothèse d'un univers constitué d'un nombre considérable de "pixels" ( $10^{89}$  exactement) ou points solides tri-dimensionnels, qui pourraient commuter du non-matérialisé au matérialisé, en fonction de l'information qu'ils peuvent recevoir: L'univers, d'après ce chercheur, serait en quelque sorte une immense machine de synthèse d'image dont toutes les manifestations visibles présentes, passées ou futures existent de toute éternité, et qui sont en constante interaction avec l'Homme qui les observe et interfère constamment avec elles.

Il est hélas clair que l'état de la synthèse d'image conçue par l'homme est loin d'avoir le degré d'interaction de l'Univers, bien que chaque jour apporte une part non négligeable "d'améliorations". (2)

C'est précisément sur ce point essentiel de l'interaction entre l'observé et l'observant que porte cette thèse (Modélisation Tri-dimensionnelle et Architectures de Rendu) . L'on constate en fait de façon très concrète cette interaction dans tout processus de conception: Une conception artistique ou technique donnera difficilement le même résultat sur 2 systèmes différents: Le système graphique comporte donc en lui-même un poids culturel et technique de nature à influencer l'opérateur et par contrecoup l'ouvrage ou l'oeuvre produits.

De même certaines théories scientifiques récentes, en particulier la théorie dite "anthropique", affirme que les lois physiques d'évolution de l'univers ne peuvent faire abstraction des créatures qui l'examinent. Plus récemment encore, l'on a démontré qu'une variation de quelques milliardièmes de secondes dans la marche d'une planète (telle que celle qui résulte des lancements de nos fusées spatiales) conduit à un comportement complètement imprévisible de celle-ci quelque centaines de milliers d'années plus tard.

Dans le même ordre d'idées, les hypothèses souvent utilisées des systèmes fermés sont des modèles considérés comme insuffisants pour décrire les lois physiques de l'Univers.

## **1-1/L'HISTOIRE PRODIGIEUSE DE LA SYNTHÈSE D'IMAGE: ou L'ART DE VOIR LA SURREALITE**

Le véritable démarrage de la synthèse d'image tri-dimensionnelle sur écran graphique à balayage Télévision peut se situer vers les années 68-69 aux USA: Les premières mémoires d'images, permettant une représentation colorée des objets y ont fait leur apparition, autorisant la visualisation par projection d'objets en volume.

La première application fondamentale de ces systèmes, et le principal moteur de leur développement a été incontestablement la simulation de vol: Les vols spatiaux APOLLO et la conquête de la Lune ont donné un coup de fouet décisif au développement de représentations visuelles d'environnements réalistes pour l'entraînement de pilotes ou de cosmonautes. Par la suite, les simulateurs de vols militaires ou civils ont pris une extension considérable.

Bien entendu, l'on peut faire remonter à plus loin les premières tentatives de représentations visuelles sur écran: Cependant, il convient de noter qu'à cette

---

(2) Entre guillemets, puisqu'il est impossible "d'améliorer " ce qui en fait préexiste déjà,

sans toutefois être manifesté

époque, (et à fortiori avant), (LER 72) il était impossible de stocker dans une mémoire le contenu d'une image de télévision, et encore moins de la relire à la cadence de la Télévision c'est à dire à 50 ou 60 Hz. Des représentations dites filaires (ou vectorielles) ont vu le jour vers les années 69-70, mais en l'absence de mémoire d'image, on peut difficilement parler de rendu réaliste des objets solides: tout au plus, peut-on représenter des silhouettes d'objets sous forme de vecteurs colorés ou non:

Ce fut "L'ERE PRIMAIRE DES ALGORITHMES D'ELIMINATION DES LIGNES CACHEES": Très utilisés surtout en architecture ou en CAO, ils permettaient de représenter des projections d'objets solides 3D sous forme de vecteurs, et de les reproduire sur des tables traçantes à plumes ou sur des écrans à balayage cavalier.

Lorsque les premières mémoires d'images apparurent, en FRANCE en 70, aux USA 5 ans plus tôt environ, le kilo-octet de RAM coûtait environ 100 fois le prix actuel (500 Frs environ), et nécessitait 3 tensions d'alimentation: + et - 5V, +12 V. Sa fiabilité était des plus incertaines, puisque tous les grands constructeurs d'ordinateurs faisaient plutôt confiance aux mémoires à tores de ferrite, ceci jusqu'aux années 75: Le MITRA 15 de la CII (Compagnie Internationale pour l'Informatique) a été l'un des derniers calculateurs au monde utilisant des mémoires à tore de ferrite.

Il était alors difficile d'imaginer une carte mémoire d'image d'ordinateur comportant les 3 fois 8 bits nécessaires <sup>(3)</sup> à la représentation des couleurs fondamentales Rouge Vert Bleu, que l'on trouve à l'heure actuelle sur des calculateurs personnels pour moins de 10000 Frs. Une telle carte, à supposer qu'elle puisse même exister avec des boîtiers de 1 Kbits seuls disponibles alors, coûterait environ 3 600 000 Frs au prix pratiqué pour les mémoires en 69-70 et indiqué plus haut.

A partir de cette époque (70) un formidable engouement s'est produit aux Etats Unis en faveur de la synthèse d'image, en parallèle avec quelques rares chercheurs isolés en Europe. A cette époque sortirent les premiers algorithmes d'élimination des parties cachées des objets solides: algorithmes de WARNOCK et de WATKINS, dont une tentative de câblage échoua aux USA en raison d'une complexité algorithmique excessive, et compte-tenu des possibilités technologiques de l'époque.

De plus, les militaires Européens ainsi que les centres de recherche de l'Armement ne prirent en compte l'intérêt de la simulation et de la synthèse d'image que 5 à 7 ans environ après leurs homologues US. Ils ne croyaient pas à l'intérêt du simulateur, et privilégièrent les vols réels jusqu'au renchérissement du pétrole en 74. La crise du pétrole aidant, les choses évoluèrent à partir des années 75-76, lorsque des délégations françaises en visite aux Etats-Unis y découvrirent

---

(3) 3 fois 8 bits sont suffisants (mais non nécessaires) pour une représentation complète d'une image couleur sur écran: En dessous de 8 bits par composantes l'oeil est sensible aux sauts de quantification dans l'échelle des gris pour chaque composante. 7 bits seraient en fait suffisants. (Cf ouvrages sur la colorimétrie en Télévision)

de gigantesques simulateurs, constitués de cabines montées sur vérins, et disposant de systèmes de visualisation très performants pour l'époque. Ces systèmes étaient basés soit sur des maquettes de terrains pour les plus anciens, soit sur de véritables matériels temps réel de visualisation d'images de synthèse pour les plus récents.

De 75 à 80, de nombreux algorithmes de synthèse d'image apparurent aux Etats-Unis, parfois aussi avec leurs homologues français intéressés par ce domaine.

C'est à cette époque que s'élaborèrent les meilleurs algorithmes d'élimination des surfaces cachées:

- A priorité statiques: (SHUMACHER)
- A priorité semi-statiques(NEWELL)
- Dynamiques (Z-BUFFER)

Pendant le même temps furent implantés au CELAR des algorithmes qui conduisirent à la première machine temps réel construite sur le sol Européen en 1979 (voir Annexe 5). (LER 80 - 81) Cette machine fut présentée aux USA en 81 lors de la conférence "IMAGE II Conference" à PHOENIX, Arizona. (LER 81).

En 81, les simulateurs de vol US occupaient couramment le volume d'une vingtaine de baies informatiques pour une capacité d'affichage de 1000 facettes relativement sommaires en monochrome. (ex: La machine de Général Electric qui fonctionnait en 80 sur la PHOENIX Air Force Base)

La machine réalisée au CELAR (Annexe 5) (LER 81) et baptisée GSI ou Générateur pour la Synthèse des Images) affichait 4 à 500 facettes en temps réel en couleur, et tenait alimentation comprise le volume d'un quart de baie informatique. De plus cette machine comportait 2 processeurs parallèles, dont le second n'a jamais été utilisé (un seul étant suffisant pour l'affichage de 400 facettes) (LER 81), l'objectif du CELAR étant la représentation d'un avion-cible de 200 facettes environ pour la simulation. Les performances de cette machine étaient donc à cette époque supérieures à celles de la machine de Général Electric, pour environ le sixième de son volume...

A cette époque, les images de synthèse furent utilisées en dehors du secteur applicatif de la simulation militaire ou civile: avions, chars, hélicoptères, missiles et terrains pour l'attaque au sol.

Deux grands domaines applicatifs industriels utilisèrent la synthèse d'image sous un angle et avec des objectifs totalement différents:

**-a/ La CAO:** ou Conception Assistée par Ordinateur:

Pour l'aide à une meilleure représentation des objets dans le processus de conception.

**-b/ L'Audiovisuel:**

Donna naissance à ce qui fut appelé "Nouvelles Images", appliquées non seulement aux désormais classiques spots publicitaires, mais également aux

films, à la simulation environnementale, notamment au QUEBEC, et à la simulation scientifique de phénomènes physiques, ex: simulation d'une chambre de combustion de voiture, simulation du déploiement de panneaux solaires d'un satellite, représentation de virus, de protéines et de macro-molécules biologiques. (DUB)

#### a/La CAO:

La branche synthèse d'image de la CAO ne prit véritablement son essor que vers les années 1980 environ aux Etats-Unis: A cette époque, l'instrument de prédilection était la table traçante, associée à un "écran de visualisation", raccordé par ligne série à un mini-(ou maxi) ordinateur dont le volume, à performance égale, représentait au moins 30 fois celui de nos actuels ordinateurs personnels. Typiquement une configuration répandue était constituée de systèmes mini-ordinateurs 32 bits, raccordés à des écrans de visualisation à tube mémoire.

Ces machines étaient orientées vers la représentation des objets sous forme filaire, c'est à dire qu'il était impossible de générer des surfaces dotées de variations de luminosité et d'ombrage: D'ailleurs, l'eussent-elles permis, le temps de transfert sur une ligne série des informations graphiques correspondantes aurait de toute façon été prohibitif pour les utilisateurs: à 9600 Bauds, une image 512 par 512 points aurait nécessité un temps de transfert de  $512 \times 512 \times 3 / 960$  soit: 13 mn environ. Il était donc, à cette époque, impensable de faire sur ces terminaux des images de synthèse (par opposition aux actuelles stations de travail, dites "bitmap" dotées d'une unité centrale locale pouvant écrire directement dans une mémoire d'image).

Comme nous l'avons indiqué plus tôt, l'arrivée très progressive des écrans de travail fit peu à peu s'intégrer le mini-ordinateur *dans* l'écran graphique, dont l'aboutissement est la station de travail personnelle ou professionnelle actuelle. Sur ces nouvelles machines, le talent des chercheurs en synthèse d'image put s'exprimer, et aboutir à des algorithmes de rendu des images très performants en qualité, sinon en temps: Il faut en effet préciser que les contraintes de la CAO sont très différentes de celles de la simulation:

- Pas de temps réel impératif
- Peu de couleurs simultanément indispensables (256 est généralement bien suffisant)
- Peu d'effets spéciaux (textures, réflexions multiples, etc...)

Les systèmes d' Evans & Sutherland migrèrent donc peu à peu du mode vectoriel pur au mode raster (ou point d'écran) vers les années 86.

A cette époque apparurent les premières cartes additionnelles sur micro-ordinateur, permettant de faire pour 30 KF ce qu'une station de travail de 300 KF faisait 2 ans plus tôt.

Simultanément l'innovation capitale des systèmes dits "bitmap", dans lesquels le (ou les) processeur(s) principal(aux) virent la mémoire d'image dans le champ d'adressage de leur propre mémoire. Cette fonctionnalité nouvelle creusa rapidement dans les années 87-88 un écart décisif entre les systèmes bitmap dotés de composants intégrés spécifiques et ceux qui n'en disposaient

pas: l'absence d'effort de développement de composants graphiques spécifiques 3D se fit durement ressentir en France, et tandis que les constructeurs US ne cessaient d'améliorer leur performances tout en diminuant leurs prix grâce à ces composants 3D, les systèmes non bitmap et dénués de composants spécifiques, entièrement dépendants de la technologie disponible sur le marché ouvert, plafonnèrent en performances et en prix. L'arrivée des Transputers en 87-88 changea légèrement les choses, mais sans résoudre le problème du volume et donc du coût des parties vitales d'une station de travail graphique 3D: Générateur de facettes, contrôleur de mémoire d'image, multiplieur matriciel 4\*4, clipper.

Aux USA, ces composants d'importance vitale apparurent, et en particulier la "Geometric Engine" de James Clark: En effet, hormis leur coût d'étude élevé, les composants spécifiques 3D ne cumulent que des avantages:

- Fiabilité accrue
- Volume réduit dans un facteur considérable (1 composant VLSI pouvant remplacer de 100 à 200 composants du commerce)
- Coûts de fabrication unitaires moindres
- Protection totale du constructeur de la machine
- Avance par rapport à des concurrents non dotés du circuit
- Possibilité d'améliorer encore les performances, grâce aux progrès constants de l'intégration micro-électronique:

Les premiers composants de CLARK étaient en effet très lents: au début, ils n'effectuaient une multiplication entière qu'en 12 micro-secondes environ, et les stations qui en étaient équipées n'étaient au départ en 85 ni moins chères ni plus performantes que celles dotées de composants du commerce. Mais les choses évoluèrent très vite: Silicon Graphics vendit de l'ordre de 1500 machines par an à partir de 87, tandis que les machines sans composants spécialisés tels que CUBI 7 ou le BOSH FGS 4000 (basé sur les algorithmes à priorité et sur l'AMD 2900) ne dépassèrent pas les quelques dizaines d'unités vendues. (Le FGS 4000 valait environ 4 MF...).

#### **b/ L'AUDIOVISUEL:**

Depuis 1980 environ, le secteur audiovisuel eut connaissance des remarquables possibilités nouvelles offertes par les systèmes de visualisation temps réel utilisés dans les simulateurs. Il en résultait une assez grande confusion, les réalisateurs selon le jargon de la profession cherchant à "s'approprier" un outil qui n'était pas du tout adapté à leurs besoins, et qui de surcroît leur était totalement étranger.

L'on vit des "Vols de rêve", essais de transcriptions artistiques de vols effectués sur simulateurs. Malheureusement, les contraintes matérielles (coût des matériels de simulation, et prix corrélatif de la seconde d'animation produite, jointe à la rigidité d'utilisation de ces machines) conduisit à de nombreuses déceptions. Jusqu'en 85 environ la "saisie" d'objets 3D se faisait "à la main" c'est à dire facette par facette, en rentrant chaque coordonnée de sommets des objets à représenter. De plus, ces machines souffraient de rigidités inhérentes d'une part aux contraintes temps réel de l'application à laquelle elles étaient destinées - la



simulation- et d'autre part à l'utilisation d'algorithmes dits "à priorité" (LER 81) bien adaptés au cadre de la simulation. En résumé, par exemple, ces machines faisaient du "temps réel" et rien d'autre: C'est à dire qu'il était impossible de leur demander des effets spéciaux plus sophistiqués : texture, ombres portées, même au prix d'un accroissement du temps de calcul.

De plus, les systèmes à priorité étaient adaptés à la visualisation d'objets rigides: avions cibles, chars, paysages d'aéroports à l'exclusion d'objets en mouvement relatif les uns par rapport aux autres. Il était par exemple impossible de définir plusieurs objets solides, et de les animer en mouvement relatif les uns par rapport aux autres. Pour un générique de 6 lettres environ et après saisie de ces lettres par extrusion d'un contour rentré sur tablette graphique, les positions relatives des lettres entre elles devaient être définies rigidement, l'ensemble des six pouvant être déplacé.

A cette époque, hormis les décevants essais d'utilisation de machines de simulation pour l'Audiovisuel, la tendance était axée sur le développement de LOGICIELS de synthèse, inévitablement très lents, mais pouvant comporter des effets spéciaux de qualité, et une indépendance totale des mouvements des objets. L'on passait donc du temps réel simulation avec sa rigidité, mais son extrême performance en temps de calcul, à l'exact opposé: grandes performances en qualité de rendu des images, mais au prix d'un temps de calcul considérable. Ces caractéristiques sont modélisées par la formule:

$$C * T / A = \text{Constante} \quad (\text{voir annexe 6})$$

Ces 2 extrêmes n'avaient bien sûr qu'un avenir limité tant en CAO qu'en Audiovisuel: Un simple calcul montre que si une image nécessite une heure de traitement informatique, il n'est possible de produire que:

$$24 * 365 / 60 = \mathbf{5.84 \text{ mn de film animé par AN.}}$$

De plus le chiffre d'une image/heure est très optimiste, la réalité étant plutôt proche de 2 ou 3 voire plus. Un coût minimum de 1MF la minute soit 17 KF/s était donc industriellement nécessaire pour équilibrer le coût du matériel, et du personnel. L'on comprend donc que malgré l'extraordinaire fascination que suscitaient ces "Nouvelles" images au sein du monde de l'audiovisuel, bien peu pouvaient se les offrir.

Dans la période 80-85 des productions publicitaires à grand budget étaient présentées au SIGGRAPH, souvent réalisées sur de très puissantes machines ou calculées images par images sur des centaines de minisordinateurs.

Les principaux centres de recherche et de production basés sur ce type d'images périclitèrent en quelques années: Digital Production fut racheté en 86 par OMNIBUS qui elle-même fit faillite, après avoir au passage racheté Robert Abel & Associates après son dépôt de bilan...

La cause principale de ces chutes brutales fut tout simplement l'essor à partir des années 85 des stations de travail, qui incorporèrent de plus en plus de **FONCTIONS CABLEES** bien plus performantes que le meilleur des logiciels. Il était possible, dès 80 (LER 80) de gagner un facteur de 1 à 1000 sur le temps de calcul des images ( ce que les simulateurs faisaient depuis 1975 environ, mais avec les rigidités que nous avons évoquées plus haut).

C'est pourquoi, à partir des années 85 disparurent pour ce type d'usage les super-calculateurs, laissant la place aux stations de travail, dérivés économiques des imposantes machines de simulations. Citons vers 86 la sortie de stations de travail telles que l'IRIS de SILICON GRAPHICS et CUBI 7 (LER 86) Dans l'équation économique apparurent les éléments nouveaux suivants:

- L'importance du hardware (matériel) et de son architecture pour accélérer dans des proportions considérables les temps de calculs
- La nécessité d'abaisser les coûts des postes de travail affectés aux infographistes opérateurs.
- L'importance de la modélisation des objets, jusque là réalisée quasiment "à la main", bien que de lourds et coûteux logiciels de modélisation soient apparus pour les besoins de la modélisation des objets pour les simulateurs ou la CAO.
- Nécessité d' "ACCELERATEURS DE RENDU " disposant de fonctions graphiques de calcul 3D câblées ou micro-programmées.

## 1-2/INTRODUCTION:

### 1-2-1/LE CONTEXTE GÉNÉRAL ACTUEL DE LA SYNTHÈSE D'IMAGES- DEFINITIONS TECHNIQUES

La synthèse d'image peut se définir sur le plan technique comme "l'ensemble des techniques de visualisation sur écran TV de solides 3D animés ou non avec rendu réaliste coloré de leur état de surface, ces solides pouvant exister réellement ou non". La synthèse d'image constitue donc l'art de "voir ce qui n'existe pas" (ou n'existe pas encore); Elle est donc au coeur du processus de conception tant industrielle qu'artistique des objets qui nous entourent.

Elle tend de plus en plus à supplanter les techniques traditionnelles consistant à réaliser une maquette solide réelle de l'objet à construire, de vérifier ses propriétés, puis de mettre en fabrication l'objet proprement dit.

Cette étape est de plus en plus lourde à mettre en oeuvre, dans la mesure où les techniques d'évaluation de performances des objets industriels s'affinent et reposent sur les techniques informatiques de modélisation des phénomènes physiques.

Pour un certain nombre d'objets industriels, ceci constitue déjà une réalité: la conception se fait entièrement sur écran: la fabrication est couplée au système de conception assistée par ordinateur. Cette tendance se renforce chaque jour davantage: En technique de construction automobile, comme dans beaucoup d'autres industries, des usines sans personnel d'exécution pourraient être construites, abstraction faite bien entendu des problèmes humains qu'elles ne manqueraient pas d'engendrer.

**1-2-2/ POSITION DU PROBLEME DE CHOIX COUT-QUALITE-TEMPS**

Paul Valery disait, bien avant l'existence de la synthèse d'image:

*"Tout ce qui est simple est faux, et tout ce qui est compliqué est inutilisable".*

Cette phrase, un demi-siècle plus tard, s'applique particulièrement bien à la création électronique des images: Elle illustre le problème scientifique et économique fondamental du domaine d'admissibilité du modèle algorithmique d'un phénomène donné. Prenons par exemple les lois de Newton sur la mécanique: ces lois ont un domaine d'admissibilité bien connu, dans la limite des vitesses très faibles par rapport à la lumière. Dans le cas de vitesses plus grandes, les lois de la relativité doivent s'appliquer.

Ces dernières mettent en jeu des formules sensiblement plus complexes que les lois de Newton, qui pourtant suffisent dans bien des cas. Entre ces deux extrêmes (lois simples mais approximant le phénomène étudié, versus lois précises mais complexes et coûteuses à mettre en oeuvre), les choix s'opèrent de façon variée:

- Le chercheur a naturellement tendance à s'intéresser d'emblée aux secondes, en privilégiant la rigueur au détriment de la recherche de la simplicité et de la réduction des coûts.

- L'industriel qui implante dans ses matériels des algorithmes tente avant tout d'en minimiser non seulement les coûts d'intégration en complexité logicielle et matérielle, mais aussi les coûts en temps de calcul, ce qui, du point de vue du chercheur est assez secondaire.

- Enfin l'utilisateur final a tendance à vouloir l'ensemble des avantages (qualité, précision, coût minimal, temps de calcul réduit) simultanément.

Dans le domaine complexe de la synthèse d'images tri-dimensionnelles, le nombre d'opérations nécessaires au calcul d'une seule image est considérable : Pour le calcul du rendu d'une image ( comportant 2 sphères et un plan ) selon la méthode du lancer de rayons sur un ordinateur personnel de type PC XT, le temps de calcul est de 8000 heures (**WHITTED 86**: Cours du SIGGRAPH p 5 volume 12), s'il ne dispose pas d'opérateur flottant sur son calculateur.

Dans son article, WHITTED présente les chiffres suivants pour la scène décrite ci-dessus:

IBM PC 8088	:	960 heures
IBM PC XT 8088		
avec 8087( coprocesseur mathématique)	:	8 heures
SUN 2-120	:	8 heures
SUN avec accélérateur flottant	:	2 heures
SUN 3-160	:	0.5 heures
Micro VAX II	:	0.5 heures
CELERITY	:	0.3 heures

Tableau 1

Il est à souligner, toujours du même auteur que si cet ordinateur comporte un circuit VLSI d'accélération du calcul flottant, ce temps passe à 8 heures. Sur des mini-ordinateurs de type VAX, Micro-VAX ou SUN, le temps passe dans le meilleur des cas à 1h.

Or le Micro-VAX II comme le SUN effectuant 1 million d'opérations scalaires par seconde (Source: Revue Ordinateurs du 22 fév 88) et que le calcul de l'image décrite ci-dessus, toujours d'après WHITTED s'effectue sur ces calculateurs en 0.5 h, nous pouvons déduire que le nombre d'opérations nécessaires pour l'affichage de l'image composée des éléments décrits ci-dessus est de:

$1\ 000\ 000 \times 0.5 \times 3600 = 1800$  millions d'instructions soit 1.8 Giga-opérations. ( pour 2 sphères et 1 plan). (voir annexe 7)

### 1-2-3/ LE PROBLEME ACTUEL DE LA SYNTHÈSE d'IMAGES

Il se trouve que le défi actuel de la synthèse d'images est précisément là: Il consiste à améliorer la technologie et l'architecture des machines afin d'en accroître les performances en termes de :

- Durée de calcul
- Qualité des images

En effet, nous avons précédemment montré (WHITTED 86) que le rendu en 3D d'une scène comportant 1 plan et 2 sphères demandait 8 heures sur un IBM PC: Ce délai important limite de façon considérable les performances du couple opérateur-machine, le premier étant dans l'impossibilité d'attendre de tels délais pour interagir. De plus, et c'est là l'argument le plus fondamental, le volume annuel de production basé sur une machine produisant 1 image par heure (ce qui est le cas des sociétés de production actuelles) ne peut excéder 5 mn environ par an à 25 images/s ( Voir la production actuelle des principaux producteurs français ou étrangers d'images de synthèse.)Deux directions simultanées d'amélioration peuvent être envisagées:

- Proposer un coût du système identique voire si possible inférieur avec des performances supérieures.
- Ou de façon corrélative proposer à des coûts moindres des performances sensiblement égales.

Voici deux exemples typiques à cet égard:

MATROX propose en 87 des cartes additionnelles sur PC à un coût dix fois inférieur à un système SILICON GRAPHICS de 85 doté des mêmes performances. (ref: doc technique MATROX).

SILICON GRAPHICS propose le système 4D au même prix que le 3130 avec des performances sensiblement multipliées par 5 en 2 ans. (passage de 200 facettes en temps réel à 1000 environ).

Cette quête de la qualité et du temps est en effet fondamentale pour accéder à une large diffusion de ce genre de technique. (cf WHITTED 86 SIGGRAPH DALLAS USA p 5 et suivantes)

*Pour donner un élément de comparaison, il serait difficilement concevable de concevoir une machine de traitement de texte dont la génération de caractère ne pourrait suivre le rythme de la main de l'homme.*

L'on conçoit aisément qu'une telle machine resterait d'un intérêt très réduit, même si les caractères générés étaient de bonne qualité.

C'est pourtant le cas en synthèse d'image, où l'opérateur ne peut contrôler visuellement ses images que plusieurs minutes voire plusieurs heures après qu'il les ait demandées au système. Cet aspect très technologique de la synthèse a à l'heure actuelle d'importantes répercussions sur la méthodologie d'utilisation de cette technique: En effet, il a fallu trouver des palliatifs à la contrainte du temps de calcul du rendu de l'image:

Une méthode classique a consisté et consiste encore à générer une image en mode vecteur, ne comportant que les arêtes des objets à représenter à l'exclusion de toute surface ombrée et de tout éclairage. (Les références sont nombreuses, pratiquement tous les constructeurs de matériels et de logiciels de CAO proposant ce type de représentation: COMPUTERVISION, TEKTRONIX, IMI, EVANS & SUTHERLAND, DIGITAL, MATRA-DATAVISION). La génération du dessin (terme employé dans ce cas à la place de l'image) est réalisable sur les matériels du commerce avec des temps de calcul compatibles avec le temps réel. La raison en est que le nombre de points à générer sur l'écran est considérablement plus réduit dans le cas du mode vecteurs que dans le cas du mode à balayage TV.

(Ce nombre est en effet proportionnel à la somme des longueurs des arêtes en mode vecteur, à la surface des objets sur l'écran en mode à balayage).

Les systèmes commerciaux permettent en effet des affichages de l'ordre de 50000 vecteurs en temps réel, (voir les documents techniques des systèmes IMI et EVANS & SUTHERLAND) alors que pour des prix équivalents, seules 1000 facettes polygonales planes peuvent être affichées (voir notice SILICON GRAPHICS).

Pour situer le niveau de complexité des images, une image de 1500 à 2000 facettes permet de représenter par exemple un aéroport avec une bonne finesse, mais sans détails très fouillés (ex: huisserie de fenêtres, etc...)(voir doc techniques des constructeurs susnommés).

Un dessin de 50000 vecteurs ( donc en mode filaire) permet de représenter un objet tel qu'un avion avec une très grande précision.

Bien entendu, tous ces chiffres sont des indications subjectives, le degré de détails nécessaires à une bonne appréciation d'une image pouvant varier d'un individu

ou d'une société à l'autre. Dès 86, des systèmes capables d'animer en temps réel des objets de cette complexité étaient déjà disponibles.

Avec un tel système, couplé à un ordinateur connecté à un terminal à mémoire d'image, l'opérateur peut vérifier rapidement la cohérence des solides qu'il a modélisés ainsi que leurs trajectoires d'animation, et lancer ensuite avec une certaine sécurité les longs calculs de rendu d'image de qualité avec représentation de surfaces et d'éclairages. Cette méthode est actuellement la seule utilisée dans le monde pour la génération d'images de synthèse hors simulation. (Il est à remarquer que depuis 2 ans environ, les constructeurs de systèmes graphiques proposent des représentations avec ombrage lisse pour la visualisation grossière des objets).

Il est à souligner que le recours à l'image vecteur est en fait imposée par les limites de la technologie, celle-ci étant actuellement encore incapable de représenter en temps réel à un coût compatible avec celui d'une station personnelle des images ombrées avec les effets spéciaux tels que les ombres portées ou les réflexions multiples.

Si l'on désire ce type d'effets, les matériels utilisés:

-Sont beaucoup plus chers

-Et dédiés à un type d'application où la base de donnée des objets à représenter ne change que rarement.

Il est clair que l'avènement de machines de rendu de qualité ET temps réel changerait beaucoup ce type de méthode. Des conséquences induites s'exerceraient également sur la saisie par surfaces mathématiques ou par numération directe actuellement utilisée.

Le propos de cette thèse est précisément d'avancer des solutions nouvelles dans ces 2 domaines :

*-Modélisation des objets*

*-Architecture de machines de rendu des images de haute performance.*

L'ensemble des solutions anciennes pour ce type de problème étant étudiées aux chapitres 2 pour la modélisation et au chapitre 3 pour les machines de rendu.

Les solutions originales proposées sont décrites respectivement aux chapitres 4 et 5.

Pour cela, nous passerons tout d'abord en revue les différentes méthodes de modélisation, en proposant ensuite une technique nouvelle.

Puis nous aborderons les choix d'architecture de machines parallèles, en proposant une solution originale.

Enfin la conclusion portera sur la combinaison des deux, c'est à dire en quoi l'augmentation de performances pourra améliorer globalement la modélisation des objets, et en conséquence les capacités de la technique de synthèse des images dans son ensemble.

## 2/LES DIFFÉRENTES MÉTHODES DE MODELISATION: ETUDE COMPARATIVE DES SOLUTIONS ACTUELLES

Le problème de la modélisation consiste à définir pour le système informatique une structure de base de données permettant d'approximer un objet ou un ensemble d'objets représentant la scène à visualiser sur l'écran de l'opérateur.

Il est clair que l'exhaustivité des informations contenues dans un objet est impossible à définir de façon complète pour la machine. De plus, en fonction de l'application, certains paramètres peuvent être inutiles:

*Exemple 1:* en audiovisuel, les informations définissant les centres de gravité, les moments inertiels d'un objet important peu. Au contraire, en construction mécanique (CAO), ces mêmes paramètres sont primordiaux.

*Exemple 2:* en CAO, les aspects de couleur, de texture des matériaux sont d'importance relative, l'essentiel étant d'obtenir une bonne intelligibilité de l'image. Au contraire, en audiovisuel, les informations de couleurs, de texture, l'aspect de surface en général, ainsi que la forme sont essentiels.

Les différentes méthodes de représentation des objets peuvent être classées en 4 grandes familles:

- Les représentations par vecteurs
- Les représentations par surfaces (Boundary representation)
- Les représentations par maillage volumique
- Les représentations par construction de solides géométriques ou CSG (Constructive Solid Geometry)

Rappelons pour mémoire certaines modélisations particulières, souvent étroitement associées à un type donné de saisie:

- Les représentations par nuage de points à partir d'une saisie directe en 3D, généralement obtenues à partir du résultat brut de modélisation par laser.
- Les représentations par voxels issus de sources de numérisation médicales (scanners, et Résonance Magnétique Nucléaire: RMN)
- Les représentations par quadrees en 2D et par octrees en 3D.

Ces différents types de représentation, très liées au domaine médical ne seront pas étudiées. Elles ne permettent pas, en effet d'atteindre la généralité de construction et de modélisation des 4 précédentes que nous allons maintenant détailler: par exemple, il est pratiquement irréaliste de faire tourner un octree dans l'espace en raison du temps de calcul que nécessite cette opération. Notons cependant que des passerelles de conversion existent entre ces 3 derniers types de représentation et les 4 précédents.

### 2-1/ LA MODELISATION PAR VECTEURS

Ce mode de représentation est le plus ancien:

Très utilisé en CAO, il présente actuellement un certain nombre d'inconvénients. Sa simplicité en fait un outil pratique pour les objets simples. Il consiste à définir:

- La liste des points de l'objet
- La liste des vecteurs le constituant.

Ce mode de représentation pose de nombreux problèmes pour les opérations topologiques.

Il ne donne en effet aucun élément sur la cohérence topologique du modèle.

## **2-2/ LA MODELISATION PAR SURFACES**

On distingue, pour ce type de modélisation, 2 types principaux:

La modélisation par surfaces élémentaires planes, constituées en général par des facettes.

La modélisation par surfaces mathématiques, qui autorise une plus grande souplesse de construction des solides.

### **2-2-1/ MODELISATION PAR SURFACES ELEMENTAIRES PLANES**

L'objet à numériser est décomposé en un ensemble de facettes polygonales planes, dont on stocke les informations suivantes:

- Liste des coordonnées spatiales des sommets des objets
- Liste de pointeurs sur les points décrivant les faces des objets

*NOTA:* Des variantes existent consistant à définir:

La liste des arêtes des polygones

La liste des facettes avec les coordonnées de chaque point.

Des transformations simples permettent de passer de l'un à l'autre de ces 3 modes de représentation:

Une amélioration de cette méthode consiste à proposer à l'utilisateur des ensembles de facettes prédéfinis, en général des solides mathématiques simples:

Prismes, Sphères, cylindres, tores, surfaces de révolution, etc...

Le logiciel de modélisation doit comporter également des algorithmes d'intersection de ces différents volumes: en fait, il s'agit d'un calcul d'intersection de facettes à facettes.

Certains logiciels possèdent une modélisation mathématique de l'intersection entre 2 solides élémentaires; ex: cylindre-sphère, etc..., (cas de PADDLE 2), mais ceci constitue un cas isolé, étant donné la difficulté de composer des assemblages d'objets déjà intersectés entre eux.

### **2-2-2/LES SURFACES MATHÉMATIQUES**

-Les surfaces mathématiques (ou patches) constituent la forme la plus élaborée de modélisation mathématique. De grands logiciels de CAO modernes, existants ou en cours de développement reposent sur ce type de représentation. Des revues spécialisées telles que COMPUTER AIDED GEOMETRIC DESIGN y consacrent exclusivement leurs articles de recherche.

Ce type de modélisation consiste à définir un polynôme de degré 3, donnant les coordonnées paramétriques  $x(s,t)$ ,  $y(s,t)$  et  $z(s,t)$  en fonction des paramètres  $s$  et  $t$  d'un maillage défini sur la surface elle-même.  $s$  et  $t$  sont des réels définis sur l'intervalle 0-1.



L'intérêt majeur de cet algorithme est de permettre l'introduction de contraintes aux limites: par exemple, la surface doit passer par une courbe donnée et être tangente à un ensemble de plans tangents donnés.

Nous limiterons ce domaine très vaste aux 3 principaux types de surfaces: Les surfaces d'HERMITE, de BEZIER et de COONS. et nous citerons les grandes tendances actuelles de recherche dans ce domaine.

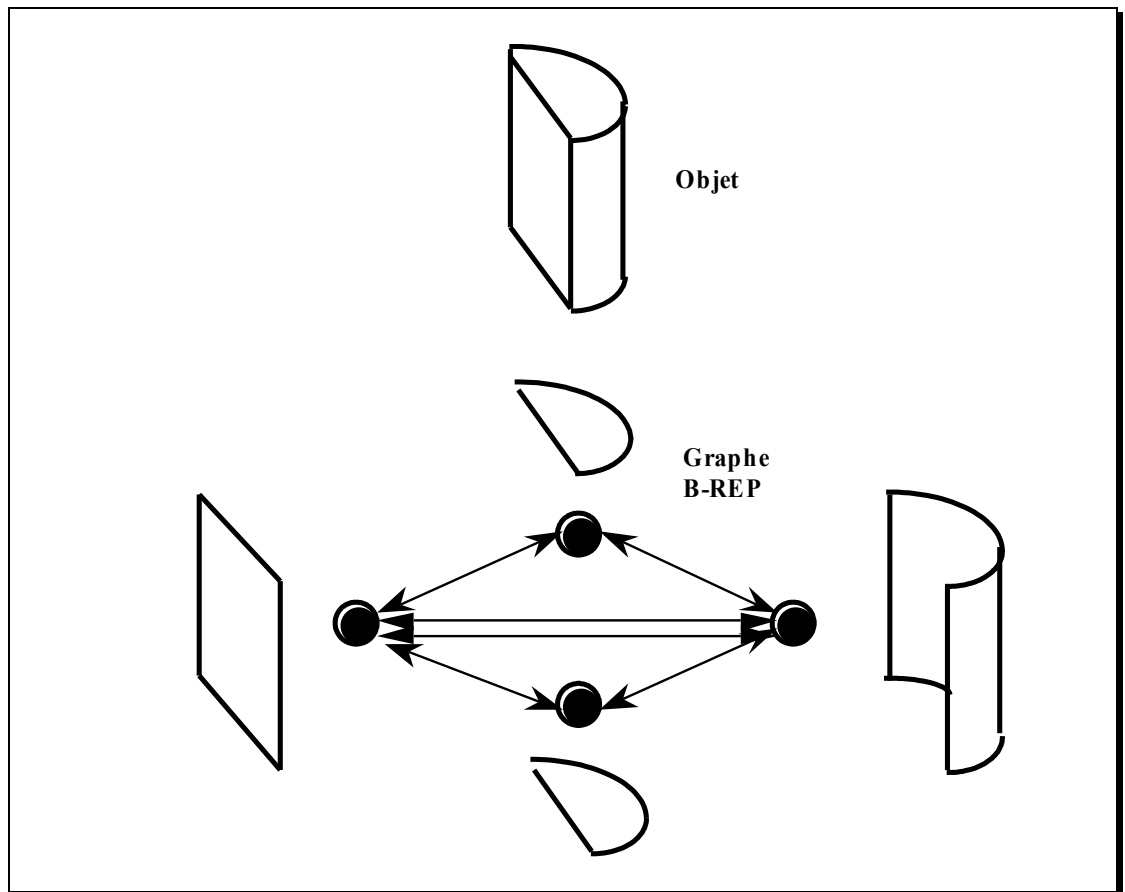


Fig 1: Exemple de représentation par surfaces limites ou Boundary Representation (B-Rep)

Examinons maintenant les différents types de modélisation de surfaces gauches:

### 2-2-3/LES SURFACES d'HERMITE

Elles se définissent de la façon suivante:

Soit un ensemble de 4 points de l'espace par lesquels l'on souhaite faire passer une surface  $S$ . L'on dispose par ailleurs des pentes selon 2 directions orthogonales repérées par les abscisses curvilignes  $s$  et  $t$ .

Le problème consiste à déterminer les coefficients  $a_{ij}$  en fonction des conditions aux limites que doit remplir la surface,  $a_{ij}$  permettant en tout point de coordonnées  $s, t$  de connaître le point courant de la surface  $S$  par ses 3 coordonnées:

$$x(s,t) = a_{11}s^3t^3 + a_{12}s^3t^2 + a_{13}s^3t + a_{14}s^3 + a_{21}s^2t^3 + a_{22}s^2t^2 + a_{23}s^2t + a_{24}s^2 + a_{31}st^3 + a_{32}st^2 + a_{33}st + a_{34}s + a_{41}t^3 + a_{42}t^2 + a_{43}t + a_{44}$$

$$y(s,t) =$$

$$z(s,t) =$$

Soit, sous forme matricielle:

$$x(s,t) = S C_x T_t \tag{1}$$

Avec:

$$S = (s^3 \ s^2 \ s \ 1) \text{ et } T = (t^3 \ t^2 \ t \ 1)$$

C étant la matrice des coefficients  $a_{ij}$ . Dans le cas des surfaces d'Hermite :

$$C_x = H Q_x H_t$$

donc:

$$x(s,t) = S H Q_x H_t T_t$$

H étant la matrice d'Hermite:

$$H = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$H_t$  étant la matrice transposée.

(Voir livre de **FOLEY and VAN DAM** p 516: FOL 83).

Cette matrice est obtenue en écrivant que le point courant de coordonnées  $x,y,z$ , passe par les points définissant la surface, avec les tangentes données dans les directions de  $s,t$ .

$Q_x$  dépendant des conditions aux limites. Dans le cas des surfaces d'Hermite, les conditions aux limites sont définies par 4 courbes passant par 2 points et tangentes à 2 vecteurs donnés en ces 2 points.

### 2-2-4/SURFACES DE BEZIER

(BEZ 87)

La structure mathématique est identique à celle des surfaces d'Hermite, à la différence près que les courbes aux limites sont définies par 4 points, les 2 premiers et les 2 derniers définissant les tangentes à l'origine.

La matrice correspondante de Bézier a la forme suivante:

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & 3 & 0 & 0 \\ 0 & 0 & -3 & 3 \end{pmatrix}$$

Il suffit de remplacer H par B dans l'expression (1).  
 B est obtenue en effectuant les mêmes opérations que pour H.

(FOLEY & VAN DAMM p 519: FOL 83)

**2-2-5/SURFACES B-SPLINE CUBIQUES**

Ces surfaces sont identiques dans leur principe aux deux précédentes, les 4 points des 4 courbes aux limites étant approximés au mieux par la surface.

$$\mathbf{Bs} = 1/6 * \begin{pmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}$$

(FOLEY & VAN DAM p 521)

Il suffit donc de remplacer H par Bs dans l'expression (1) pour obtenir l'équation de la B-spline. Bs est obtenue en effectuant les mêmes opérations que pour H.

**2-2-6/ COMPARAISON CRITIQUE DE CES DIFFÉRENTES FORMES DE MODÉLISATION**

Chacune de ces courbes présente un intérêt en fonction des circonstances:

La forme d'Hermite lorsque l'on veut faire coïncider ces courbes avec une surface pré-définie.

La forme B-SPLINE est intéressante lorsque l'on veut approximer au mieux un nuage de points et lorsque l'on veut conserver la continuité de la dérivée seconde.

Les formes de BEZIER et les B-SPLINE sont recommandées pour les manipulations interactives d'objets, puisqu'elles ne comportent que des points de définition et non des tangentes, plus difficiles à définir dans l'espace. L'étude des courbes mathématiques fait actuellement l'objet de travaux de recherche très importants dans le monde. (voir revue COMPUTER AND GRAPHICS ou COMPUTER AIDED GEOMETRIC DESIGN ).

**2-3/ LA MODELISATION PAR MAILLAGE VOLUMIQUE**

Ce type de représentation est surtout utilisé en calcul de structures. Il permet en particulier d'effectuer tous les calculs liés aux solides: évaluation des volumes, des masses, des centres de gravité, des moments d'inertie, etc..

Cette modélisation repose sur une décomposition des volumes en tétraèdres élémentaires : Elle n'a qu'un intérêt limité en synthèse et pour tout problème de représentation, en dehors des possibilités évoquées ci-dessus. En fait, l'on a généralement recours, pour la représentation des objets, à des surfaces mathématiques décomposées en triangles élémentaires. (Tessellation problem aux USA ou décomposition en formes polygonales élémentaires). (SIG 88).

## 2-4/ LA MODELISATION PAR CONSTRUCTION SOLIDE (CSG)

L'idée de base de ce type de modélisation repose sur deux éléments:

- Le recours à des solides géométriques élémentaires (sphères, cônes, cylindres, etc...)

- L'utilisation d'opérateurs booléens entre ces solides.

L'utilisateur construit un graphe de son objet à l'aide de ces opérateurs. Mais le système ne calcule pas effectivement les différentes intersections, (du moins dans le cas d'un affichage par lancer de rayons) mais se contente de stocker l'ensemble des solides élémentaires utilisés dans le graphe de l'objet.

Ce type de modélisation est actuellement en net recul, en raison du fait suivant: La CSG s'appuie par définition sur l'ensemble des êtres mathématiques de base (cubes, sphères, cylindres, etc...) permettant, à l'aide d'opérations booléennes ("+" ou fusion, "." ou intersection, "-" (moins) ou différence) de construire le modèle demandé.

La construction CSG est bien adaptée à des solides de type mécanique, décomposables en formes mathématiques élémentaires (sphères, cônes, cylindres).

Cependant, la difficulté de manipuler des formes mathématiques plus complexes que les principaux solides élémentaires, (cas des surfaces mathématiques) est liée au fait que les intersections de volumes ne sont pas effectivement calculées sous forme d'arêtes communes inter-facettes.

En conséquence, la représentation interne est d'autant plus lourde que le modèle final s'écarte des formes élémentaires citées plus haut et qu'il conserve l'ensemble des volumes de base entrant dans la définition du solide. L'on peut, à la rigueur, avoir recours à des surfaces mathématiques ou patches mais dans ce cas, les calculs d'intersection rayons lumineux-primitives, dans le cas de la méthode d'affichage par lancer de rayons devient très lourde puisque nécessitant de nombreux calculs, ainsi que des comparaisons.

Un autre inconvénient plus critique réside dans le fait que la modélisation CSG est très mal adaptée à des objets déformables, de type plastiques, peaux, qui sont de plus en plus utilisés tant en audio-visuel qu'en CAO.

L'on en revient à la modélisation par frontières (ou Boundary representation). La méthode CSG peut avoir quelques avantages dans le cas de formes simples, d'exemples d'école, ou dans le cas où l'on veut avoir des informations liées aux volumes. (centres de gravité, masses d'inerties, etc...), mais que l'on peut également obtenir par maillage à partir de tétraèdres (voir plus loin).

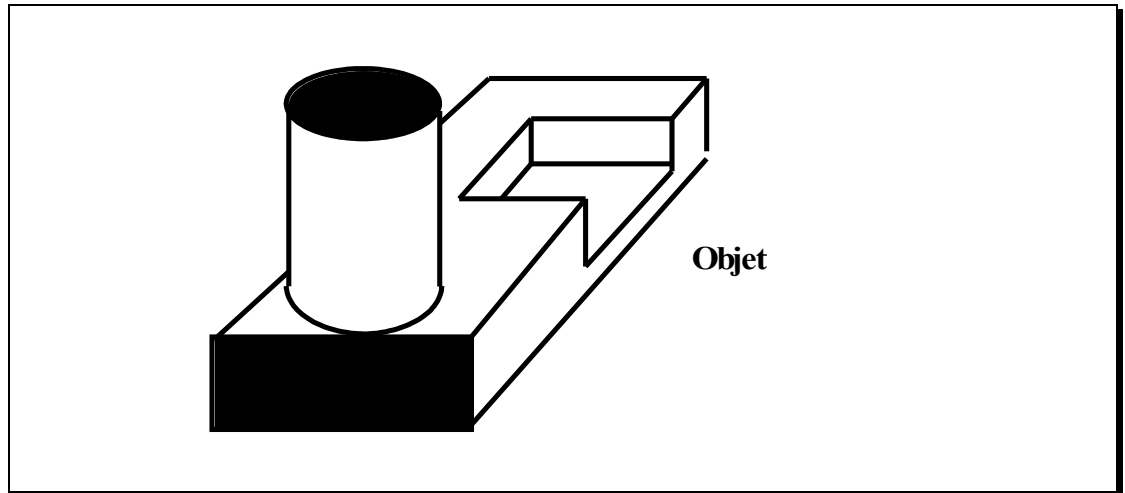


Fig 2: Solide à modéliser

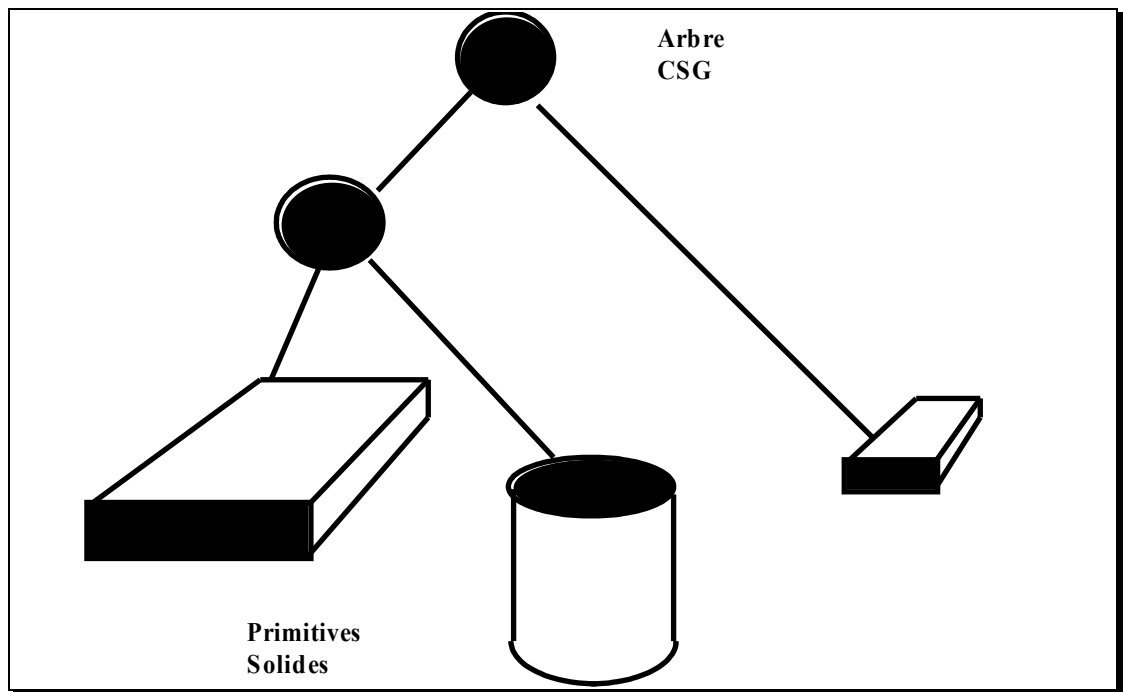


Fig 3: Exemple d'objet modélisé en CSG avec son graphe correspondant

### 2-5/PASSAGE À D'AUTRES MODES DE REPRÉSENTATIONS

Il est bien entendu possible de passer de la représentation CSG à une autre, à condition de calculer effectivement les intersections des solides élémentaires .

L'on peut ainsi passer de la CSG à la représentation par vecteurs, ou surtout à la représentation par surfaces après avoir effectué cette transformation. Celle-ci est en fait très lourde, puisqu'elle constitue la base de tout modéleur solide géométrique évolué.

Ceux-ci, en effet utilisent constamment cet outil mathématique d'intersection de volumes à chaque fois que l'opérateur assemble deux solides par l'une des opérations

booléennes définies ci-dessus. La différence réside dans le fait que le modeleur CSG garde trace des solides antérieurs (à partir des plus simples) alors qu'un modeleur surfacique ne conserve que le solide final avec l'ensemble des intersections calculées.

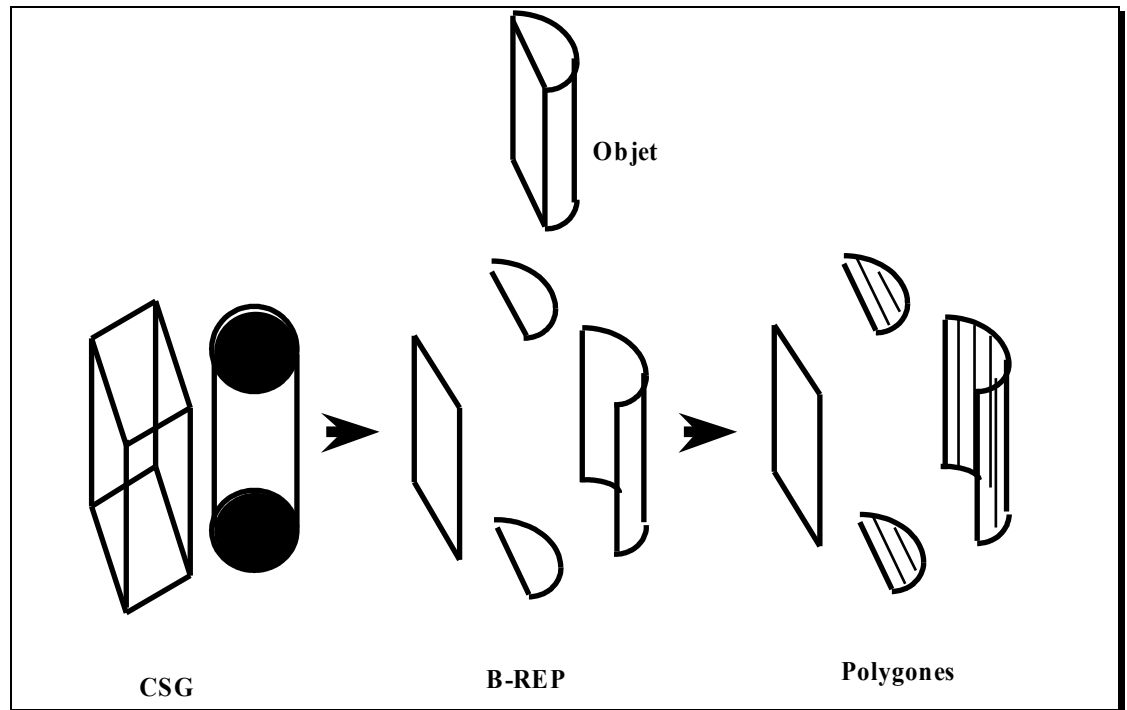


Fig 4: Exemple de passage de la CSG à d'autres modes de représentation

## 2-6/PAR NUMERISATION DIRECTE D'OBJETS

Dans ce qui précède, on entend par "modèle" une structure géométrique permettant à la machine d'obtenir la connaissance géométrique d'un objet réel. Lorsque cet objet est trop complexe, ou se réduit difficilement à un ensemble de formes géométriques (cas d'un visage, d'un téléphone, d'un objet complexe, etc...) l'on peut avoir recours à la numérisation directe d'un objet: le "MODELE" obtenu est alors un ensemble de points, qu'il importe donc de transformer en un modèle géométrique tel que la machine puisse en faire la synthèse.

L'esprit de la méthode consiste à numériser un objet existant, le plus souvent complexe, et à le retraiter ensuite à l'aide d'un modeleur géométrique qui en déduira la structure géométrique complète: surfaces mathématiques, facettes, etc....

Ce procédé est intéressant pour la saisie de formes non directement réductibles à des solides élémentaires de type CSG.

ex: corps humain, pièces mécaniques comportant de nombreuses surfaces gauches, bâtiments, etc...

La saisie permet alors de numériser l'objet et de le retraiter avec des méthodes de CAO.

Les principaux systèmes de ce type reposent sur les méthodes suivantes:

### Numérisation mécanique

**Numérisation par mesure de position d'un stylet 3D** ou d'un objet repérable par tout moyen sonique, magnétique, ou infra-rouge:

**Numérisation par laser:** .

**Numérisation par stéréoscopie:**

### 2-6-1/NUMERISATION MECANIQUE

Cette méthode est la plus ancienne:

Elle consiste à utiliser des systèmes articulés à 2 ou 3 dimensions. Ces systèmes sont munis de palpeurs qui sont mis en contact avec la pièce à numériser. La précision est très grande, mais la durée de mesure est considérable: Cette méthode est utilisée en métrologie de pièces mécaniques professionnelles, pour vérifier les cotes d'un objet fini.

La modélisation par système mécanique est extrêmement longue. Une pièce mécanique nécessite couramment 2 jours pour la numérisation de 2000 points seulement. La méthode sert en fait, le plus souvent, de vérification métrologique plus qu'à la modélisation de solides.

### 2-6-2/ NUMERISATION PAR MESURE DE LA POSITION d'un STYLET

Deux systèmes conçus aux USA existent actuellement sur le marché:

-Un système sonique développé par Science Application Corporation (S.A.C.)

-Un système magnétique développé par MAC DONNEL DOUGLAS (POLHEMUS).

*-Le système sonique de S.A.C.* comporte 4 microphones disposés sur un panneau de un mètre carré environ. L'opérateur dispose d'un stylet muni à son extrémité d'un éclateur produisant une étincelle très ponctuelle (moins d'un demi millimètre) dont le bruit est capté par les 4 microphones après un délai de propagation dans l'air de l'onde sonore.

Le procédé repose sur la numérisation par localisation SONIQUE d'un point: Un appareil électronique, par la mesure des 4 temps de propagation déduit la position spatiale x, y, z du stylet par calcul trigonométrique et par suite les coordonnées d'un point de l'objet à numériser, si le stylet est contre celui-ci.

Ce système permet non seulement de saisir des points à la surface d'objets, mais encore de numériser de trajectoires de solides en mouvement. Il suffit pour cela d'attacher le stylet avec son éclateur au mobile (qui peut également être constitué par la main de l'opérateur).

*AVANTAGE:* méthode interactive semi-automatique relativement rapide. Elle permet un contact opérateur-objet 3D à modéliser exceptionnel.

*INCONVENIENT :* L'éclateur doit toujours être visible des 4 microphones: En conséquence, la numérisation de trous dans l'objet est impossible. L'on peut tourner la difficulté en utilisant non plus un, mais trois éclateurs répartis aux sommets d'un triangle équilatéral supportant en son centre une tige de longueur parfaitement connue. La localisation spatiale des 3 éclateurs permet de déduire la position du repère lié au triangle dans l'espace, donc la position de l'extrémité de la tige, que l'on peut introduire sans difficulté dans une partie concave de l'objet. Cependant la précision ne peut excéder le demi-millimètre, et le système est très sensible aux parasites sonores ambiants.

**-Le système MAC DONNELL:** un champ magnétique est généré dans l'espace et mesuré à l'aide d'une petite bobine d'induction; Les paramètres recueillis permettent de déduire la position x,y, z du centre de la bobine ainsi que les coefficients directeurs d'un repère lié à celle-ci.

*Avantage:* L'objet peut comporter des concavités. La méthode sort également les paramètres directeurs du stylet.

*Inconvénient:* L'objet à numériser ne doit pas être ferro-magnétique.

### 2-6-3/PROCEDE PAR CAMERA INFRA-ROUGE:

Une variante de ce procédé est constituée par les éléments suivants: Le stylet et son éclateur sont remplacés par de petits cones en plastique blanc, que l'on attache à la surface de l'objet à numériser. Un ensemble de 2 à 3 caméras infra-rouge capte l'image des cones et il est aisé de déduire par triangulation les coordonnées de chaque cone. Ce système est principalement utilisé pour la capture de mouvements plus que pour la numérisation d'objets fixes. Il a été en particulier utilisé pour la numérisation du mouvements d'athlètes sportifs ou d'animaux.

### 2-6-4/PAR SYSTEME LASER

Plusieurs types de systèmes rentrent dans cette catégorie:

**Le système de l'ENST,**\_\_consiste à illuminer l'objet à saisir par un faisceau laser plan.L'objet mobile sur un plateau à axe vertical est illuminé selon un méridien. L'objet est placé sur un plateau mobile pouvant tourner autour d'un axe vertical. Une caméra vidéo est dirigée vers l'objet et analyse pour chaque position de l'objet en rotation (par exemple tous les degrés) la réflexion du faisceau laser sur celui-ci.

On obtient en sortie un nuage de points dont la précision dépend:

- Du pas de rotation de l'objet
- De la définition de la caméra.

*AVANTAGES :* La saisie est rapide (de l'ordre de la minute)

*INCONVÉNIENTS:* La précision est liée à la définition de la caméra: 512 points, pour rester dans des gammes de matériels relativement économiques.

### **Le système de l'INRIA:**

Celui-ci utilise un faisceau laser linéaire. L'analyse de la réflexion est faite par des barettes CCD dont la définition peut aller jusqu'à 2000 points et plus. La table supportant l'objet peut se déplacer en rotation et dans le sens vertical. L'analyse se fait donc par rotation de l'objet joint à un mouvement de translation verticale de bas en haut.

Le domaine de numérisation est de l'ordre du quart de m3.

Le résultat obtenu est un nuage de points.

*AVANTAGES:* Précision accrue par rapport au système ENST

*INCONVENIENTS:* Coût plus élevé; temps de numérisation beaucoup plus long (de l'ordre de l'heure)



Ce système a été notamment utilisé pour la saisie de statues ou de visages; Il peut aussi s'adapter à la saisie d'objets complexes tels qu'une voiture ou un char.

***Le système du CNRC: (OTTAWA-CANADA)***

Ce dernier utilise un faisceau laser linéaire mais le balayage est réalisé par un dispositif à miroirs multiples tournants (ou à galvanomètre) qui permet à l'objet de rester fixe. La précision obtenue est très grande, mais le volume numérisable est petit (de l'ordre du  $\text{dm}^3$ ).

*AVANTAGE:* Très grande précision de la numérisation;

*INCONVENIENT:* Cout relativement élevé.

Ce système a été utilisé au CANADA pour la numérisation d'objets d'art pour archivage, et remodelage éventuel sur des copies en plastique pour des musées. Le système a en particulier été connecté à une machine de CFAO permettant d'usiner directement des formes dans un matériau plastique spécial à partir de la base de données des objets.

**2-6-5/PAR VUES STEREOSCOPIQUES-TECHNIQUES DE MOIRAGES**

A partir de clichés stéréoscopiques, l'on extrait l'altitude des points selon un maillage contenant l'objet.

Cette méthode comporte des variantes: L'objet est parfois directement éclairé par un système à franges d'interférences, produisant directement un maillage visible à sa surface. Il ne reste plus ensuite qu'à photographier l'ensemble, et utiliser ensuite un système de quadrillage automatique d'objets extrayant l'altitude de chacun des points du maillage. La technique de saisie d'objets par vues stéréoscopiques revient en fait à déterminer des équipotentielles de hauteur sur les objets, et de balayer ensuite les coupes obtenues pour générer un maillage régulier des cotes de l'objet à saisir. Le système du NYIT (New-York Institute of Technology) utilise ce système à maillage direct. Le procédé utilisé consiste à illuminer directement l'objet à représenter à l'aide de franges d'interférences, générant directement le maillage d'équipotentielles à la surface de l'objet. Il suffit ensuite à un programme de calculer sur un maillage plan les différentes coordonnées z de chaque sommet.

Des procédés plus artisanaux ont été utilisés récemment en 87 par MAGNENAT THALMAN à l'Université de Montréal pour modéliser le buste de Marilyn Monroe et de Humphrey Bogard. La méthode utilisée était la suivante:

- Réalisation d'un buste en plâtre des deux acteurs.
- Dessin de facettes à la surface du plâtre.
- Photographie de face et de profil des bustes.
- Traitement informatique pour corrélérer les points entre les deux vues.
- Ensuite, un traitement informatique de déformation d'objets définis par facettes a été appliqué, permettant de décrire les différentes émotions des visages. Ce traitement repose sur des déformations polygonales, obtenues manuellement sur écran (hors de toute modélisation des muscles faciaux) et permettant l'obtention

d'images clefs (expression normale, de peur, de colère, prononciation de phonèmes, etc...)

-Enfin interpolation des positions des facettes pour obtenir une expression contenant un certain pourcentage de ces expressions clefs.

AVANTAGES: Procédé entièrement automatique.

INCONVENIENT: Ne traite pas les concavités cachées au faisceau lumineux. Procédé non interactif.

### **2-6-6/CONCLUSION SUR LES SYSTEMES DE NUMERISATION DIRECTE**

En fait chacun des systèmes décrits dépend du modèle à saisir. La taille, l'opacité, la concavité, la perméabilité magnétique de ce dernier, en particulier, est essentielle pour le choix de la méthode utilisée. En fait, aucun système n'est universel.

Un autre critère est constitué par la nature du matériau:

-Ferro-magnétique ou non (si oui, ceci exclut le recours au procédé magnétique.

-Sensible aux rayons laser : cas du corps humain, de certaines plantes, etc...Ceci limite le recours au système à laser.

-Présentant ou non des concavités : ceci rend difficile l'utilisation des systèmes à balayage, et assez difficile le recours à des systèmes par numérisation ponctuelle, mais ne gêne nullement le système magnétique.

Une dernière variante de ces méthodes consiste à retraiter les nuages de points ainsi obtenus à l'aide d'un système graphique de type CAO. (Cas du CNRC)

### **3/LE RENDU REALISTE DES IMAGES 3D (RENDERING) ETUDE COMPARATIVE DES SOLUTIONS ACTUELLES**

#### **3-1/ LES PRINCIPALES TECHNIQUES ALGORITHMIQUES DE RENDU**

La synthèse d'images est née vers les années 70 environ grâce aux besoins croissants en imagerie pour l'entraînement de pilotes d'avions, d'hélicoptères, ou de cosmonautes. Son but consiste à reproduire généralement sur un écran à balayage de type télévision l'image la plus réaliste possible d'un environnement ou d'une collection d'objets solides.

Dans le cas de la simulation de vol, l'on distingue principalement 2 types de génération:

- La génération du relief d'un terrain, doté de mobiles en mouvement. Dans ce cas, l'observateur est en déplacement dans la base de données qu'il explore au moyen de son système de visualisation 3D.
- La génération de mobiles ou d'objets se déplaçant les uns par rapport aux autres. Dans ce cas, l'observateur est plus généralement fixe par rapport au repère de l'écran.

Depuis ces années, de très nombreuses études algorithmiques théoriques portant sur ces différents domaines sont apparues. Ces algorithmes se décomposent en deux grandes catégories:

#### **3-1-1/LES ALGORITHMES FONDAMENTAUX D'ELIMINATION DES PARTIES CACHEES:**

Ceux-ci sont fondamentaux dans la mesure où ils caractérisent complètement un type de machine avec des fonctionnalités et des performances précises. Le cœur d'un système de synthèse d'images est en effet constitué par cette fonction réalisée de façon naturelle par les lois de l'optique classique (réflexion, réfraction de la lumière sur les obstacles qu'elle rencontre), déterminant pour l'œil ce que l'on convient d'appeler les parties cachées des objets:

- Lignes cachées dans le cas de la représentation de type vectoriel;
- Surfaces cachées dans le cas de la représentation surfacique sur écran à balayage.

Les algorithmes portant sur les représentations de type vecteur ne concernent plus actuellement qu'un secteur de plus en plus limité de la CAO. D'autre part ils constituent un sous-ensemble des algorithmes à balayage de surfaces puisque ces derniers peuvent fort bien représenter des objets en mode filaire: même des industriels de la CAO commencent à utiliser la méthode surfacique du tampon en Z pour résoudre de façon élégante et rapide en temps de calcul les problèmes d'élimination de lignes cachées. L'on peut également remarquer que la plupart des matériels industriels récents utilisent ce procédé qui est actuellement de plus en plus intégré sur composants VLSI à base de silicium.

L'ensemble des algorithmes de surfaces cachées ont été comparés dans les articles de NEWMANN et SPROUL (IEEE 71) (NEW 73).

La technique du tampon en Z, qui sera décrite plus loin, est actuellement la plus largement utilisée sur le plan industriel. Les principales familles d'algorithmes d'élimination des parties cachées des images sont les suivantes:

- Algorithmes de lignes cachées (pour les représentations en mode vecteur)
  - Algorithmes de surfaces cachées par priorités statiques (SHUMACHER)
  - Algorithmes de surfaces cachées par priorités semi-statiques (NEWELL)
  - Algorithmes de surfaces cachées par priorités dynamiques (Z-BUFFER)
- Ces différents algorithmes ont tous vu le jour dans les années 70-74 aux USA.

### **3-1-2/ LES ALGORITHMES D'AMÉLIORATION DE LA QUALITÉ DES IMAGES:**

Sous ce vocable peuvent être regroupés tous les algorithmes dont l'importance est loin d'être secondaire, puisqu'ils concourent à l'amélioration et à l'affinage de la qualité de rendu de l'image en fonction des choix de l'utilisateur.

Les principaux algorithmes de ce type sont les suivants:

- ANTIALIASING ou ANTICRENALAGE
- EFFETS d'OMBRES :  
Lissages de GOURAUD, de PHONG, Modèle de TORRANCE
- TEXTURES
- OMBRES PORTÉES
- FRACTALS pour les structures montagneuses
- EFFETS PARTICULAIERS :eau,feu,...
- GENERATION d'objets particuliers:  
(arbres, nuages, mer, houle, etc...)

Notre propos n'est pas ici de décrire dans le détail les algorithmes permettant de réaliser ces différents effets puisque tous ont fait l'objet de nombreuses communications dans les revues spécialisées. Ils sont tous très connus par la communauté scientifique internationale.

Citons la meilleure d'entre elles: les Proceedings de L'ACM SIGGRAPH et les cours associés qui sont édités chaque année à l'occasion d'un très important congrès annuel, le SIGGRAPH.

Notre propos actuel -qui constitue d'ailleurs l'enjeu principal de ces techniques- consiste plutôt à concevoir les architectures de machines les plus puissantes pour les exécuter.

Cet enjeu est en effet de taille, car les différences de performances obtenues en implantant sur une machine spécialisée divers algorithmes peuvent être considérables.

### **3-2/ LES GRANDS TYPES D'ARCHITECTURES DE MACHINES DE RENDU:**

#### **3-2-1/MACHINES DE TYPE TAMPON EN Z (Z-BUFFER)**

L'origine de l'algorithme remonte aux années 70; A cette époque, il n'était pas possible de l'implémenter pour des raisons techniques: La mémoire en silicium coûtait fort cher, et sa mise en oeuvre comportait des contraintes assez difficiles: Triple alimentation, boîtiers ne contenant que très peu de capacités en nombre de bits ( de l'ordre de 1K). A partir de 1980 cependant, l'abaissement du coût des mémoires rendit attractif son implémentation.

C'est ainsi qu'en 1981, un système appelé CUBI 7 a été conçu et réalisé selon cet algorithme au CCETT. Depuis cette époque et vers les années 85 à 90 d'autres machines utilisant ce même algorithme apparurent sur le marché international.

La technique proprement dite du tampon en Z est la suivante:

La machine de synthèse dispose en plus d'une mémoire d'image classique (par exemple de 1024 par 1024 points codés sur 3 fois 8 bits pour les 3 composantes principales Rouge Verte et Bleue (RVB) ) d'un tampon supplémentaire dit tampon en Z. Ce tampon permet de stocker la cote Z de chaque point de l'image correspondant à la cote de l'objet le plus proche, en ce point, de l'observateur.

En outre le système comprend un comparateur en Z qui effectue pour chaque point d'image nouveau, calculé par exemple par un générateur de polygones, le test de distance en Z de ce point par rapport au point précédemment stocké en mémoire (ou du fond de l'image si aucun objet n'a été jusqu'à présent calculé en ce point).

Appelons:

$X_i, Y_i, Z_i$  les coordonnées d'un point  $M_i$  à afficher aux coordonnées  $X_i, Y_i$  de l'écran.

$X_0, Y_0, Z_0$  les coordonnées du point  $M_0$  stocké en mémoire tampon en  $X_i, Y_i$ .

La couleur de  $M_i$  est codée par exemple sur 3 fois 8 bits  $R_i, V_i, B_i$

La couleur de  $M_0$  est codée par  $R_0, V_0, B_0$ .

Lorsque le générateur de polygone envoie au dispositif du tampon en Z un point  $M_i$ , la cote  $Z_i$  de ce point est comparée à celle du point  $M_0$  déjà stocké (qui peut éventuellement être une valeur du fond si rien n'a été généré à cet endroit).

Si  $Z_i > Z_0$  c'est que le point  $M_i$  est plus éloigné de l'observateur que  $M_0$ : dans ce cas, le système conserve dans sa mémoire d'image les couleurs du point  $M_0$  en  $X_0$ .

Si  $Z_i < Z_0$  le point  $M_i$  est plus proche de l'observateur que  $M_0$ : et le système remplace les couleurs de  $M_0$  en  $X_i, Y_i$  par celles de  $M_i$ .

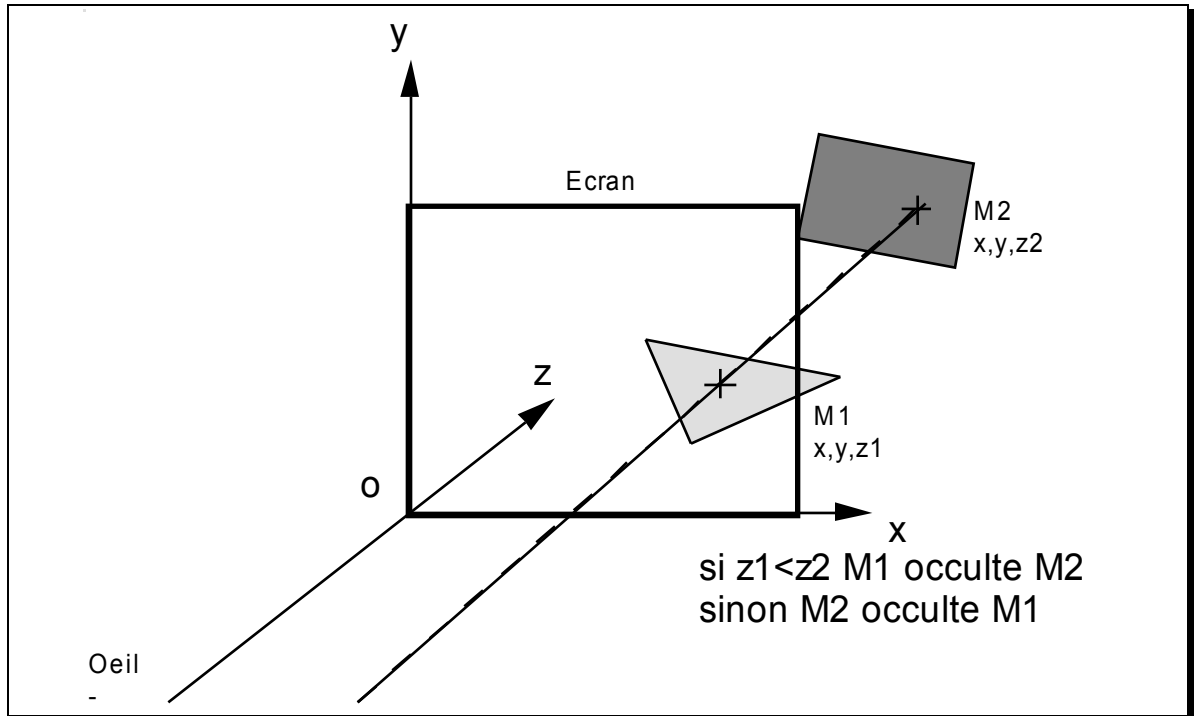


Fig 4: Principe de l'algorithme du tampon en Z

#### ALGORITHME DU TAMPON EN Z (OU Z-BUFFER) AVANTAGES ET INCONVENIENTS DU PROCÉDE:

Les avantages: Contrairement aux méthodes à priorités, la méthode du tampon en Z est entièrement dynamique, c'est à dire qu'elle autorise toute animation des objets ou même des points les uns par rapport aux autres: Elle peut notamment s'appliquer dans le cas extrême des effets particuliers, où chaque particule élémentaire est indépendante des autres et peut se réduire à 1 ou plusieurs points d'écran: rappelons que les modèles particuliers permettent la représentation d'effets spéciaux tels que l'eau, le feu ou les particules en suspension telles que la fumée. (effets qui peuvent être traités de façon systématique par les techniques dites de radiosité).

En outre, cette méthode est très adaptée à une implémentation câblée: C'est pourquoi elle est maintenant très largement utilisée par les matériels industriels et même intégrée en composants VLSI.

Les inconvénients qu'il présente sont les suivants:

-Sur le plan matériel, la mémoire nécessaire est importante. Cet inconvénient est à l'heure actuelle pratiquement négligeable car le prix des mémoires a baissé de façon considérable rendant le surcoût du tampon très faible par rapport au coût du reste de la machine.

En tout état de cause le stockage de la couleur nécessite un tampon mémoire de 3 Mega Octets en  $1024 * 1024$ , exactement la taille requise pour un tampon en Z de 24 bits sur 1 Mega Octet.

Enfin il est possible (mais de moins en moins utilisé sur les systèmes cablés) de réduire le tampon en Z à une seule ligne d'écran où l'on stocke les  $Z_i$ : Il suffit d'ajouter un prétraitement des facettes calculant les valeurs de lignes  $Y_i$  où un polygone nouveau apparaît (en supposant un balayage de haut en bas comme en Télévision). Le tampon porte alors le nom de tampon de profondeur ligne.

-Un autre inconvénient est que l'implantation de techniques d'anti-aliasing (ou anti-crênelage des arêtes) est assez difficile avec le tampon en z, dans la mesure où avec cette technique, l'on ne connaît pas à priori l'ordre de proximité (aussi appelé ordre de priorité) des différentes facettes sortant du générateur de polygones).

Cet inconvénient - à mon sens le plus grave pour cette méthode - est actuellement quasiment résolu par le recours à un calcul dynamique de priorités qui stocke par exemple celle des 4 dernières facettes calculées par le générateur de polygones.

L'on est alors ramené à la méthode classique de pondération par surface utilisée dans les techniques dites à priorités (voir annexe 3).

#### Conclusion:

Cependant les inconvénients énumérés ci-dessus, somme toute assez minimes et dont la plupart sont actuellement résolus, doivent être mis en balance avec la très importante complexité de mise en oeuvre du calcul des priorités des objets.

De plus la limitation à des objets fixes les uns par rapport aux autres est inadmissible en dehors (et même parfois aussi) dans les applications de simulation de vol, ou d'atterrissage.

### **3-2-2/MACHINES DE TYPE LANCER DE RAYONS**

Intrinsèquement, la méthode du tampon en Z comporte un certain nombre de limitations: En particulier, les interactions lumineuses des objets entre eux ne sont solubles qu'au prix d'un certain nombre d'artifices. Par interactions lumineuse entre objets, nous entendons:

- Les ombres portées
- Les réflexions multiples
- La transparence

De plus, les réfractions sont également très difficilement représentables.

La raison de ces limitations est la suivante: La méthode du tampon en Z consiste à balayer l'ensemble de la base de données de la scène à représenter par:

- Les différents objets
- Les facettes de ces objets
- Les lignes de balayage des facettes
- Les points de ces lignes de balayage.

L'on constate donc que lorsqu'un objet est traité et représenté sur l'écran, il n'est plus repris par l'algorithme. Au contraire, l'algorithme de lancer de rayons procède de la manière suivante:

- Balayage par rayons d'abord, puis, pour chaque rayon:
- Balayage par objet.

Il est donc intrinsèquement possible de traiter pour chaque rayon, l'ensemble des interactions de tous les objets affectés par ce rayon et ses rayons éventuellement dérivés.

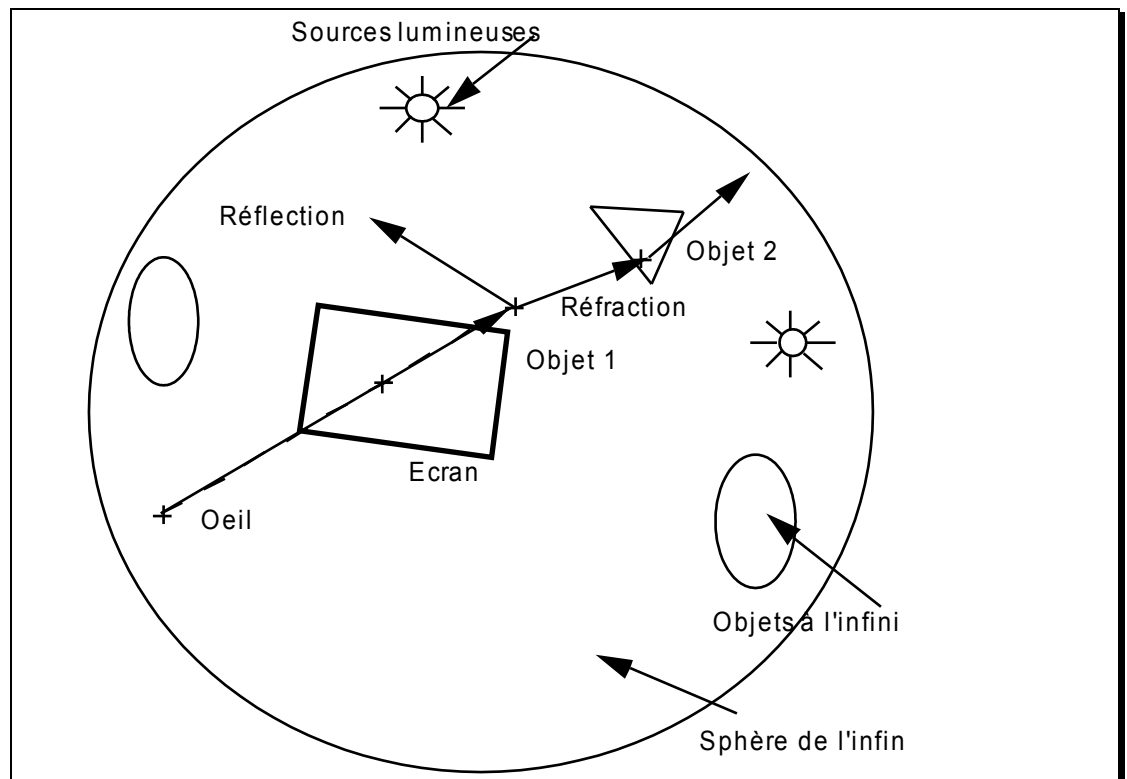
**LANCER DE RAYONS** : Description de la technique algorithmique:

L'algorithme interroge pour chaque rayon joignant l'oeil de l'observateur à un point de l'écran la base de données pour savoir quels objets sont potentiellement intersectés par ce rayon. Il est donc intrinsèquement très aisé de détecter, par le suivi de la trajectoire du rayon (son historique en quelque sorte) si le dernier point visible est ou non dans une zone d'ombre, est ou non dans une zone de reflection par rapport un autre objet.

De plus il est possible de calculer les opérations booléennes éventuellement demandées sur les objets :

Union, intersection, différence, puisque l'ensemble des intersections de chaque rayon avec les primitives, jointes à l'opérateur booléen ci-dessus permet de classer ces points et de choisir le plus proche visible par l'observateur.

Cette opération n'est bien entendu nécessaire que lorsque la base de données est constituée par le modèle CSG.





*Fig 5: Principe de l'algorithme du lancer de rayons*

Les entités intersectées peuvent être soit des polygones, soit des solides géométriques, soit des surfaces mathématiques. Dans ces deux derniers cas, l'algorithme d'intersection conduit à la résolution de polynômes de degré égal à celui de la surface coupée.

#### **AVANTAGES ET INCONVENIENTS DE LA METHODE:**

L'on conçoit bien évidemment la lourdeur d'une telle approche, environ 1000 fois plus longue que la méthode du tampon en Z. Cela tient essentiellement aux raisons suivantes:

-Le calcul de l'intersection rayon-objet peut être très complexe, dans le cas où ce dernier n'est pas un polygone: Dans le cas d'une surface mathématique, le calcul peut comporter la résolution d'équations de degré 18; (splines cube de degré 3) ce qu'il est impossible d'implanter sur du matériel câblé.

Au contraire, un intersecteur de facettes est linéaire, et ne fait intervenir que des additions, des comparaisons, des multiplications et des divisions (qui peuvent être entières) et en nombre très limité et parallélisable. D'ou l'écart de temps de calcul entre les deux méthodes. (Typiquement un verre sur son socle prend sur un VAX en lancer de rayons avec ombre portée, transparence, réfraction et réflexions multiples de l'ordre d'une heure, et quelques secondes avec la méthode du tampon en Z, mais avec transparence seule.

De plus, si l'on considère l'ensemble des objets utilisés en synthèse d'image, il n'est pas systématiquement indispensable de leur adjoindre le calcul des reflets multiples, ni les ombres portées. Les classes d'applications déterminent des critères de qualité de rendu minimaux, qui souvent ne le nécessitent pas.

De plus, l'on s'aperçoit souvent qu'une partie seulement de l'image nécessiterait le recours au lancer de rayons, par exemple pour rendre plus réaliste un reflet métallique, une transparence ou une architecture avec ombre portée.

La solution proposée consiste alors à ne traiter que localement en lancer de rayons les objets ou les parties d'objets qui le nécessitent : métal, verre, etc...

Il faut noter qu'actuellement les recherches en la matière s'orientent vers le rendu d'effets encore plus sophistiqués:

-*Flou de bougé* ou MOTION BLURR.

-*Mode distribué* pour rendre compte de l'effet de granularité des surfaces réelles.

-*Radiosité*, pour la représentation de particules en suspension dans l'espace:

(Fumées, brouillard, feu, eau, etc...)

Pour ces différents effets, des solutions spécifiques sont nécessaires, que nous qualifierons "d'ALGORITHMES d'AMÉLIORATION DE LA QUALITÉ DU RENDU" par opposition à l'algorithme fondamental d'élimination des surfaces cachées.

### **3-2-3/MACHINES MIXTES-SOLUTIONS RECENTES :**

Par rapport à la solution monoalgorithmique, (par exemple: Tampon en Z OU Lancer de rayons, etc...) les travaux de recherche s'orientent à l'heure actuelle vers des systèmes multi-modes, c'est à dire adaptant leur choix algorithmique en fonction des classes d'objets à représenter, ainsi qu'en fonction du temps et du degré de réalisme demandé par l'utilisateur.

### **3-3/ LES PROBLEMES DE LIEN ENTRE MODELISATION ET RENDU**

Nous distinguerons dans la suite de l'exposé 2 types de contraintes de représentation des objets:

- Les contraintes de représentation interne des objets, limitées nous l'avons vu par les tailles mémoire et par les méthodes d'accès.
- Les contraintes techniques de représentation des objets pour le rendu;
- Les contraintes ergonomiques de la représentation des objets. Le temps de calcul qui conditionne le temps de réponse du système en constitue l'une des plus délicates.

### **3-3-1/LES CONTRAINTES DE REPRESENTATION INTERNE DES OBJETS POUR LA MODELISATION**

Dans le domaine professionnel de la CAO, l'affichage de type réaliste de l'image d'un objet sur un écran était souvent davantage un argument de vente spectaculaire plus qu'un outil aidant véritablement l'opérateur dans sa tâche de conception. Cet état de fait est significatif de la période des années 70-85 où le rendu réaliste d'un objet avec élimination des parties cachées, ombrages, couleurs, etc..., c'est à dire ce que recouvre actuellement le vocable de "RENDU" ou de "SYNTHÈSE d'IMAGE" était une tâche coûteuse en temps et en investissement matériel. (voir le préambule).

La méthodologie de conception a donc eu recours pour l'affichage graphique à des solutions économiquement plus acceptables: Le procédé communément utilisé a été la représentation filaire des objets appelée aussi mode vecteurs.

Cependant, la richesse de contenu interne à la machine de la représentation des objets n'a cessé de s'améliorer: Nombre d'informations actuellement présentes dans les grands systèmes de CAO: textures des matériaux, informations de masses, de moments d'inertie, de vecteurs de champ, cinématique, etc... existent en interne dans les mémoires des machines, mais celles-ci sont incapables de les représenter de façon rapide et compréhensible sur l'écran: Prenons un exemple:

La plupart des modeleurs de CAO utilisent des écrans haute résolution (1200 \* 1024 points) mais ne disposant souvent que de 8 bits en sortie: Ceci a pour conséquence de permettre une représentation très correcte en mode vecteurs, mais très pauvre en couleurs: En particulier, il est impossible sur de tels systèmes de représenter des nuanciers de couleurs correspondant à de simples textures, ou à des dégradés de couleurs que l'on rencontre fréquemment dans la nature ou sur des produits manufacturés.

Certains travaillent même par tables symboliques de couleurs, affectant par exemple à une texture d'un type donné un numéro logique, sans aucun support visuel.

Le confort de l'utilisateur (qui doit imaginer les teintes qu'il définit) et donc sa productivité, s'en ressentent d'autant.

Or les contraintes internes de représentation des objets ne sont plus actuellement limitées que par leur taille et par la rapidité des méthodes d'accès: Ce qui signifie concrètement que les logiciels ne sont plus limités par des tailles de mémoire vive, mais plutôt par l'efficacité des méthodes d'accès aux fichiers. Cette évolution est due essentiellement au bas coût des mémoires, qui permettent maintenant à des logiciels de s'implanter dans leur intégralité avec leurs bases de données et sans segmentation en mémoire centrale.

L'on peut cependant noter que les contraintes liées à la modélisation des objets sont particulièrement complexes dans le cas des maillages volumiques où tout solide occupe rapidement des tailles mémoire très importantes. Ce type de représentation fait l'objet de très nombreuses études à l'heure actuelle, débouchant sur une meilleure maîtrise des contraintes précédemment citées.

Les systèmes de modélisation actuels s'appuient de plus en plus sur des représentations par surfaces mathématiques (ou surfaces à pôles) qui autorisent une plus grande souplesse de manipulation grâce à la gestion de ces mêmes pôles.

Tout récemment, l'introduction des quaternions permettra sans doute une souplesse encore plus grande ( Cf: Livre de Mr DE CASTELJAU).

En conclusion, les contraintes liées à la représentation interne des objets est actuellement bien maîtrisée par rapport aux techniques relatives à la représentation visuelle.

### **3-3-2/LES CONTRAINTES TECHNIQUES DE REPRÉSENTATION DES OBJETS POUR LE RENDU:**

Nous venons de voir précédemment qu'en définitive, la base de données des objets à modéliser pouvait contenir en interne pratiquement toutes les informations liées au domaine étudié:

Ces informations sont répertoriées en 3 catégories:

- Les informations de nature géométrique* définissant la forme proprement dite des objets .
- Les informations autres que géométriques* directement affichables sur l'écran: Couleur, texture, etc...
- Les informations non représentables sous forme graphique*: masses , moments d'inertie, contraintes, élasticité du matériau, orientations des vecteurs vitesse dans un champ, un écoulement, métré d'un plan (ex: longueur de tuyauterie dans un bâtiment), etc... Ces dernières doivent cependant également pouvoir être représentées visuellement par autre chose que des données numériques. Ceci est particulièrement vrai dans le cas des champs de vecteurs, puisque les listes de chiffres ont depuis longtemps montré leurs limites dans ce domaine.

Les contraintes techniques de représentation des objets pour le rendu sont liées aux performances des systèmes de synthèse d'image en terme de coût et de temps de calcul, et résumées par la formule citée plus haut:

$$C \cdot T / A = \text{Constante} \quad (\text{ANNEXE 7})$$

Avec :

C: Coût du matériel

T: Temps de calcul

A: qualité Algorithmique

Les contraintes de représentation peuvent être classées par ordre de complexité décroissantes selon le tableau suivant:

<ul style="list-style-type: none"> <li>-<i>Elimination des parties cachées des objets</i></li> <li>-<i>Ombres portées</i></li> <li>-<i>Réflexions multiples</i></li> <li>-<i>Anti-aliasing</i></li> <li>-<i>Textures des objets par fractals</i></li> <li>-<i>Textures des objets par variation des normales à la surface des objets.</i></li> <li>-<i>Textures des objets par motif.</i></li> <li>-<i>Textures aléatoires</i></li> <li>-<i>Colorisation avec lissage (GOURAUD,PHONG)</i></li> <li>-<i>Effet de brume</i></li> <li>-<i>Effets particuliers.</i></li> </ul>
--

*Tableau 1*

Nous citerons à part de cette liste les nouveaux effets investigués de façon très récente:

-Flou de bougé (Motion Blurr)

-Radiosité

qui n'ont pas fait l'objet d'implémentations suffisantes pour être analysés avec suffisamment de recul.

Les contraintes du tableau 1 demeurent à peu près de même complexité lorsque l'on utilise des solutions matérielles ou logicielles. Celles-ci n'ont été véritablement réalisées que pour les premiers éléments du tableau:

-*Elimination des parties cachées*

-*Anti-aliasing*

-*Textures*

Toutes ces contraintes sont à l'heure actuelle liées aux performances des matériels et aux temps de calculs acceptables pour un opérateur.

Les informations de type 1 et 2 (géométrique et non géométrique affichable) sont relativement aisées à implanter.

Les informations de type 3: (informations non directement affichables) commencent à trouver des solutions acceptables, par exemple en représentation moléculaire ou dans le domaine de la visualisation des résultats de calculs: écoulements fluides par exemple, représentés par des ensembles de points ou de flèches avec effets de transparence.

### **3-3-3/LES CONTRAINTES ERGONOMIQUES DE LA REPRESENTATION DES OBJETS**

Les différentes méthodes de modélisation décrites jusqu'ici contraignent l'utilisateur à une ergonomie de type informatique: le clavier jouant un rôle fondamental associé, il est vrai, à des organes d'entrée de type plus 'ergonomique' tels que le crayon lumineux, la tablette graphique ou la souris.

Cependant, aucun des types de dialogue actuellement utilisés ne permettent une interaction directe sur le plan physique entre l'opérateur et l'objet:

En effet, le succès encore remarquable des systèmes bi-dimensionnels de type palette (2D) tient essentiellement au fait que le graphiste opérateur modèle directement son image (en 3D, on ne peut en effet raisonner en terme d'images) grâce à l'action de sa tablette ou de sa souris qu'il assimile peu ou prou à un pinceau; du moins cet organe de dialogue en tient lieu avec les avantages du contact direct entre la main et le résultat tracé sur l'écran.

En 3D, rien de tel: Un sculpteur n'a à l'heure actuelle aucun substitut électronique de ses outils de travail en synthèse d'image: Ce type de modélisation que nous proposons au chapitre 4 peut modifier non seulement l'ergonomie du poste de travail, mais également -et c'est ce point qui nous intéresse- modifier la structure des bases de données utilisées.

En effet, l'information manipulable par un opérateur humain est par essence une information non structurée: ex:

Geste du sculpteur, de la main de l'infographiste, dont l'action doit être accompagnée d'ordres sémantiques, de mieux en mieux structurés actuellement par les systèmes à multi-fenêtrage de type ordinateur personnel. L'information elle-même, donnée par l'opérateur est essentiellement non structurée, et entachée d'erreurs dues au à l'absence de repère fixe utilisable en 3D:

En effet, lorsque l'on rentre un caractère sur un clavier, l'opérateur dispose de repères surfaciques précis (les touches du clavier) qui donnent à la machine une information directement structurée. Ce type d'entrée est transposable en 2D: Certaines approches, en architecture, on consisté à saisir à main levée un plan de maison, l'ordinateur s'efforçant ensuite d'extraire une information graphique sous forme de droites et de rectangles.

Cependant, en 3D la démarche est beaucoup plus complexe, car s'il est relativement aisé de saisir dans l'espace les coordonnées spatiales x, y, z d'un stylet tenu par la main d'un opérateur il est encore très malaisé de définir le type d'action qu'il souhaite accomplir:

En effet l'on s'aperçoit que les informations de type topologique: Modification en fonction de contraintes d'un point, création d'une entité nouvelle sont facilement

compréhensibles par la machine et compatibles avec la base de données des objets. Cependant ce type d'interaction est très difficile à implémenter sur un système pour 2 raisons:

- L'opérateur ne dispose pas de boîte à outils structurée des contraintes à donner.
- L'espace où il évolue ne permet pas de façon aisée (comme c'est le cas en 2D) de fournir des repères dimensionnels ou de désignation par l'intermédiaire de logos, de traits etc...

## **4/ UNE SOLUTION NOUVELLE POUR LA MODELISATION ELECTRONIQUE DIRECTE D'OBJETS 3D.**

### **4- 1/ ANALYSE CRITIQUE DES DIFFERENTES METHODES DE MODELISATION**

Comme il vient d'être souligné au paragraphe précédent, aucune modélisation interactive ne repose à l'heure actuelle sur un concept véritablement 3D:

Seul le processus de modélisation utilisé pour certains objets complexes assure une saisie directe selon ce mode: Mais il n'est pas question, pour l'opérateur à l'heure actuelle de pouvoir modifier dans son espace 3D la forme ainsi rentrée.

Nous nous proposons donc de décrire une méthode originale dite de SCULPTURE ELECTRONIQUE SUR ECRAN, permettant non seulement de saisir, mais aussi de modifier -d'éditer- des objets tri-dimensionnels. Un tel procédé devrait permettre de retrouver le contact direct entre l'opérateur et sa machine que l'on trouve généralement dans le cas des machines 2D où l'action sur l'organe de saisie (par exemple la tablette graphique) se répercute directement et de façon géométriquement similaire au niveau de l'écran.

### **4- 2/ LA SOLUTION PROPOSEE**

Elle consiste essentiellement de disposer d'un outil de saisie tri-dimensionnel associé à une machine de synthèse d'image temps-réel. Cet organe d'entrée peut être soit sonique, soit magnétique, soit à la rigueur mécanique.

Un tel dispositif n'est cependant pas suffisant pour permettre à un opérateur de modeler et d'éditer sa forme: Une information de retour essentielle lui manque: En effet si nous examinons la méthodologie de travail d'un sculpteur, celui-ci dispose non seulement de la vision de l'oeuvre en cours d'ébauche, mais encore, au niveau tactile de sa main, de la rétroaction de force engendrée par la rencontre de son outil avec la forme à créer.

Dans le cas d'une vision sur écran, il est bien entendu impossible d'adjoindre un générateur d'effort dans la main de l'opérateur. Théoriquement plausible, une telle machine poserait sans aucun doute des problèmes de robotique assez pointus.

Il existe cependant à notre avis une solution: Celle-ci consiste à substituer à la réponse tactile du système une réponse de type sonore, le cerveau de l'opérateur pouvant associer, après une période d'apprentissage, le signal perçu à une information spatiale liée à la fréquence émise:

Par exemple, fréquence basse dans le cas où l'organe de saisie (stylet) est à l'extérieur de son objet, et fréquence plus élevée dans le cas où le stylet est à l'intérieur de la forme. De nombreux raffinements ergonomiques peuvent être ajoutés à cette idée: Fréquence intermédiaire à la surface de l'objet, battements, etc...

### **4-2-1/DOMAINES D'UTILISATION**

Couplée à un système de rendu temps réel, un tel système peut s'appliquer à un nombre d'objets important, dans la mesure où l'opérateur peut disposer d'ébauches

volumiques à échelle réduite, dans le cas, par exemple de formes de grandes dimensions (batiments, ouvrages d'art, etc...)

#### 4-2-2/DESCRIPTION THEORIQUE

Le système comprend:

- Une unité de visualisation de formes tri-dimensionnelles fonctionnant en temps-réel, c'est à dire générant chaque image au rythme de 40 mS. Cette image est directement visible par l'opérateur.
- Un organe de saisie 3D, par exemple sonore fournissant à un calculateur hôte une information de position x, y, z de l'extrémité de son stylet qu'un opérateur humain peut saisir à la main. Cet organe est placé, par exemple à la droite de l'opérateur. Le volume explorable par ce dernier est un cube de 1 m de côté environ.

En face de l'opérateur, l'écran du système graphique présente à celui-ci l'image de la forme en cours de modelage. Enfin, un dispositif sonore est relié à l'ordinateur hôte.

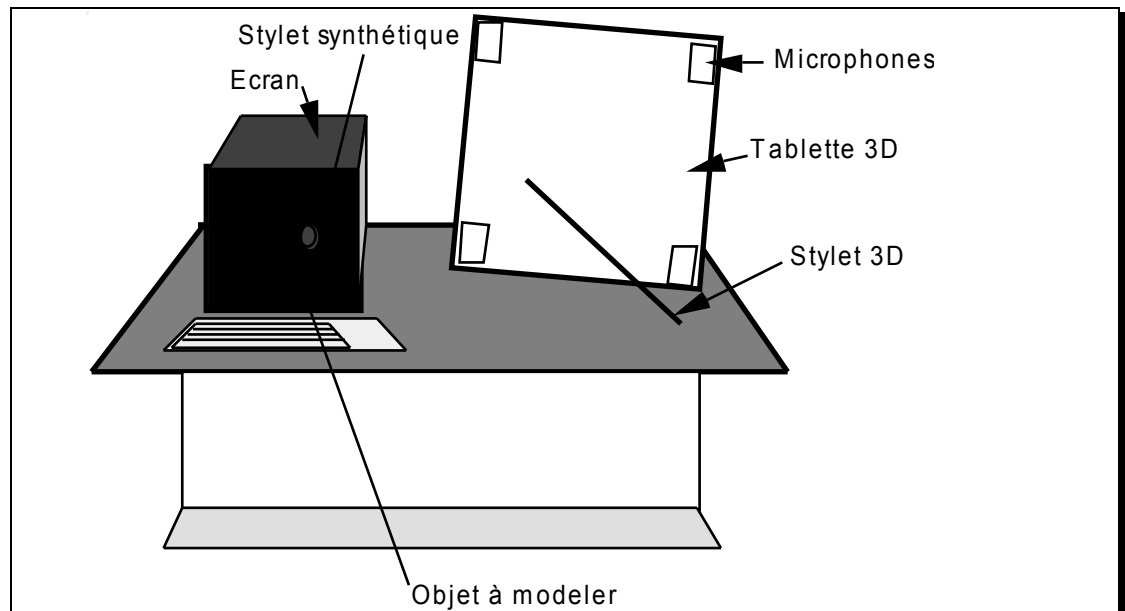
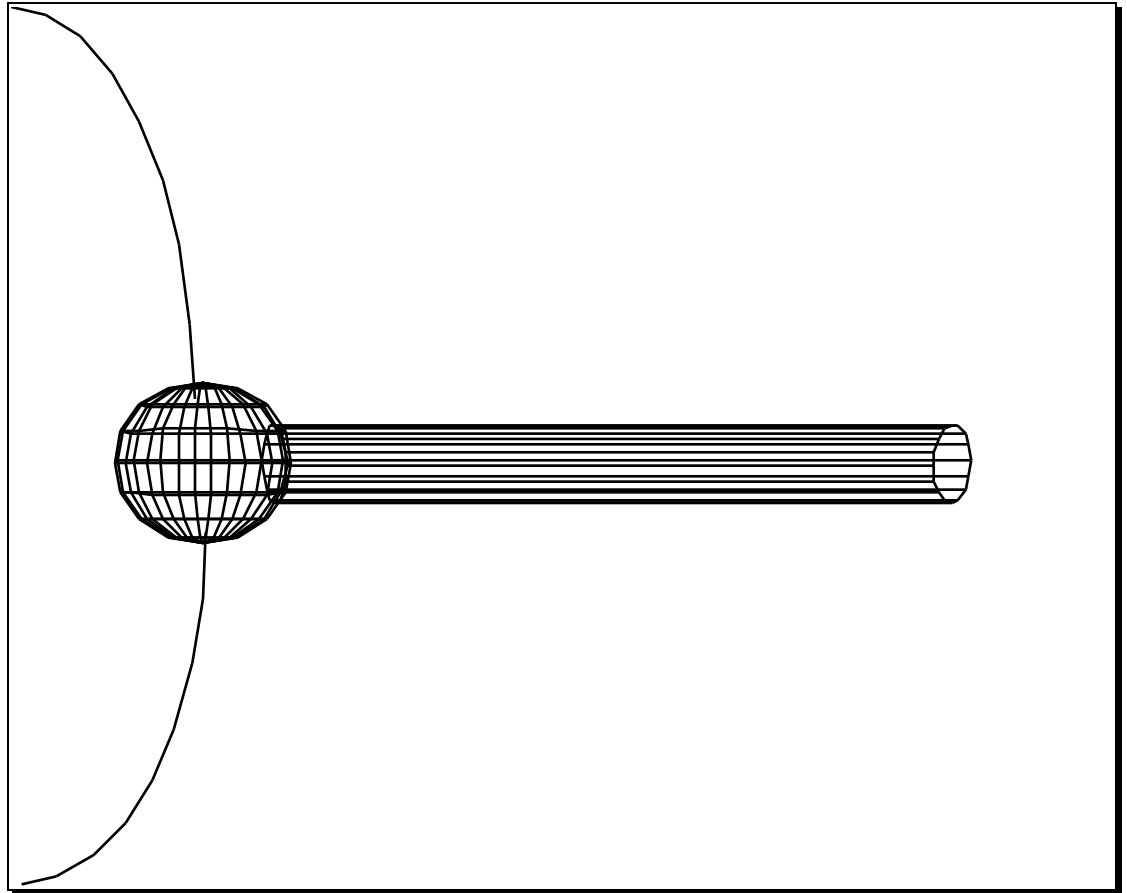


Fig 7 Poste de sculpture 3D sur écran





*Fig 8 Détail de l'écran: outil virtuel de sculpture 3D sur écran*

#### 4-2-3/IMPLANTATION TECHNIQUE

La difficulté principale de ce système réside dans l'écriture du logiciel. Une technique que nous avons testée consiste à définir des "outils virtuels" pour l'opérateur, c'est à dire que l'ordinateur possède en mémoire une forme d'outil (par exemple une sphère de diamètre paramétrable) qu'il fait tourner dans l'espace pour le faire coïncider avec le repère du stylet tenu en main par l'opérateur.

Son travail consiste ensuite:

- A déterminer si le stylet est à l'extérieur, tangent ou intérieur à la surface en cours de modelage. Dans les deux premiers cas, il envoie vers le synthétiseur sonore un ordre de génération de son de fréquence basse, caractéristique de la position du stylet.
- Dans le cas où le stylet pénètre dans l'espace de l'objet à modeler, il calcule l'intersection de la forme de l'outil virtuel (sphère) avec la surface à modeler. Ce calcul s'effectue de la façon suivante:

**POUR** tous les sommets polygonaux de l'outil virtuel  
 { **SI** un Z est > au Z de la surface à modeler  
   **ALORS** calculer la **DIFFÉRENCE** entre

la forme à modeler moins l'outil virtuel

}

Une autre opération peut consister non à enlever de la matière, mais à en ajouter: l'opération effectuée sera une **FUSION**. Une variante à cette méthode pourrait consister à créer un nuage de points correspondant à la position des sommets de l'outil virtuel, et d'afficher ces nouveaux points avec leur Z en lieu et place des Z de la surface à modeler.

Cette méthode a été utilisée en raison des temps de calculs de l'opération différence ou fusion. Cependant, il reste à effectuer au bout d'un certain temps le remaillage du nuage de points ainsi constitué. (Voir annexe INTERSECTION DE VOLUMES et MAILLAGES).

Le principe repose sur un découpage adaptatif d'un triangle en sous-triangles en fonction d'une fonction de coût que l'on cherche à minimiser.

Le terminal de synthèse d'image calcule alors la représentation visuelle de la forme étudiée, et l'affiche sur l'écran de l'opérateur, qui peut à nouveau interagir avec son stylet.

Simultanément, un ordre de signal sonore plus aigu est envoyé au terminal de synthèse sonore. A l'issue d'une série de tests effectués sur ce matériel, nous avons pu déduire les conclusions suivantes: L'opérateur peut après un temps relativement court (quelques heures) apprendre à se positionner dans l'espace, et assujettir sa main à l'objet qu'il désire modeler.

Le problème essentiel est de passer d'un retour d'information tactile sensitif, à un retour de type sonore.

L'expérience a montré que le modelage d'objets était possible dans ces conditions. (voir annexes) Une fois traité, l'objet comporte un nuage de points stocké dans la mémoire d'image du système, contenant les coordonnées x, y, z.

Un algorithme de maillage tel que celui utilisé à l'ENST pour le numériseur LASER (voir bibliographie POTIER-VERCKEN et SCHMITT) peut servir à transformer ce nuage de points en un ensemble de polygones reconnaissables et visualisables par la machine de synthèse d'image. Le nouvel objet peut alors être retraité par la méthode de sculpture électronique, donnant lieu à un nouveau nuage de points, etc...

Divers algorithmes peuvent être choisis:

- Exploration de la surface idéale et minimisation des distances des points par rapport à cette surface
- Surfaces mathématiques de type Coons, approximant au mieux le nuage de point.
- Minimisation des rayons de courbures locaux des courbes u v sur la surface.

Une seconde difficulté réside dans la relative lenteur des temps de réponse du système: En effet le système de connection du calculateur hôte au système graphique est très lent: chaque échange de données doit être précédé d'un protocole de dialogue dans le processeur hôte qui demande de l'ordre de 10 mS. Ce dialogue se répète à chaque envoi de mot vers la machine de synthèse d'image. L'on conçoit aisément que

le temps de réponse est de l'ordre de 1s pour l'envoi de plusieurs centaines de mots correspondant aux coordonnées des points modifiés. De plus le temps de transmission vers le port série contrôlant le périphérique de synthèse sonore est également lent. Ces contraintes techniques induisent des difficultés dans la mise en oeuvre opérationnelle de ce procédé qui seront détaillées plus loin.<sup>(3)</sup>

#### **4-2-4/ETUDE DES PERFORMANCES**

Une telle méthode modifie les concepts traditionnels de modélisation. En effet l'opérateur a dans ce cas physiquement contact avec la pièce à modéliser (du moins, en a-t-il la sensation, ce qui est essentiel).

L'expérience tentée sur cette méthode, malgré ses limites dues à la relative lenteur du matériel utilisé, a permis de modéliser des objets de formes diverses à partir de volumes de départ simples. ( voir photos en annexe 2 du rapport). Cette technique donne accès à une modélisation très fine, dans l'espace, comparable à ce que connaissent les infographistes dans le plan.

#### **4-2-5/LIMITES ET AMÉLIORATIONS POSSIBLES**

L'expérimentation qui a été menée comportait les limitations suivantes:

- Lenteur du matériel de synthèse temps réel utilisé (bien que très performant dans l'état actuel du marché)
- Fragilité et sensibilité aux parasites de la table à numériser en 3D.

Les améliorations possibles d'un tel système sont les suivantes:

- Amélioration du temps de réponse du terminal de calcul d'image: Celui-ci doit être au moins capable de générer en temps réel des images de l'ordre de 2 à 3000 facettes.
- Couplage du système sonore au terminal de visualisation, afin de minimiser le temps de réponse sonore du système, lorsque l'opérateur change la position de son stylet.

### **4-3/ CONCLUSION, LIMITES ET AVANTAGES DE LA SOLUTION**

#### **PROPOSEE**

Ce système nouveau de modélisation ne pourra véritablement s'intégrer dans un processus de conception CAO que lorsqu'un effort d'industrialisation important aura été fait pour résoudre les problèmes cités plus haut.

Sous cette condition il pourrait apporter un changement important sur les méthodes de modélisation actuellement appliquées en CAO électronique.

---

<sup>(3)</sup> Cette étude a été réalisée en 87 sur un ordinateur de type VAX de DIGITAL et une machine de synthèse d'image CUBI 7. Les systèmes actuels possèdent des performances très supérieures (en ce qui concerne les micro-ordinateurs, en particulier pour les liaisons séries rapides temps réel, qui atteignent 300 kBauds/s); De plus les machines de synthèse actuelles atteignent comme CUBI 7 de l'ordre de 1500 facettes par seconde, alors que 100 à 200 facettes étaient affichables en temps réel à cette époque. En conséquence, l'intérêt de ce procédé serait actuellement encore plus significatif.

#### **4-4/ UNE PROPOSITION NOUVELLE DE SYSTEME DE MODELISATION NEURO-MIMETIQUE:**

La proposition précédente s'appuie sur une conception classique de matériel de synthèse, encore qu'à l'heure actuelle non encore réalisée. Ce système nécessite en effet une puissance d'affichage temps réel, afin que toute intervention du stylet induise immédiatement une modification sur l'écran. La machine CUBI 7 depuis 87, et actuellement d'autres machines de synthèse d'image répondent actuellement à cette condition, puisqu'il est actuellement possible d'afficher avec du matériel commercial de l'ordre de 500 à 1000 facettes de 2 à 300 points avec un ombrage lisse en temps réel.

Cependant, la méthodologie utilisée peut dans certains cas paraître longue lorsque l'on souhaite obtenir rapidement la numérisation d'un objet complexe: La situation est à peu près identique aux palettes graphiques d'ancienne génération non dotées de scanners: Le graphiste doit alors toujours partir de zéro pour réaliser un dessin, de même qu'un sculpteur part d'une pierre, ou d'un morceau de bois brut.

Il est donc logique d'imaginer une sorte de scanner 3D permettant la numérisation directe d'objets 3D pour lequel le système ci-dessus proposé permettrait la retouche fine. Le système serait similaire dans son principe aux scanners à balayage laser. La différence résidant dans le fait qu'au lieu d'une saisie sans intelligence, le système comporterait un système neuro-mimétique préalablement entraîné à la reconnaissance de formes et de mouvements 3D: Chaque objet nouveau présenté devant lui enrichissant sa base de formes élémentaires. (De manière similaire à un être humain qui en fait "reconnait" un objet par auto-association avec une base de donnée comportementale stockée dans les coefficients synaptiques de ses neurones.

L'approche nouvelle que nous proposons repose sur l'association avec apprentissage des 2 techniques suivantes:

- La technique du "SHAPE FROM SHADING" ou extraction de la forme à partir de l'ombrage.
- Les systèmes neuronaux ou Neuro-Mimétiques (voir annexe 9);

#### **PRINCIPE DE LA METHODE:**

Les techniques de "SHAPE FROM SHADING" actuellement explorées en recherche avancée dans le monde butent sur la formulation mathématique complète qui permet d'obtenir la forme d'un objet à partir de son ombrage, éventuellement donné par un couple de vues stéréoscopiques et par son mouvement.

En effet, les hypothèses simplificatrices nécessaires à l'obtention d'un ensemble d'algorithmes formels mathématiquement exprimables sont irréalistes dans le monde environnant; A l'opposé, la formulation complète est impossible à exprimer mathématiquement.

Afin d'appuyer sur une réalisation concrète cette proposition, nous allons décrire maintenant un système d'analyse neuro-mimétique d'objets 2D réalisé sur un micro-ordinateur: (LERAY 90)

Le réseau utilisé comporte 2 couches:

- *Une couche de neurones d'entrée* (NE) qui reçoit les valeurs de pixels de l'objet 2D à reconnaître. Nous avons pris un ensemble de  $16 \times 8$  pixels dans le programme que nous avons implémenté, ce qui correspond donc à 128 neurones d'entrée. (Ce nombre pourrait bien entendu être modifié pour l'analyse d'objets de plus grande complexité.)

- *Une couche de neurones de sortie* comportant autant de neurones que de prototypes (ou classes) d'objets différents à numériser. Nous appellerons ce nombre NS.

Le type d'apprentissage que nous avons réalisé est un apprentissage différentiel direct.

Chaque coefficient synaptique est stocké dans une matrice de taille  $NE \times NS$ . Il représente l'efficacité  $w_{ij}$  du lien qui lie un neurone d'entrée  $i$  à un neurone de sortie  $j$ . L'opération effectuée durant la phase d'apprentissage consiste à stocker les valeurs de pixels reçues dans les coefficients synaptiques.

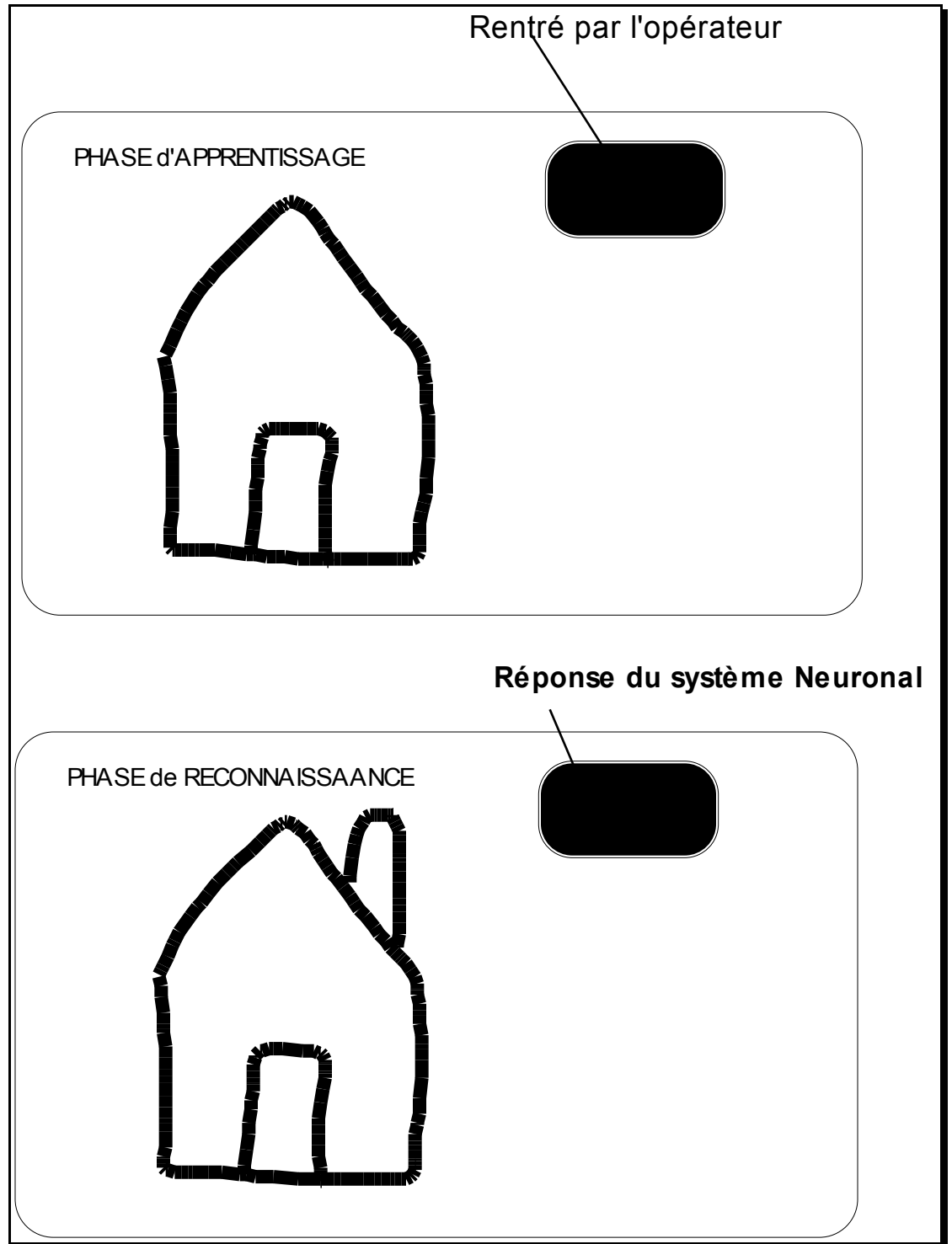
Durant la phase de reconnaissance l'on procède à l'opération suivante: Soit un vecteur de composantes  $e_i$  présenté sur la couche d'entrée des neurones:

Pour tout neurone de sortie:

```
{
    Pour tout neurone d'entrée
    Calcul de l'énergie  $(e_i - w_{ij})^2$ 
}
```

**Calcul de l'énergie minimale:** le neurone de sortie gagnant est donc le plus proche vecteur correspondant à l'entrée présentée.

*UN EXEMPLE DE SAISIE D'OBJET 2D*



Dans l'exemple ci-dessus, il est possible d'associer en plus du vecteur "maison" un vecteur représentant la géométrie de l'objet "maison" fourni durant la phase d'apprentissage.

Il est envisageable d'étendre ce procédé à des ensembles complexes d'objets, dont les éléments auront préalablement fait l'objet d'une phase d'apprentissage.

L'article de J. HERAULT (HER 90) décrit en annexe 9 fait état d'une solution de ce type pour l'analyse neuronale d'objets 3D.

**ANALYSE CRITIQUE DE LA SOLUTION PROPOSEE:**

La solution proposée constitue une préfiguration très partielle des futurs systèmes de numérisation (analyse), de modélisation et de synthèse des images. De nombreux problèmes demeurent liés à cette approche: constitution d'une bibliothèque d'objets 3D invariants en rotation-translation pour la reconnaissance, compréhension de la profondeur, et du mouvement. De récents livres sur les techniques du SHAPE from SHADING ou extraction de la forme à partir de l'ombrage laissent présager de remarquables développements dans cette voie, malheureusement jusqu'à présent non encore couplés au calcul neuronal, à mon sens seul capable de prendre en charge une approximation sous-optimale des algorithmes mathématiques d'extraction de formes 3D à partir de l'ombrage, ainsi qu'une réponse comportementale pouvant intégrer l'ensemble des formes discernables par le cerveau humain.

*La synthèse d'image proprement dite pouvant à terme devenir la mise en correspondance dans la phase d'apprentissage du réseau neuronal entre une scène réelle et ce que l'on souhaite voir dessiné par la machine (cette image étant précisément réalisée par technique de synthèse classique).*

## **5/UNE PROPOSITION d'ARCHITECTURE ORIGINALE DE MACHINE DE RENDU**

### **5- 1/ ETUDE CRITIQUE DES SOLUTIONS ACTUELLES**

Les solutions d'architecture actuellement disponibles ou en cours de développement (excepté PIXAR et AT&T) sont des solutions de type station de travail: ex: SILICON GRAPHICS, HEWLETT-PACKARD, etc...

Elles répondaient et répondent encore à un besoin précis, celui du poste de travail de saisie et de modélisation des objets. Or le coût des stations individuelles tend actuellement sauf dans des domaines très particuliers (militaire, spatial, avionique, etc...) à s'aligner sur les coûts des matériels de type PC. Comme les fonctions de rendu évolué nécessitent des complexités de machines très supérieures, résolues actuellement par l'utilisation de super-calculateurs, l'on constate qu'il est illusoire de doter des stations de travail professionnelles de telles possibilités.

**NOTA:** Sur les stations de travail de la classe IRIS de SILICON GRAPHICS, la fonction de rendu n'utilise les fonctions câblées de la machine, (c'est à dire: le tampon en Z, et les processeurs VLSI) que pour le rendu grossier des objets, pendant la phase de modélisation. Le rendu de qualité est généré sur de gros calculateurs rapides non dotés de fonction VLSI graphique particulière.

Il est logique en fait que les stations de travail se limitent au rôle de saisie et de visualisation élémentaire des objets 3D.

La fonction de rendu, ressource chère par excellence, doit être en conséquence partagée entre plusieurs utilisateurs -locaux ou distants- sur des architectures de machines spécialisées (et non des super-calculateurs dont le coût est très élevé pour une puissance spécifique faible).

### **5- 2/ PROPOSITION D'ARCHITECTURE DE RENDU: CUBICAST**

#### **5-2-1/DOMAINES D'UTILISATION**

L'architecture proposée vise essentiellement à apporter une solution au rendu d'image de haute qualité, avec des temps de réponse très courts. Il est clair qu'une telle architecture ne saurait être rapidement transposée à des stations de travail individuelles. Son domaine premier d'utilisation est le serveur d'images connecté en réseau local ou public à des stations de type PC.

#### **5-2-2/DESCRIPTION THEORIQUE**

Nous avons montré plus haut qu'entre les stations de travail dotées de fonctions VLSI et les super-calculateurs utilisés pour le rendu, d'intéressantes fonctionnalités pouvaient être réunies dans des systèmes comportant:

- Des processeurs VLSI
- Des réseaux de processeurs parallèles configurables dynamiquement.



L'expérience montre, en effet, que ces 2 concepts matériels sont complémentaires:

***Les processeurs VLSI apportent une très importante amélioration des temps de calcul grâce:***

-*Au parallélisme possible de fonctions élémentaires implantables soit sur silicium soit dans plusieurs boîtiers discrets: Opérateurs arithmétiques, comparateurs, etc...*

-*A la puissance de calcul intrinsèque de ces derniers : par rapport à un calculateur général, chaque fonction implantée directement sur silicium permet un gain de temps de l'ordre de 10: ex: addition entière sur un super-mini classique 500 à 1000 nS; sur un additionneur en silicium: 50 à 100 nS.*

En contre-partie, leurs fonctions et leur possibilités d'amélioration sont beaucoup plus rigides que celles des processeurs parallèles.

En fait, l'on s'aperçoit de façon très qualitative que leur performance est grosso-modo inversement proportionnelle à leur souplesse. Ce point est très difficile à quantifier précisément (qu'est-ce que la souplesse? ) et ne peut se juger que dans le contexte de l'évolution globale de ces techniques: Seuls les systèmes à la fois les plus flexibles et les plus performants se développeront.

-***Les architectures de processeurs parallèles*** apportent au contraire une moins grande amélioration en performance intrinsèque (non négligeable tout de même, puisqu'un TRANSPUTER, par exemple effectue 10 millions d'instructions par seconde (10 MIPS), alors qu'un mini classique n'en fait que 2 à 3.

Par contre leur souplesse de programmation est totale, puisque l'on peut les reprogrammer entièrement. (cas du TRANSPUTER, par exemple). La faculté de parallélisation est corrélativement moins grande que pour la logique câblée du fait que pour fonctionner, chaque processeur élémentaire a besoin d'un environnement de boîtiers extérieurs: Mémoires, Unités de contrôles, interfaces avec l'extérieur, etc...

### 5-2-3/IMPLANTATION TECHNIQUE

***Remarque préliminaire:*** Les calculs de temps de traitement des algorithmes qui seront développés par la suite s'appuient uniquement sur des structures de machines travaillant à partir de données préalablement stockées en mémoire vive: Sont en particulier **exclus de ces calculs** les phases de chargement de données et de programmes à partir des **disques informatiques**, l'ensemble des traitements de synthèse d'image se faisant à partir de la mémoire vive associée soit avec un microprocesseur (option **P** ci-dessous) en fonctionnement classique, soit avec un microprocesseur doté de micro-programmes graphiques spécifiques (option **MP** ci-dessous), soit enfin avec un ensemble microprocesseur doté d'une logique câblée (option **C** ci-dessous).

Les temps d'accès des périphériques de type disque sont en effet plus lents d'un facteur de 100 000 environ par rapport à une mémoire vive. Il est donc totalement exclu d'utiliser ces matériels dans les phases de calcul d'image de synthèse, sauf bien

*entendu pour les phases de chargement ou de stockage des données et des programmes.*

Les remarques précédentes tendent à envisager une solution mixte à base de logique câblée (en VLSI) d'une part et microprogrammée d'autre part comportant une architecture multi-processeurs.

Le point essentiel consiste à trouver un équilibre entre les algorithmes implantés sur l'un ou l'autre type de support matériel. A ces 2 types, il convient d'ajouter une solution intermédiaire consistant à micro-programmer et en fonction de chaque algorithme, d'étudier le gain en performance obtenu, ainsi que le surcoût engendré par les différentes solutions. Dans le diagramme suivant :

- P:** signifie que l'algorithme est programmé
- MP:** signifie que l'algorithme est micro-programmé
- C:** signifie que l'algorithme est câblé.

ALGORITHME:	P	MP	C	
ELIMINATION des				
PARTIES Cachées:	1	5	900	gain en temps
	1.	1.1	2	Surcoût
ALIASING	1	5	1000	gain en temps
	1.	1.1	2	Surcoût
TEXTURE	1	5	2000	gain en temps
	1.	1.1	2	Surcoût

Ces chiffres ont été mesurés à partir des gains de temps obtenus sur des machines commerciales (VAX 750, CUBI 7), et calculées dans le cas où l'implémentation n'a pas été encore faite, avec les surcoûts corrélatifs lorsque l'option câblée est implantée. L'on sait, par expérience (voir gammes de prix des constructeurs de simulateurs) que le seul ajout d'une option de texture sur une machine temps réel pour la simulation double quasiment son volume, et en conséquence son prix.

Ces chiffres sont donnés en tenant compte également du fait que le coût de l'électronique est proportionnel à son volume. Cette règle est vérifiée par tous les équipements électroniques mondiaux.

Les conditions sont toutes autres lorsque l'option câblée est réalisée non plus de façon classique à base de cartes et de composants du marché, mais à partir de boîtiers VLSI définis de façon spécifique. L'investissement au niveau de la conception du composant est alors très importante (en moyenne 5 à 10 MF pour 100000 transistors). Cependant l'économie est reportée sur la fabrication proprement dite du produit final, qui peut voir ses cartes remplacées par un ou deux composants spécifiques. Leur prix peut alors chuter dans un rapport très important, par exemple de 1 à 10 tout en conservant les mêmes performances.

Ex: les cartes additionnelles sur IBM PC avec Z-Buffer (de type MATROX ou ATT) qui sont actuellement commercialisées aux alentours de 30 KF remplacent des

ensembles à plus de 300 KF à la pointe de l'électronique il y a 2 ans (type Silicon Graphics).

L'un des concepts principaux sur lequel repose le système proposé est la notion de multimode, c'est à dire que le fonctionnement interne de la machine peut se faire selon des modes algorithmiques différents:

Compte-tenu des études décrites au chapitre 3, deux algorithmes fondamentaux de rendu permettent de couvrir la quasi-totalité des demandes en synthèse d'image: Ce sont:

***-Les algorithmes de tampon de profondeur (ou Z-Buffer)***

pour les images temps-réel ou de qualité dite interactive (dont le temps de réponse est de l'ordre de grandeur de la seconde, et compatible avec un dialogue opérateur-machine). Les algorithmes dits d'amélioration de la qualité des images (ou d'effets spéciaux) compatibles avec le tampon en Z sont les suivants:

- Antialiasing
- Textures surfacique par motifs
- Textures par variation de normales
- Texture volumique
- Transparence
- Sources d'éclairage multiple
- Réflexion spéculaire
- Ombre portée rapide par double tampon en Z.

***-Les algorithmes de lancer de rayons:***

Pour le rendu de haute qualité en temps différé. Les algorithmes d'effets spéciaux compatibles sont les suivants:

- Tous effets de lumière disponibles par la méthode du tampon en Z plus ceux issus d'interactions entre objets:
- Réflexions multiples
- Réfraction

A ces effets peuvent être ajoutés ceux liés à la méthode du mode distribué, c'est à dire mettant en jeu non plus un rayon par direction donnée, mais les réponses à un échantillonnage de rayons.

Les effets réalisables par cette méthode sont les suivants:

- Flou de bougé (ou Motion Blurr)
- Texture et anti-aliasing fin
- Profondeur de champ

Les seuls effets non directement implémentables sont ceux nécessitant le recours à la méthode de la radiosité pour notamment la prise en compte du bilan énergétique d'une scène.

Cependant, la prise en compte de l'affaiblissement de la lumière sur son trajet dans différents milieux peut être implémenté. Ceci peut être d'une grande utilité pour les simulations d'éclairage en mode visible ou infra-rouge.

La solution proposée doit donc permettre de fonctionner dans les deux modes, et de mixer les images obtenues dans un tampon en Z disposant de bits supplémentaires (de l'ordre de 8) pour le fondu anti-aliassé des différentes parties d'images (technique du A-Buffer).

La machine comportera un ensemble de processeurs parallèles de type transputer (voir annexe 1). Sur les feuilles du graphe constitué par les transputers pourront être greffés de façon modulaire les circuits spécialisés du tampon en Z:

- Circuit de prétraitement effectuant les opérations de rotation , clipping, calcul des normales, transformation perspective
- Circuit de génération de facettes ou de remplissage
- Circuit de gestion de la mémoire et du tampon en Z.

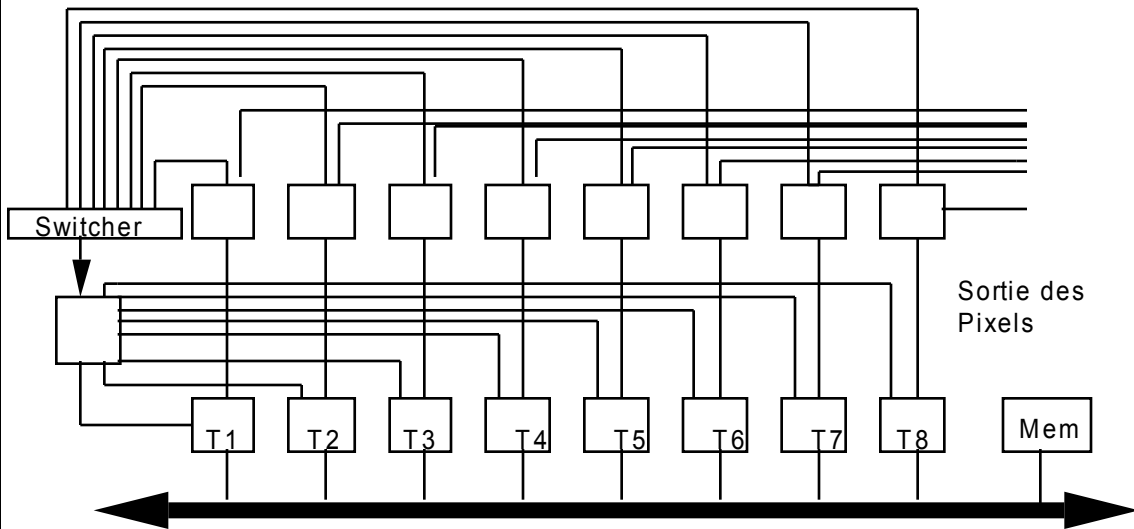
En fonction des algorithmes retenus, la machine pourra utiliser ou non les modules VLSI cités ci-dessus en fonction des parties de la base de données à traiter. Ce type d'architecture permettra de pallier de façon efficace aux insuffisances actuelles des calculateurs généraux dotés uniquement d'unités centrales parallèles coûteuses et surtout non équipées de processeurs spécialisés adaptés au rendu graphique.

Une machine de type Convex ou Alliant offre une puissance de l'ordre de 20 à 30 Mflops ce qui est très insuffisant en rapport aux puissances nécessaires pour la synthèse d'image (Une image demande de l'ordre de 1000 Millions d'opérations).

La machine proposée, dotée de processeurs graphiques sur chaque unité terminale de ses unités centrales, pourrait avoir un gain en puissance identique à celui constaté entre un calculateur mono-processeur et la logique câblée ( de l'ordre de 1000: voir calculs pages suivantes). Son schéma est donné ci-dessous.

- Fonction synthèse rapide
- Fonction synthèse haut de gamme éventuellement multimode
- Fonction calcul parallèle (Transputers)

**FONCTION SYNTHÈSE HAUT DE GAMME**



**FONCTION SYNTHÈSE RAPIDE ET SYNTHÈSE HAUT DE GAMME**

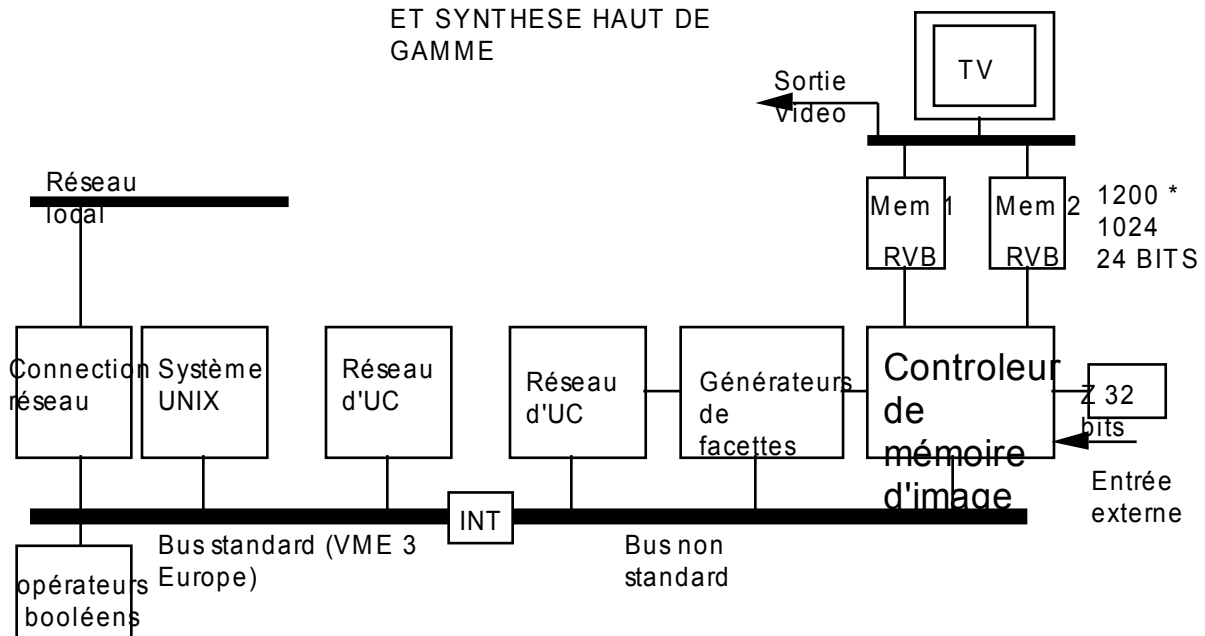


Fig 9: Schéma synoptique de Cubicast

### 5-2-4/ ETUDE DU FONCTIONNEMENT ET CARACTERISTIQUES ORIGINALES DE CETTE ARCHITECTURE

L'étude du fonctionnement de cette architecture de machine doit être réalisée en faisant la remarque suivante: Le bus représenté en trait noir gras de la figure 8 est un bus monodirectionnel, c'est à dire que les informations qu'il véhicule vont toujours dans le sens BUS vers TRANSPUTERS, ce qui élimine tout conflit de données, et supprime du même coup les problèmes d'arbitrage de bus, source des lenteurs habituelles des bus classiques.

Cette solution est ici proposée en raison de la particularité de l'algorithme de synthèse:

Dans le cas du tampon en Z, comme dans celui du lancer de rayons, le bus, par l'intermédiaire de sa mémoire de contrôle et de son transputer de commande SOUMET en fait à l'ensemble des transputers qui lui sont connectés soit:

- Une facette à remplir dans le cas du tampon en Z.
- Un rayon à traiter dans le cas du lancer de rayons .

Dans ce cas, chaque transputer dispose dans sa mémoire locale d'une partie de la base de données des objets à représenter. Cette base de données peut être très vaste: de 100000 à 1000000 d'équivalents-polygones. Il convient donc de PARTAGER la tâche consistant à tester, pour chaque rayon, si un objet connu du transputer T l'intersecte.

L'intérêt de l'architecture proposée repose sur le fait que cette tâche peut être réalisée en parallèle, et que l'ajout de 1 ou de N transputers supplémentaires n'obère en rien les performances de l'ensemble.

Ainsi, en résumé, si  $dt_0$  représente le temps élémentaire de test d'une intersection rayon-objet et si  $N$  est le nombre de transputers T, la performance globale du système est, en temps, de :

$$dt_0/N$$

A partir de cette formule, et compte-tenu du coût d'un ensemble élémentaire constitué de:

-1 Méga-octet de RAM

-1 Transputer

L'on s'aperçoit qu'au prix d'une machine actuelle de 20 MFLOPS (de type CONVEX par exemple) l'on obtient:

3000 MFLOPS pour le système proposé.

Le ratio de puissance obtenue est de 150 fois, à prix égal.

Ces évaluations de performances ne tiennent compte que de la puissance estimée par processeur en MFLOPS, indépendamment de la structure fine de l'algorithme retenu. Elles montrent cependant un ordre de grandeur intéressant concernant les puissances obtenues à prix égal. Le gain final reposant, rappelons-le sur le bus mono-

directionnel adapté aux algorithmes de rendu, et permettant une mise en parallèle sans dégradation de performances, grâce à sa mono-directionnalité.

Il nous reste maintenant à aborder l'analyse fine de l'algorithmie, afin d'évaluer de façon précise les temps de calculs.

**5-2-5/ETUDE DES PERFORMANCES**

Pour évaluer le gain en performance du type d'architecture proposé, il nous faut examiner 2 points qui conditionnent la puissance d'une machine de synthèse d'image:

*-La capacité de traitement du pré-processeur:* qui effectue les rotations, clipping et transformation perspective. Sa puissance est mesurable en nombre de polygones traités par seconde.

Cette puissance est indépendante de la surface des polygones, étant donné qu'ils effectuent des opérations arithmétiques sur les sommets de ceux-ci. Afin de prendre une référence constante, nous choisirons de ne traiter que des triangles. La capacité globale de cet étage est donc directement liée au nombre de polygones.

*-La capacité de traitement du générateur de polygones:*

Celle-ci est directement proportionnelle à la surface de ces derniers et au temps de calcul d'un point d'écran. Si l'on considère un générateur de facettes pouvant calculer 1 point en 50nS (ceci représente les performances actuellement atteintes par les meilleurs circuits actuels) d'une part, et si l'on évalue d'autre part les opérations de calcul d'un point courant d'image:

$$\begin{aligned}
 x &= x + dx \\
 R_x &= R_x + dR_x \\
 V_x &= V_x + dV_x \\
 B_x &= B_x + dB_x
 \end{aligned}$$

TEST :  $x = x_{\text{droit}}$  ?

Ceci représente en prenant pour référence un Transputer seul (en précision entière: virgule fixe) (voir Annexe 1):

4 additions	à	100 nS	400nS
1 test	à	100nS	100nS
4 Chargements LOADS	à	100 NS	400 nS
4 Stockages STORE	à	100nS	400 nS
<b>TOTAL:</b>			<b>1300 nS</b>

Gain de temps :  $1300/50 = 26$

Si nous prenons pour référence un calculateur tel que le VAX ou le micro-VAX (en précision entière) le même calcul donne:

ADD: 4\*1000  
 TEST: 1\*1000  
 LOAD: 4\*500  
 STORE: 4\*500

Total: 9000 nS

GAIN en temps:  $9000/50 = 180$

**REMARQUE:** Ceci suppose que les opérations élémentaires sur le point image n'ont jamais à attendre les résultats des calculs des étages supérieurs (pré-traitement de l'image, clipping, etc...) ce qui est actuellement réalisé sur les meilleures machines câblées pour la simulation actuellement, à l'aide de composants à haute intégration VLSI.

#### **CAS DU LANCER DE RAYONS:**

La tâche fondamentale est celle du test des boîtes englobantes des objets de la scène avec le rayon à analyser:

Les premières techniques utilisées ont consisté à tester systématiquement TOUTES les boîtes englobantes de TOUS les objets de la scène: L'on balayait donc l'ensemble:

PAR RAYON puis :  
 PAR BOITE  
 PAR OBJET

De nouvelles techniques récemment apparues ont consisté à subdiviser l'espace en sous-espaces à l'intérieur desquels le suivi de chaque rayon peut se faire de manière plus optimisée: En effet, dès que l'intersection d'un rayon primaire avec une boîte est connue, et si de plus ce rayon intersecte une primitive à l'intérieur de cette boîte, alors les rayons réfléchis et réfractés issus de ce rayon sont beaucoup plus rapidement calculables, dans la mesure où la probabilité est forte pour qu'ils affectent des boîtes et des objets voisins.

De plus, au franchissement d'une boîte par un rayon (sans même qu'il y ait intersection) il est relativement aisé de déduire quel sous-espace il va parcourir.

L'opération fondamentale caractéristique du temps global de calcul sera donc l'intersection des boîtes avec les rayons.

En effet, les autres opérations telles que le calcul fin d'intersection et l'ombrage peuvent être pipe-linés <sup>(4)</sup> avec les tests d'intersection rayons-boîtes qui sont naturellement plus nombreux. Seule l'expérience peut donner sur des scènes concrètes des chiffres précis, étant donné que cet algorithme est essentiellement adaptatif, en

---

<sup>(4)</sup> L'opération de pipe-linage d'un algorithme consiste à le découper en un ensemble d'opérations élémentaires effectuées sériellement. La durée d'obtention d'un résultat étant égale au temps élémentaire de chaque opération. L'exemple le plus significatif consiste à mettre N personnes pour faire une chaîne de transport de paquet. En bout de chaîne, chaque paquet arrive à chaque temps d'opération élémentaire.



fonction de la scène à calculer, mais également en fonction du découpage en boîtes qui a préalablement été fait lors de la constitution de la base de données.

Afin de conférer le maximum de puissance à cet ensemble, la phase de calcul d'intersection boîte-rayons doit se superposer en temps aux tâches d'intersection fine et d'ombrage. Si tel n'était pas le cas, car le calcul d'intersection fine est plus long, il suffirait de consacrer d'autres étages de transputers en pipe-line au système.

Détail des calculs pour le lancer de rayons:

Nous supposons que la primitive utilisée est la facette polygonale plane.

Le calcul d'une intersection facette-rayon nécessite la résolution d'un système d'équations du type suivant:

Equation d'un rayon:

$$x = a*t + x_0 \quad (1)$$

$$y = b*t + y_0 \quad (2)$$

$$z = c*t + z_0 \quad (3)$$

Equation du plan:

$$Ax + By + Cz + D = 0$$

Soit, pour le calcul de t:

$$(a*t + x_0) + B*(b*t + y_0) + C*(c*t + z_0) + D = 0$$

soit:

$$t = (-A*x_0 - b*y_0 - c*z_0)/(A*a + B*b + C*c) \quad (4)$$

x étant calculé selon (1), (2) et (3)

Soit :

9	multiplications
7	additions
1	division

Tests d'appartenance du point d'intersection à la facette supposée triangulaire: revient à la résolution de 3 équations à 3 inconnues et 3 tests de signe ; donc:

54	multiplications
27	additions
3	tests

Soit au total: 63 multiplications	
34	additions
1	division
3	tests

En supposant que le temps nécessaire à une multiplication en mode flottant est de 300 nS, d'une division de 900 nS, d'une addition de 100 nS, nous aboutissons pour le calcul d'intersection rayon - primitive au total suivant:

25 microsecondes

Si l'on ne tient compte dans un premier temps que des rayons primaires, pour une définition de 1024 \* 1024 points d'écran et pour un nombre de 10000 facettes, le temps de calcul sera de :

$25 * 1024 * 1024 * 10000 = 72 \text{ h}$  avec 1 transputer

Si nous supposons que la cohérence spatiale est utilisée, chaque sous-espace contenant a priori de l'ordre de 2 facettes, et que chaque rayon franchisse en moyenne 20 boîtes (données expérimentales observées sur des objets tests) ce temps devient:

$25 * 1024 * 1024 * 40 = 1048 \text{ s}$

Calcul des tests de boîtes: Nous supposons que le temps de calcul élémentaire, composé de tests réalisables à partir de nombres entiers, est négligeable.

Calcul des rayons secondaires: En limitant le nombre de rayons secondaires à 3 par rayon primaire nous obtenons:

$1048 * 3 = 3144 \text{ s}$  soit : 52 mn

Avec un seul processeur.

Avec N processeurs ce temps se réduit à:

$52/N$  en minutes .

Soit, pour 100 processeurs à: 0.52 s environ ce qui est un temps tout à fait remarquable par rapport aux heures de calcul nécessaires aux images en lancer de rayons. (qui portent généralement sur quelques centaines de facettes seulement) .

### REMARQUES:

L'utilisation de la cohérence spatiale jointe à cette proposition d'architecture montre que:

*-L'ajout d'un processeur supplémentaire fait effectivement gagner un facteur de  $n/n+1$  en temps de calcul*

*-Le temps de calcul est à ce stade pratiquement indépendant du nombre effectif des facettes et ceci pour 2 raisons:*

Le nombre de boîtes à traverser ne varie que comme la racine cubique du nombre des facettes (environ  $10000^{(1/3)}$ ).

Le nombre de facettes par sous-espace est constant et égal à 2.

Ce chiffre résulte d'une étude comparative entre l'augmentation du nombre des boîtes (avec plus de tests élémentaires d'intersections de boîtes) ou l'augmentation du nombre de facettes par boîte (avec plus de tests complexes rayon-facettes).

Les chiffres cités plus haut sont des estimations de temps de calcul de l'architecture proposée, réalisés à partir des données expérimentales recueillis dans les notes : (MAD 87 CCETT 87: Thèse de MADANI).

Enfin, les estimations précédentes s'appuient sur les performances des Transputers, qui effectuent de l'ordre de 10 millions d'instructions par seconde (10 MIPS). Or en 1990 plusieurs processeurs nouveaux sont à l'étude, d'une puissance attendue de 1400 MIPS, soit 140 fois la puissance d'un seul Transputer. Les résultats des calculs précédents seront donc à améliorer dans les mêmes proportions. Ces nouveaux produits confirment l'intérêt des structures multi-processeurs de type RISC (Reduced Instruction Set Computer).

### **5-2-6/LIMITES ET AMÉLIORATIONS POSSIBLES**

Les performances des calculateurs généraux actuellement utilisés pour le rendu peuvent être, nous l'avons vu d'après les calculs précédents, améliorées dans des proportions considérables ( de 29 à 180 fois la puissance actuelle par processeur) au prix de l'utilisation de circuits spécialisés associés à chaque processeur général de type Transputer ou équivalent.

Des limites apparaîtront lorsque le débit des points calculés se rapprochera du débit correspondant au temps réel (25 images par seconde), en raison de la saturation du bus de sortie des points d'écran; Pour le reste de la machine, il n'existe à priori pas de limitation à l'accroissement de puissance, si ce n'est sa taille globale et son coût, étant donné que l'adjonction d'un processeur supplémentaire n'obère en rien les performances de ses voisins, puisque le bus 32 bits sur lequel il est alimenté en entrée est mono-directionnel. ( le processeur ne fait que RECEVOIR de l'information en même temps que ses voisins) donc il ne peut survenir les classiques conflits d'arbitrage de bus survenant lorsque ceux-ci sont utilisés en mode bi-directionnel, chaque processeur pouvant demander des informations différentes à ses voisins à un instant donné

De plus, à l'étage supérieur, un transputer collecte les résultats positifs d'intersection rayon-boîte et lance les calculs réels d'intersections en fonction de la charge du reste de la machine.

Les rayons éventuellement réfléchis ou réfractés sont ensuite réinjectés à la base de la machine et le processus recommence.

### **5-3/ CONCLUSION, LIMITES ET AVANTAGES DE LA SOLUTION PROPOSEE**

L'architecture proposée est une forme sans compromis de système de rendu modulaire, pouvant satisfaire les besoins de calcul d'image de très haute ou de moyenne qualité. (voir liste des effets pour les 3 classes retenues:

HAUTE QUALITE  
QUALITE INTERACTIVE  
TEMPS REEL

Il est clair qu'à l'heure actuelle, les besoins en rendu de qualité sont considérables et que les solutions classiques à base de calculateurs généraux de type ALLIANT, CONVEX, etc... présentent un ratio performance/prix très inférieur à celui de la solution proposée. (ANNEXE des COUTS)

Pour ce qui est du temps réel, l'architecture à retenir reste similaire, mais les bus de sortie des points constituent le principal goulot d'étranglement. Il semble donc que pour rester compétitif il faille constituer deux modules:

**UN MODULE HAUTE QUALITÉ** et qualité interactive avec 1000 transputers environ et un bus par point relativement lent. (de l'ordre de la micro-seconde), ceci pour la raison suivante: Si le calcul d'un pixel reste et restera dans les dix années à venir relativement lent en haute qualité et si l'on estime à 3 secondes l'objectif de temps de calcul de chaque image, l'on voit que ce temps est largement supérieur, et donc compatible avec le temps d'inscription des points dans une mémoire d'image: en effet: Pour une image de 1024 par 1024 le temps d'écriture d'un point est de:

$$3000000/1000000 = 3 \text{ microsecondes,}$$

ce qui très largement supérieur aux temps de stockage actuels réalisés par les cartes du commerce.

Pour la version machine de rendu de très haute ou de moyenne qualité (avec Z-BUFFER) la structure proposée n'est absolument pas pénalisée par l'écriture des points en mémoire d'image. Nous allons examiner à présent le cas de l'option temps réel:

#### **UN MODULE TEMPS - REEL:**

Comportant un bus de sortie de points multiplexé, ce qui permet d'abaisser le temps d'écriture par point: Si l'on prend un temps d'écriture dans une mémoire RAM de 200 nS, et une mémoire de 1024 par 1024 points, il faut pour la rafraîchir en temps réel (à 25 hertz) :

$$400000 / 1000000 = 40 \text{ nS}$$

L'on voit donc qu'il est nécessaire de disposer d'au moins :

$$200/40 = 5 \text{ processeurs en parallèle}$$

Chacun sortant 1 point en moins de 200 nS. Ce calcul ne tenant pas compte des temps perdus par le pipe-line lors des initialisations de boucles (initialisation d'une facette, initialisation d'une ligne) qui peuvent dans certains cas retarder la machine (ce cas se rencontre lorsque la taille des facettes à remplir est petite, ne laissant pas le temps aux étages supérieurs du pipe-line de terminer le pré-traitement (rotations clipping, etc...) avant la fin du remplissage (filling) du polygone.

Cependant si un tel bus, qui ne pose par ailleurs pas de problème de faisabilité (si ce n'est évidemment une certaine complexité et donc un coût nettement plus important que dans le cas précédent), les étages supérieurs peuvent s'inspirer de la solution proposée plus haut, sachant que l'ensemble doit respecter la formule suivante:

$$\text{Temps de calcul d'un point/Nombre de processeurs} < 40 \text{ mS}/(X_{\text{résolution}} * Y_{\text{résolution}})$$

(soit 40 nS dans le cas d'un écran de 1000 \* 1000 points)

(Ce calcul est établi en tenant compte du fait que le temps de calcul d'un point écran doit être inférieur à son temps d'affichage.)

A ce stade de l'exposé, les éléments techniques utilisés pour construire la solution sont ceux directement issus de la théorie de la synthèse d'image, et des architectures de machines classiques. Or actuellement il ne peut être ignoré une forme radicalement nouvelle de structure de machine: Cette structure est celle proposée dans les systèmes Neuro-Mimétiques (ou Neural Networks) qui font depuis peu l'objet d'intenses travaux de recherche. Nous allons examiner successivement en quoi ces systèmes peuvent influencer l'architecture des machines de synthèse d'image, ainsi que les techniques de saisie.

#### 5-4/ L'EMERGENCE ET LE LIEN AVEC LES SYSTEMES NEURO-MIMÉTIQUES

Nous avons précisé dans l'annexe 9 les bases essentielles sur lesquelles s'appuie la théorie des systèmes Neuro-Mimétiques dont une utilisation directe a été proposée au chapitre 4 pour la modélisation des objets solides 3D.

Ces techniques s'appuient au niveau matériel sur des architectures de machines d'un type nouveau, dont l'élément essentiel est le neurone simulé ou neuro-mime, assemblé en nombre important d'éléments interconnectés entre eux. De plus, la notion de neuro-mimétisme inclut un algorithme d'apprentissage à l'instar de ce qu'effectue le cerveau humain lors de sa formation.

Nous avons vu en quoi ces théories nouvelles pouvaient bouleverser les techniques traditionnellement utilisées en saisie d'objets. Or il est clair que la restitution réaliste de ces scènes peut également s'appuyer sur des machines d'architecture similaire, dont la structure serait adaptée également par l'exemple non plus sur l'analyse, mais sur la restitution des objets.

Il est concevable d'imaginer un système constitué de 3 parties, à la manière d'un automate à états finis:

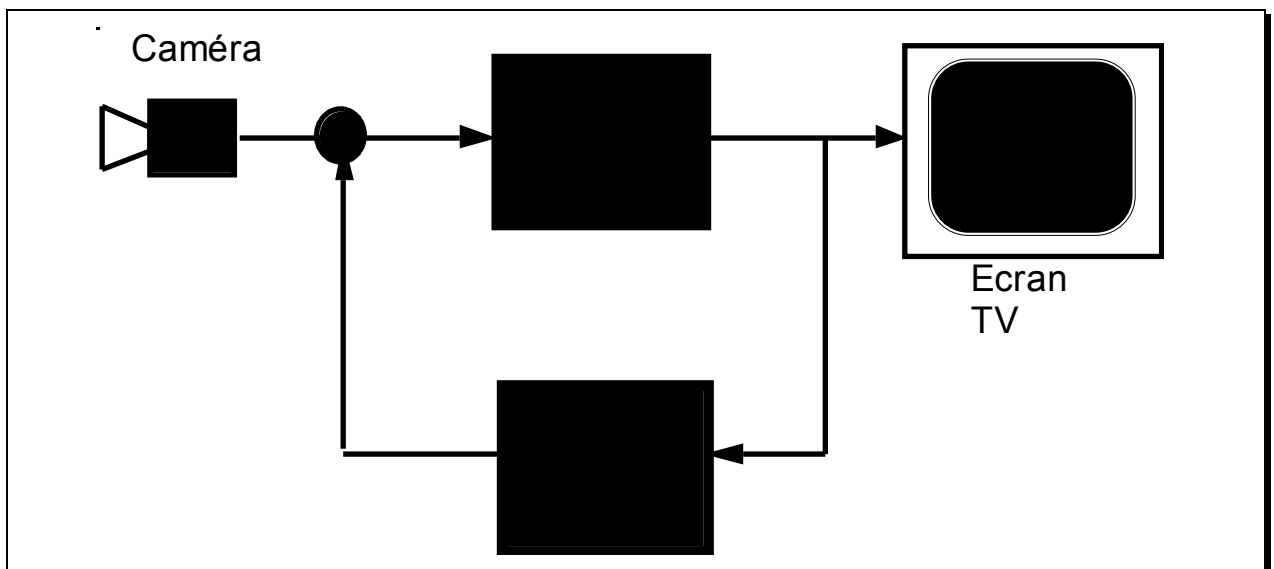


Fig 10: Machine d'analyse - restitution neuronale

En effet, un système neuro-mimétique peut être considéré comme une boîte noire recevant en entrée un vecteur (ou une matrice d'éléments linéarisée), et fournissant en sortie un vecteur, selon 2 modes de fonctionnement:

-Un mode d'apprentissage, durant lequel on fournit au système les vecteurs d'entrée et de sortie.

-Un mode de reconnaissance, durant lequel on fournit seulement un vecteur à l'entrée, le système neuronal donnant en sortie le vecteur se rapprochant le plus possible de cette entrée.

En conséquence, un tel système peut recevoir:

- En entrée la matrice de points correspondant à une image numérisée issue par exemple d'une caméra.
- En sortie, alimenter un téléviseur par l'intermédiaire d'une mémoire de trames.

Le système proprement dit comprenant une machine de type automate à états finis et composée:

- a/- D'un système neuro-mimétique adapté à l'analyse d'image 3D (décrit dans l'article HERAULT 89)
- b/- Un système neuro-mimétique orienté synthèse.

L'ensemble comparant en permanence la sortie restituée par b/ avec l'entrée analysée par a/ selon l'un des nombreux algorithmes d'apprentissage décrits dans la littérature traitant des théories neuro-mimétique.

Un tel système pourrait donc en quelque sorte "apprendre à dessiner en 3D" à la manière dont procède tout peintre ou graphiste pour reproduire une scène réelle. En effet celui-ci compare en permanence l'image de ce qu'il dessine avec les données stockées dans ses structures neuronales. Analyse et synthèse se font donc en totale symbiose, contrairement -jusqu'à présent- au processus de synthèse d'image.

## 6/ CONCLUSION GENERALE et CONSEQUENCES DES PROPOSITIONS PRECEDENTES SUR L'INDUSTRIE DE L'IMAGE

Nous avons abordé au cours de cette thèse les deux principaux points durs de la synthèse d'image:

- La Modélisation et la saisie - édition des formes d'objets à représenter*
- L'Architecture des machines de rendu, dont les performances sont loin d'atteindre les espoirs et de répondre pleinement aux besoins des utilisateurs actuels.*

Ces deux opérations ont été et sont encore, par la force des choses ( et la difficulté d'améliorer les performances matérielles) cloisonnées surtout en simulation temps réel:

- a/ Phase interactive, sur station de travail pour la modélisation*
- b/ Phase non interactive sur crayette ou architecture spécialisée pour le rendu. ( en particulier pour les applications temps-réel).*

L'objectif de cette thèse a été de proposer des améliorations en termes de coût/performance pour la phase b/, afin de l'intégrer au sein de la phase a/, comme cela apparaîtra plus clairement lorsque des machines plus performantes seront disponibles. Il est à ce stade indispensable d'anticiper quelque peu sur l'avenir, et de souligner le:

### **Lien entre la modélisation des objets et le rendu des images:**

qui conditionne la méthodologie même d'utilisation des techniques de synthèse d'image:

Assez éloignées en apparence, ces deux fonctions doivent être le plus rapprochées possibles dans les systèmes modernes, afin d'apporter à l'utilisateur un confort aussi proche que possible du modelage direct sur un volume.

Qui pourrait prétendre sculpter de la matière, si les temps de réponse aux sollicitations du sculpteur étaient de quelques minutes, voire de quelques heures?

Qui pourrait prétendre éditer un texte si le caractère ne s'affichait que quelques secondes, voire quelques minutes sur l'écran après chaque sollicitation du clavier?

C'est pourtant actuellement le cas de l'image de synthèse, où un nombre toujours plus grand d'utilisateurs créent laborieusement des formes tri-dimensionnelles à partir d'outils bi-dimensionnels. Leur visualisation effective de qualité, avec textures, effets de réflexion, etc... nécessite actuellement des heures, voire des jours à de puissants calculateurs.

L'émergence de solutions industrielles mettant en oeuvre les idées originales décrites en terme de modélisation et d'architecture aux chapitres 4/ et 5/ permettront à l'utilisateur de disposer grâce à des systèmes de type serveurs d'image, de tous les effets spéciaux joints à une rapidité de calcul voisine du temps réel.

Il est prévisible que de telles architectures parallèles dotées de capacités de rendu en temps réel, et associées à des systèmes de saisie directe dans l'espace devraient



faire évoluer les méthodologies traditionnelles des bureaux d'études, à base essentiellement de plans.

La conception des objets (de type CAO par exemple) se fera de plus en plus directement en représentation volumique, comme cela commence à se pratiquer aux USA.

Le couplage à des machines outils à commande numérique permettrait également une vision rapide des formes ainsi conçues (notamment sur des objets tests en plastique). Il existe déjà aux Etats-Unis et en Europe des systèmes de synthèse directement couplés à des machines outils de découpe de plastique. Déjà l'on voit apparaître des pièces en plastique directement fabriquées à partir d'un poste de CAO.

Un progrès considérable reste cependant à faire sur le plan des matériels, en particulier en ce qui concerne:

- Les temps de réponse du calcul d'une image
- La qualité du rendu
- Le coût des matériels, lié au temps de production de chaque image.

(Les systèmes actuels de rendu à base de crayettes n'autorisent qu'une production annuelle de 5 à 10 mn animées, à raison de 30mn par image calculée ce qui est notoirement insuffisant pour sortir l'image de synthèse de sa paradoxale marginalité actuelle ).

Une fois ces étapes franchies, grâce au progrès technologique, le bureau d'étude traditionnel pourrait se transformer en centre de modelage tri-dimensionnel, où les notions de plans de cotation seraient remplacés à tous les stades allant de la conception à la production des objets par une visualisation directe associée -et c'est cela l'essentiel- à une possibilité d'interaction et d'édition dans l'espace tri-dimensionnel.

Déjà l'on constate l'émergence en Californie de sociétés de modélisation dont la tâche est de numériser à l'aide d'un maillage surfacique tout type d'objet solide tri-dimensionnel.

Les besoins sans cesse croissants de la conception d'objets industriels devraient avoir recours dans des proportions grandissantes à ce type de technique.

Il serait très long de recenser les nombreux domaines où la synthèse des images intervient déjà, ou pourrait, dans un avenir proche, intervenir:

Citons cependant, dans les 3 secteurs fondamentaux:

*La CAO*

*L'Audiovisuel*

*La Simulation*

l'émergence d'applications entièrement nouvelles pour ce genre de technique:

- Représentation de macro-molécules
- Conception de vêtements ou de drapés en synthèse d'image
- Visualisation de trajectoires de satellites au moment de leur déploiement sur orbite.

- Visualisation des champs de vitesses dans la chambre de combustion d'un moteur à explosion .
- Modélisation de trajectoires de mobiles en simulation.
  
- Visualisation du comportement de bancs de poissons en temps réel (Vivarium project d'APPLE )
- Portraits robots
  
- Chirurgie esthétique avec modélisation par synthèse
  
- Soins dentaires avec visualisation de prothèses en images réalistes.
  
- Etc...

Enfin, l'analyse des images, processus inverse de la synthèse, pourrait à terme converger pour les problèmes immédiats de la transmission avec réduction de débit, mais surtout pour l'analyse des images en général.

La synthèse, en effet constitue le système de compression idéal pour réduire la base de données des objets en vue de leur transmission: Prenons, par exemple une sphère: sa transmission sous forme de base de données comporte quelques octets seulement, pour définir:

- Le rayon
- Le centre
- La couleur

Si nous rapprochons ces 3 nombres aux octets nécessaires à la représentation des points qui décrivent cette sphère, nous nous apercevons qu'un très fort taux de compression peut être atteint en n'envoyant que les 3 paramètres définis ci-dessus. Tout récemment, des centres de recherche Japonais et US travaillent à des études de compression reposant sur l'analyse de scènes 3D (modélisation des objets), et leur reconstruction sur un terminal distant après transmission.

Ainsi commence à se faire ce pont naturel entre analyse (modélisation ) et synthèse, par le biais de la demande de compression des images liée aux problèmes de télécommunications. Il est à prévoir que les apports théoriques et techniques issus d'une possibilité nouvelle et directe de modélisation apportera d'importants changements non seulement dans les techniques de codage, mais également dans celles de la synthèse d'image, et également de la robotique, en particulier pour le pilotage automatique de mobiles.

Difficile procesus de création à l'état pur, la synthèse et la modélisation d'objets est une science en pleine évolution, ce qui en fait tout à la fois son intérêt et son caractère universel car touchant de plein fouet toutes les formes de la conception issue de l'esprit humain.

## ANNEXE 1: LE TRANSPUTER

Le transputer est un circuit VLSI comportant à l'intérieur d'un même circuit:

- Une unité centrale de type classique dotée d'un jeu d'instructions de type RISC (Reduced Instruction Set Computer). Sa puissance en arithmétique entière est de 10 Mips (10 millions d'instructions par seconde).
- Une mémoire locale rapide de 4K octets, pouvant être complétée par une mémoire externe de plusieurs méga-octets.
- Un module de calcul en arithmétique flottante effectuant en particulier les opérations d'addition/soustraction en 200 nS et de multiplication/division en 400 nS environ.
- Un ensemble de "liens" série permettant d'effectuer des transferts d'information rapides avec d'autres Transputers à 20 Mbits par seconde.

Ce composant dispose actuellement d'un langage de programmation parallèle qui lui est propre, le langage OCCAM, et de plusieurs compilateurs C.

**ANNEXE 2: SCHEMA SYNOPTIQUE DU TRANSPUTER**

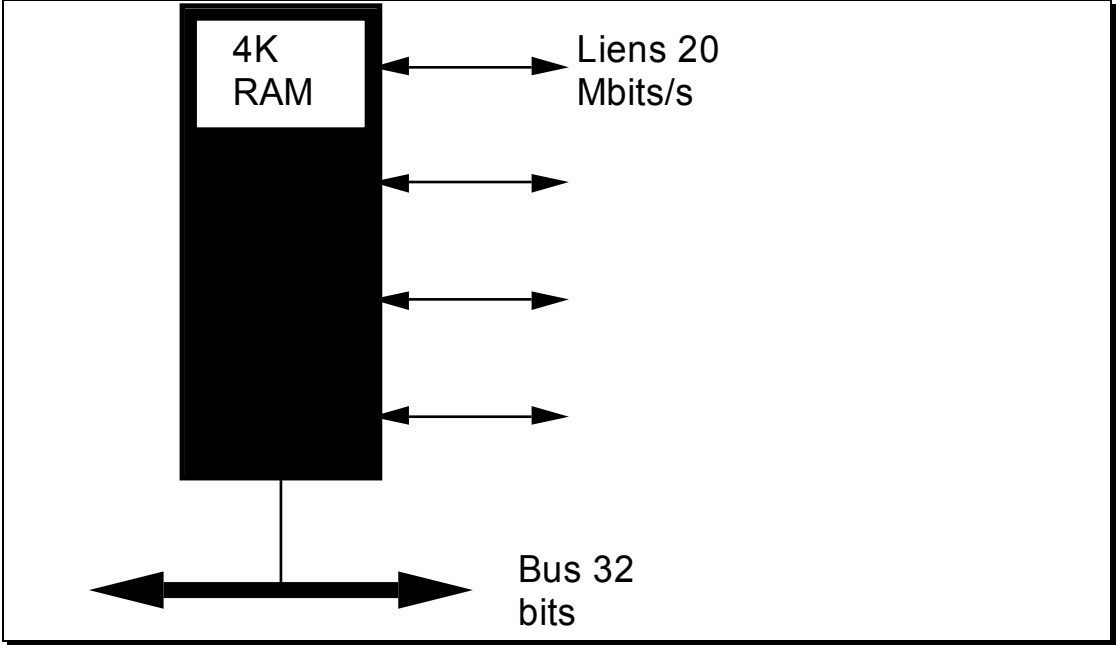


Fig 11: TRANSPUTER à Architecture RISC (Reduced Instruction Set Computer)

### ANNEXE 3 : TECHNIQUES DES PRIORITES

La technique des priorités a constitué vers les années 70-75 la première technique véritablement efficace de synthèse d'image pour la simulation. Cet algorithme d'élimination des parties cachées repose sur le fait que les objets à représenter sont fixes les uns par rapport aux autres, et qu'un opérateur se déplace à l'intérieur de l'ensemble formé par ces objets.

Cette méthode est donc très adaptée à la simulation de vol où un paysage solide par rapport à lui-même doit être visualisé en évolution par rapport à un pilote dans le cockpit de son mobile (avion, char, voiture, etc...).

L'intérêt essentiel de ce procédé est qu'il se présente en 2 phases:

La première consiste à précalculer en temps différé les différentes priorités des entités composant la scène à représenter: polyèdres, polygones, etc...

Pendant la phase temps réel d'exploration de la scène par l'opérateur, il suffit d'afficher par exemple dans la mémoire d'image la liste des polygones triés afin d'obtenir l'élimination des parties cachées des objets.

De plus, l'implantation cablée de l'anti-aliasing est dans ce cas très aisée, puisque pour tout point d'écran ( $X_i, Y_i$ ) la machine connaît la liste des priorités des facettes en ce point :

Soit un pixel P de surface unité;

Soit S1 la surface visible occupée par le polygone P1 d'ombrage O1 sur P.

Soit S2 la surface visible occupée par le polygone P2 d'ombrage O2 sur P.

La contribution de P1 à l'ombrage de P sera:

$$Op_1 = S_1 * O_1$$

Par conséquent, la contribution de P1 et P2 à l'ombrage de P sera:

$$Op_{1p2} = S_1 * O_1 + S_2 * O_2$$

SI un polygone P3 d'ombrage O3 recouvre partiellement la surface restante, la contribution de P1, P2, P3 à l'ombrage de P sera:

$$Op_{1p2p3} = S_1 * O_1 + S_2 * O_2 + S_3 * O_3$$

L'on voit donc que cette méthode conduit à un anti-aliasing mathématiquement rigoureux et facilement implémentable au niveau matériel (bien que cher puisque ces calculs doivent être faits en 10 nS par points d'écran en  $1024 * 1024$  en raison de la fréquence de balayage).

Dans le cas du tampon en Z, cette méthode ne peut être appliquée sans précaution, car les facettes arrivent dans le désordre. Si l'on suit l'algorithme, il peut

arriver que la face  $P_3$ , par exemple recouvre presque entièrement le pixel: en conséquence,  $O_{p_1p_2p_3}$  contiendra des contributions de  $O_1$  et  $O_2$  qui en fait sont cachés. C'est ainsi que l'on peut apercevoir dans de mauvaises implémentations des couleurs de facettes en fait cachées sur les arêtes.

**ANNEXE 4: INTERSECTIONS DE VOLUMES et MAILLAGES***Les techniques d'intersection de volumes:*

Elles consistent à calculer géométriquement l'intersection, la fusion ou la différence de 2 ensembles d'objets. Dans le cas d'une représentation par facettes, ce traitement réalise l'intersection de tous les polygones des objets par toutes les arêtes, trouve les arêtes nouvelles engendrées par l'intersection, et supprime les parties d'arêtes modifiées par cette dernière. Le Docteur Encarnaço de l'Université de Darnstadt a très clairement explicité le mode de calcul des lignes d'intersection entre objets.

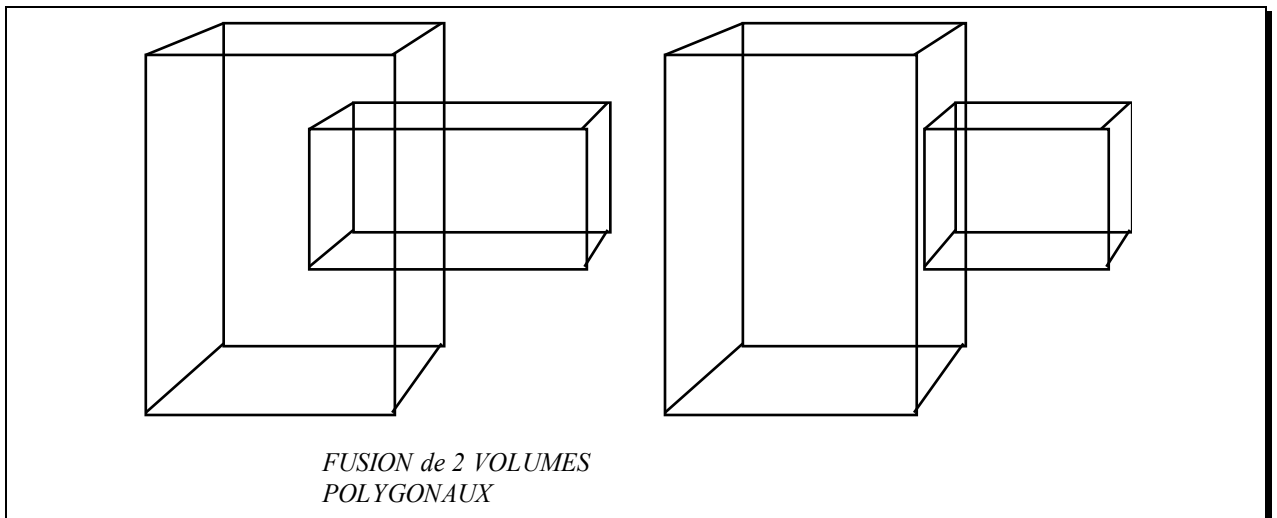
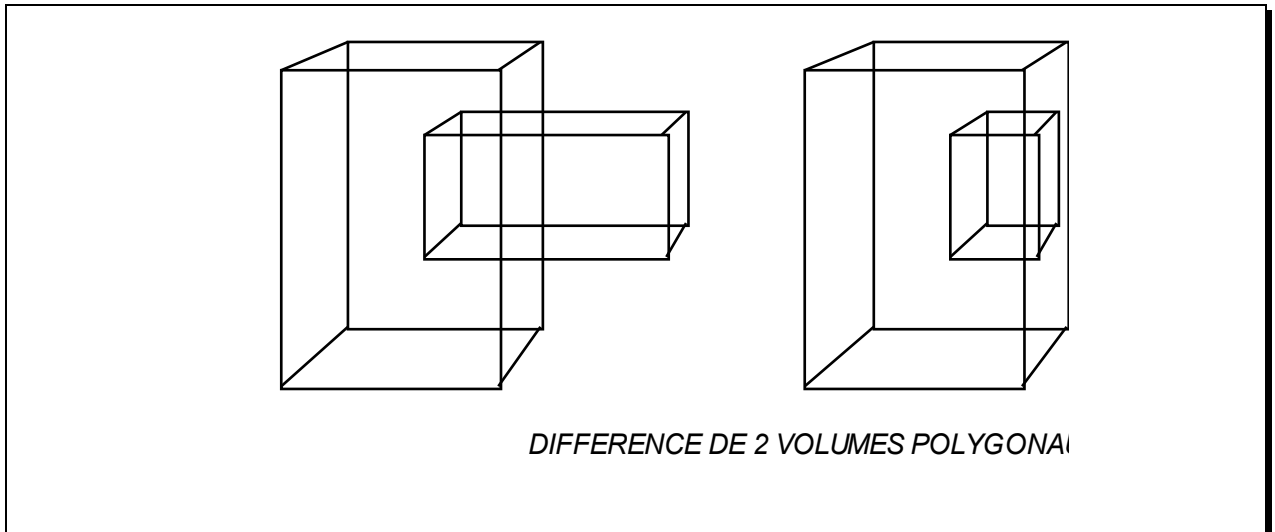
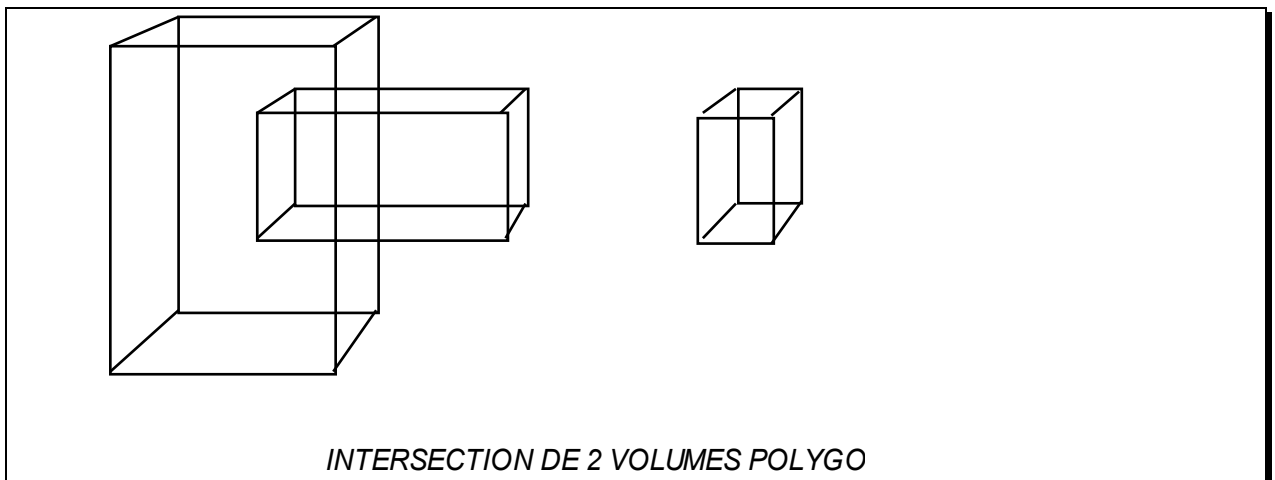
*Cas de 2 parallélépipèdes s'interpénétrant:*

Fig 12: Fusion de 2 volumes polygonaux



*Fig 13: Différence de 2 volumes polygonaux*



*Fig 14: Intersection de 2 volumes polygonaux*

*Les techniques de maillages consistent à générer un ensemble de polygones plans approximant de façon optimale un nuage de points dans l'espace, supposé limiter un volume.*

*D'innombrables travaux ont été réalisés sur ce point: il serait impossible de les recenser exhaustivement: Ce problème intervient en effet au cours de la numérisation des objets à partir de systèmes laser, qui ne fournit que les coordonnées d'un ensemble de points dans l'espace. Les méthodes les plus classiques consistent à regrouper les points par ensembles de 16 (4\*4) par exemple, ces points jouant le rôle de points de contrôle d'une surface mathématique qui approxime le nuage de points. (Surfaces de Bézier, de Coons ou d'Hermite).*

*Le problème fondamental réside dans la minimisation du nombre de facettes finales: Les travaux les plus significatifs sont ceux de F. SCHMITT de l'ENST Paris, et d'ALAN BARR au CALTECH, précédés de ceux de Bernstein et de Voronoy pour la triangulation.*



***Extraits de "ADAPTIVE POLYHEDRAL APPROXIMATION OF DIGITIZED SURFACES" F. SCHMITT 1985***

**ANNEXE 5: IMAGES DE SCULPTURE ELECTRONIQUE SUR ECRAN**

## ANNEXE 6 : ETUDES ET DEVELOPPEMENTS PERSONNELS SUR L'INFOGRAPHIE

### PRESENTATION CHRONOLOGIQUE

Les recherches et réalisations que j'ai menées dans le domaine du graphique par ordinateur se décomposent chronologiquement en 6 phases:

- Etudes sur les calculateurs synchrones à la Radiotechnique en technologie TTL (en 69).*
- Etude et réalisation du logiciel graphique GRAFOR (FORtran-GRaphique) sur calculateur CII 10070 et ensemble graphique SINTRA.*
- Etude et réalisation d'un système de génération d'images de synthèse temps-réel GSI au CELAR.*
- Conception et réalisation de la machine polyvalente et modulaire CUBI 7 pour la synthèse des images dans le cadre de mon laboratoire d'étude et de recherche au CCETT (Centre Commun d'Etude de Télédiffusion et Télécommunication)*
- Etude des réseaux neuronaux au CCETT pour le codage, la compression des images de télévision. Réalisation d'un simulateur neuronal sur ordinateur pour la reconnaissance de formes.*
- Etude sur les fractales et les fonctions itératives*

Nous détaillons dans les 3 tableaux ci-dessous les différents travaux soit traités directement par moi-même, soit menés sous ma direction dans mon Laboratoire, soit menés à partir d'idées développées dans mon Laboratoire.

### ETUDES ET REALISATIONS PERSONNELLES:

- Réalisation d'un additionneur binaire sur 6 bits en série FH, avec calcul de la retenue en 6 nS (1969)
- Logiciel de visualisation graphique structuré GRAFOR au CELAR: conception complète du système d'interruption, et écriture du logiciel, ainsi que du méta-assembleur sur 10070.
- Nouveau théorème sur les racines des polynômes de degré n-application à une nouvelle méthode de calcul du nombre PI
- Structure bloc de la machine GSI du CELAR
- Conception et réalisation du jeu de micro-instructions graphiques de la machine de synthèse d'image GSI au CELAR
- Structure Bloc de la machine CUBI 7 au CCETT
- Structure Bloc de la première architecture de machine à tampon en Z, réalisée ensuite matériellement par Sogitec
- Macro-générateur en APL
- Logiciel graphique de création de formes et d'animation en APL sur CUBI7 et sur MAC-INTOSH.
- Compilateur APL
- Simulateur Neuronal sur ordinateur appliqué à la reconnaissance de formes ou de caractères
- Idée et étude d'association de la technique du "shape from shading" (ou extraction de formes à partir de l'ombrage) avec les facultés d'apprentissage des systèmes neuronaux pour l'analyse d'image, du mouvement des objets et pour la compression d'image. (Etude en cours)
- Idée et étude d'association des Réseaux neuronaux aux fractales pour la compression d'images. (Etude en cours)
- Images fractales à partir de fonctions itératives pour la modélisation fractale des objets contenus dans une image. (Etude en cours)

### ETUDES et REALISATION FAITES SOUS MA DIRECTION OU DANS LE CADRE DE MARCHES D'ETUDE EXTERIEURS A PARTIR D'IDEES ALGORITHMIQUES QUE J'AI MOI-MEME INITIEES

- Modélisation directe dans l'espace d'objets 3D à partir d'un numériseur sonique. (marché CCETT GRAVI)
- Réalisation matérielle de la machine de synthèse d'image GSI au CELAR
- Réalisation matérielle de CUBI 7
- Ecriture du logiciel système de CUBI 7

**ETUDES REALISEES SOUS MA DIRECTION A PARTIR D'IDEES  
DEVELOPPEES AU SEIN DE MON LABORATOIRE**

- Incrustation en Z d'objets réels dans une scène synthétique ou inversement (Z key).
- Structure de machine de type lancer de rayons.

### **7-1/ ETUDE SUR LES CALCULATEURS SYNCHRONES DANS LE CADRE DE LA SOCIÉTÉ RADIOTECHNIQUE EN 1969:**

J'ai effectué dans cette Société un stage de 4 mois au cours duquel j'ai mené à bien les réalisations suivantes:

- Participation à la conception, à la réalisation et à la programmation du premier ordinateur digital programmable de LOGABAX, société pour laquelle la Radiotechnique avait un contrat d'étude. Dans le cadre de ces travaux, j'ai pu approfondir mes connaissances en informatique au niveau matériel et logiciel.

Ce ordinateur était réalisé entièrement en logique synchrone en boîtiers DTL. Il comportait 4 cartes de 20 par 30 cm environ:

- Une carte horloge
- Une carte de séquençage
- Une carte unité arithmétique
- Une carte interface extérieur (périphériques imprimante, mémoire, etc...)

Par ailleurs j'ai expérimenté un terminal alphanumérique IBM relié à un IBM 360, sur lequel j'ai développé plusieurs programmes (notamment une méthode originale de calcul du nombre  $\pi$ , basé sur un nouveau théorème sur les équations de degré N). Après un stage à la CII aux Clayes sous bois pour la réalisation d'un implanteur sur ordinateur MITRA 15, la seconde phase de mes travaux s'est réalisée dans le cadre du CELAR: Centre d'Electronique de l'Armement situé à Bruz près de Rennes.

### **7-2/ ETUDES et DEVELOPPEMENTS GRAPHIQUES REALISES AU CELAR**

Ma première mission dans ce Centre a consisté à réaliser le logiciel graphique système d'un ensemble VU2000 dont la société SINTRA avait réalisé la partie matérielle.

Cet ensemble de haute performance pour l'époque (chaque console valait environ 1MF en Francs 72) ne disposait pas de logiciel graphique standard sous forme de modules appelables depuis un ordinateur hôte. Les applications, toutes spécifiques étaient écrites en assembleur. Elles étaient surtout destinées à la surveillance aérienne militaire ou civile.

Mon rôle a consisté à écrire une bibliothèque de programmes graphiques système (c'est à dire vus comme une librairie) callable par tout utilisateur programmant en Fortran sur le ordinateur hôte connecté à l'ensemble VU2000. De plus, le Fortran ne disposant pas de primitives de gestion de structures, j'ai inclus dans ma bibliothèque des commandes pour la gestion d'éléments graphiques.

Ce logiciel que j'ai baptisé **GRAFOR** ou **GRAPHIQUE FORtran** comportait 3 parties:

- Les ordres graphiques proprement dits
- Les ordres de gestion des structures
- Les ordres d'interface avec le ordinateur hôte

#### **7-2-1/L'ES ORDRES GRAPHIQUES**

Ces famille d'ordres contenaient les commandes de tracé sur écran:

## SUBROUTINE TRAIT(X,Y,N,k)

où l'on pouvait afficher un trait selon la valeur de k:

-De la position courante à X,Y en coordonnées absolues

-De la position courante à X,Y en coordonnées relatives

-si N était différent de 1, X Y était un tableau de dimension N, 2 (N points à 2 coordonnées) comportant un ensemble de traits à afficher (polyline).

## SUBROUTINE MARK(X,Y,N,K)

Ce sous-programme affichait des caractères de marquage selon les mêmes critères que pour TRAIT.

## SUBROUTINE TEXT(text,n,x,y,k)

permettait d'afficher un texte de n caractères contenus dans le tampon de caractères alphanumérique : "text".

**7-2-2/LES ORDRES DE GESTION DES STRUCTURES**

Ces ordres constituaient l'une des originalités de ce logiciel. Je faisais partie à l'époque d'un groupe graphique de l'association française AFCET qui réunissait les principaux centres de recherche en graphique français: Y étaient représentés:

L'Université de Grenoble

L'Université de Lille

L'Université de TOULOUSE

Les Sociétés THOMSON, SINTRA et ALCATEL, seuls industriels actifs dans ce domaine à l'époque.

Le Centre d'Electronique de l'Armement que je représentais.

L'IRIA (avant sa dénomination en INRIA)

L' ETCA (Etablissement Technique Central de l'Armement)

Les experts considéraient à l'époque que les notions de structure devaient être incluses dans le langage standard et non dans la librairie graphique. Beaucoup croyaient au succès d'ALGOL 68 qui effectivement contenait d'intéressants concepts, mais dont la suite industrielle fut un échec total.

Or le seul langage de programmation était le FORTRAN (ni le C ni le PASCAL n'existaient encore); en conséquence, les utilisateurs de graphiques où ces notions de structures hiérarchiques jouent un rôle important en étaient généralement réduits à ré-écrire eux-mêmes leurs propres programmes de gestion des structures.

J'ai donc décidé d'inclure dans ce logiciel, les primitives de définition de structures suivantes:

DEFELN(nom1, nom2,... , nomN)

Procédure qui définit N noms d'éléments dont la valeur logique est retournée par le système graphique. Ces noms constituaient les éléments terminaux des graphes de structures qu'un utilisateur pouvait définir.

DEFNOEUD(noeud1, noeud2,..., noeudN)

Permettait de définir N noeuds du graphe de l'utilisateur. La particularité de ce graphe à 1 niveau était que chaque noeud pouvait contenir d'autres noeuds, et que des éléments ou des ensembles d'éléments pouvaient être rattachés à un ou plusieurs noeuds.

IAPPN(NOEUD, e1, e2,... eIn)

Définissait l'appartenance des éléments e1, e2, ..., eIn au neud de nom "NOEUD".

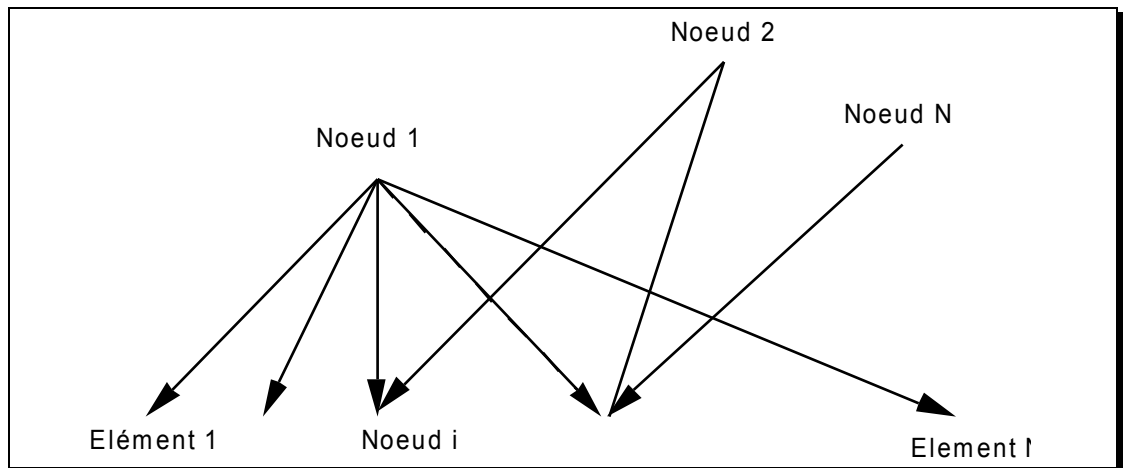


Fig 15: Structure de donnée de GRAFOR

L'ordre de base était:

IAPPN(noeud, ELi | NOEUDn)

Qui testait l'appartenance de l'élément ELi ou du noeud NOEUDn au noeud "noeud". Ce test était utilisé quand l'opérateur désignait sur l'écran une entité par l'ordre:

IELT(Noeud, ELT)

et recueillait le numéro de noeud et d'élément désigné.

Enfin l'ordre le plus utilisé était :

Noeud = ISOM ( Noeud1, OPE1, Noeud2, OPE2, Noeud3, ...)

qui effectuait les opérations OPEi aux différents noeuds ou éléments désignés. OPEi pouvait être:

Une opération de type:

ET (Intersection d'ensemble d'éléments ou de noeuds)

OU (Union)



NON (Négation)  
 ETNON (ET suivi de NON)  
 OUNON (OU suivi de NON)  
 IN (Affectation)

La structure de données utilisées par ces commandes reposait sur des chaînes de bits pour représenter les noeuds et les éléments. Des programmes en assembleur puis en FORTRAN permettaient par de simples opérations logiques booléennes sur N bits d'effectuer les opérations décrites ci-dessus.

La structure par champs de bits permettait de gérer directement par mot de 32 bits 32 éléments ou noeuds, et d'effectuer une opération logique directement sur 32 bits en une seule instruction.

### 7-2-3/LES ORDRES D'INTERFACE AVEC LE CALCULATEUR HÔTE

Ces ordres permettaient de gérer le dialogue avec le terminal graphique: Ce dernier était constitué de la façon suivante:

*-a/ Un calculateur 16 bits appelé ET 1016, calculateur développé par SINTRA dont les rôles étaient les suivants:*

- Recevoir les données extérieures issues du calculateur hôte
- Transmettre les messages issus de l'opérateur.
- Gérer les interruptions issues du système graphique:

Ces interruptions étaient les suivantes:

Interruption du clavier alphanumérique

Interruption du clavier numérique

Interruption de la boule roulante

Interruption du light-pen

Interruption d'entrée-sortie extérieure.

Interruption du processeur de visualisation.

-Piloter l'affichage des listes de visualisation.

-Commander les périphériques d'entrée et de sortie du système graphique  
 (Claviers, Boule roulante, Light-Pen )

Ce calculateur possédait de l'ordre de 70 instructions.

*-b/ Un système graphique qui était en fait un véritable calculateur qui effectuait les tâches suivantes:*

- Lire la liste de visualisation comportant les ordres graphiques à exécuter:
- Positionnement du spot

-Affichage de traits

-Affichage des caractères de marquage

-Affichage des caractères.

-Instructions de branchement, de branchement avec retour à un sous-programme graphique.

-Instructions de lancement et d'arrêt du processeur graphique.

Générer effectivement les vecteurs et les caractères sur un écran graphique à pénétration noir et blanc ou couleur. Ce système comportait un système de décodage des instructions graphiques, et les amplificateurs de puissance commandant :

- La déflexion du tube
- Le wehnhelt.

Ce système était livré nu sans aucun logiciel.

Mon rôle a consisté à concevoir et à réaliser un système temps-réel pour ce calculateur, c'est à dire:

- Prenant en charge les différentes interruptions
- Assurant toutes les fonctions de dialogue temps-réel:
  - Poursuite du light-pen
  - Désignation des éléments
  - Edition des textes frappés sur claviers
  - Gestion des indicateurs lumineux du clavier numérique
  - Dialogue de transmission et de réception des messages avec le calculateur hôte (CII 10020 d'abord puis CII 10070).

J'ai développé un système temps-réel multi-taches dont le principe était le suivant:

-Pour la gestion des interruptions: Sur ce système, les interruptions n'étaient pas empilées: En conséquence lorsqu'une interruption de même type (par exemple, frappe d'un caractère au clavier) survenait lorsque le calculateur traitait déjà un même niveau, la machine s'arrêtait. Il fallait donc, en conséquence:

-Réaliser une tâche de fond de priorité inférieure à toutes les interruptions.

-Faire des séquences d'interruptions les plus courtes possibles mettant en mémoire les éléments d'information reçues (par exemple: type d'organe ayant émis l'interruption (ex: clavier), la valeur transmise: par exemple: code Ascii frappé) et ressortant aussitôt .

La tâche de fond mettait en pile la demande reçue, et l'exécutait lorsqu'aucune autre interruption ne survenait.

Je n'insisterai pas sur les nombreuses difficultés matérielles qui jalonnèrent cette réalisation qui dura environ 6 ans de 1970 à 1976 , difficultés essentiellement dues à des problèmes hardware (Défauts de conception du matériel, mauvais contacts, absence de tout logiciel: il existait un assembleur mais il fallait plus d'un quart d'heure pour le charger en mémoire sur télétype, et presque autant de temps pour l'éditer, logique non synchrone avec triggers déréglés, portes d'armoires qui changeaient par effets magnétique le déroulement opératoire de certaines instructions selon qu'elles étaient ouvertes ou fermées, etc...)

En particulier, en raison de ce manque de fiabilité, j'étais contraint d'insérer dans ma tâche de fond une fonction de surveillance de la mémoire qui testait en permanence la parité des 16 K mots de mémoire dont disposait la machine. C'est à dire qu'à chaque boucle de la tâche de fond, je venais tester un mot mémoire dont j'incrémentais ensuite l'adresse.

Le système comportait également des tâches opératoires telles que:

**-La gestion du crayon lumineux ou light-pen:**

Celui-ci ne délivrait en effet au niveau du matériel qu'une interruption lorsqu'il "voyait" un trait sur l'écran dans le champ du crayon lumineux. Et la visualisation s'arrêtait aussitôt.

Il fallait pour calculer les coordonnées d'un point sur écran:

-Lancer des suites de points dans les 4 directions autour de la position précédente du light-pen x y.

-Déduire des interruptions reçues ainsi que du rang du trait vu par le light-pen :

Quel sens de déplacement a effectué le light-pen depuis sa position précédente, et estimer une position future x',y' en fonction de la vitesse estimée du Light-pen. Ce programme temps-réel complexe, assimilable à un asservissement avec boucle de retour en position et correction en vitesse a été réalisé en assembleur. Sa difficulté de mise au point était grande puisqu'il n'existait à cette époque aucun outil de test en temps réel. En cas d'erreur la machine se "plantait" sans retourner la moindre information.

Avec ce programme je pouvais recueillir les informations suivantes:

Détermination de la position courante x, y du light pen, avec poursuite automatique du spot devant ce dernier, en fonction des sollicitations d'un opérateur.

Etant donné une image déjà tracée, détermination de l'entité logique désignée par l'opérateur. A un niveau supérieur, (voir ci-dessus) les fonctions IELN donnaient au programmeur sur le calculateur hôte (après transmission par mon système) accès à l'entité désignée et au(x) noeud(x) dont elle dépendait.

**-La gestion de la boule-roulante :**

Celle-ci était dépourvue de toute logique câblée délivrant les simples informations x et y, maintenant disponibles sur les équipements récents. Son principe consistait à émettre une interruption chaque fois qu'elle était sollicitée: L'on pouvait alors compter le nombre d'incrémentés effectués depuis la position précédente.

**-La gestion de la liste d'affichage**

Cette tâche était conçue pour arrêter et relancer soit de façon automatique, soit sur ordre du calculateur hôte l'affichage de l'image. En effet l'écran utilisé était de type à pénétration, c'est à dire que le spot ne passait qu'aux endroits définis par la liste de visualisation : C'est à dire que si l'on affichait sur l'écran une image simple, le processeur graphique, peu chargé, pouvait la visualiser souvent. Au contraire si l'image était chargée, le spot repassait moins souvent sur l'image.

Il était donc nécessaire de gérer une interruption spéciale, appelée interruption image, dont le rôle était de signaler au calculateur le passage du spot en un point donné de la liste d'affichage.

Le calculateur hôte calculait alors une temporisation durant laquelle le rafraîchissement (ou réécriture) de l'image était stoppé, puis passé ce délai, relançait l'affichage selon un cycle défini. Le problème se compliquait lorsque la durée nécessaire au rafraîchissement excédait la temporisation. L'ensemble était paramétrable depuis le calculateur hôte.

#### **7-2-4/CONCLUSION**

Malgré ces difficultés l'ensemble logiciel GRAFOR a été utilisé au CELAR environ 10 ans de 71 à 81 (même après mon départ de cet organisme en 81) par une dizaine d'utilisateurs.

Sur GRAFOR ont été développées de nombreuses applications militaires opérationnelles qui ont donné lieu à des démonstrations en présence de responsables des armées. (Les premières simulations du système RITA au CELAR ont été réalisées avec GRAFOR)

(Le réseau RITA a fait ultérieurement l'objet d'une commande de l'armée Américaine à THOMSON/CSF.)

#### **7-3/ MES RECHERCHES EN SYNTHÈSE D'IMAGE AU CELAR**

##### ***INTRODUCTION:***

Depuis 1974, je pressentais l'importance croissante des techniques graphiques pour les besoins industriels, et notamment celles permettant la présentation réaliste d'objets 3D sur écran à balayage de type télévision.

L'expérience que j'avais acquise sur les écrans à balayage cavalier de type VU2000 (voir plus haut) m'a révélé l'existence de contraintes assez importantes:

- Au niveau du rendu des surfaces:* Un système à balayage cavalier en effet possède une liste de visualisation (ou Display List ); c'est à dire un ensemble d'ordres graphiques de positionnement, de tracé, de chaînes de caractères, etc... qui sont interprétés et exécutés par un processeur graphique qui en assure l'affichage. Ce qui signifie que si l'on désire obtenir une surface sur un écran à balayage cavalier, l'on doit stocker dans la liste d'affichage l'ensemble des vecteurs de hachurage constituant cette surface. La liste d'affichage (alors limitée à 16 Koctets) se sature très vite si l'on souhaite obtenir plus de 10000 traits.
- Au niveau de la couleur:* Le tube utilisé est un tube à pénétration, c'est à dire que les différentes teintes sont obtenues en faisant varier les tensions d'accélération du faisceau électronique qui par son énergie atteint ou non les couches les plus internes des phosphores déposés sur l'écran. En général, 3 couches de phosphores sont déposés sur le fond des tubes: Rouge, marron et vert.
- Au niveau des nuances de couleur:* Toute variation d'intensité change en fait la couleur, puisqu'elle fait varier la couche de phosphore excitée par le faisceau électronique. En conséquence, seule une très faible variation de tension est autorisée dans une teinte donnée si l'on souhaite rester dans la même couleur.

Au contraire, dans le cas d'un tube à balayage associé à une mémoire d'image il n'y a pas de limitation de teintes puisque sur un tube à balayage la teinte est obtenue par le pourcentage des faisceaux Rouge Vert et Bleu, l'intensité par la somme des trois. Ces deux grandeurs sont donc -du moins en première approximation- orthogonales.

En revanche, le tube à pénétration associé à une liste de balayage possède un avantage: celui de pouvoir désigner sur l'écran une entité, et de pouvoir recueillir

directement le numéro logique de cette entité étant donné que le processeur sait, lorsqu'il reçoit l'interruption light-pen, quelle entité il est en train d'afficher. Dans le cas du tube à balayage, cette information a longtemps été considérée comme impossible à obtenir:

Ce procédé repose sur l'utilisation d'une mémoire d'image associée à l'écran TV. Il permet la reconnaissance directe par la machine des structures désignées par un opérateur humain devant son terminal. Sa description précise sort du cadre de cette thèse.

### ***RECHERCHE SUR LA SYNTHÈSE DES IMAGES AU CELAR:***

Mes premières recherches remontent aux années 1974: A cette époque, les seuls systèmes disponibles sur le marché étaient des terminaux graphiques à balayage cavalier dominés de très loin par la société US TEKTRONIX. Cependant commençaient à apparaître des mémoires d'images mais celles-ci étaient dépourvues de processeurs spécialisés en graphique de type Z-BUFFER ou processeurs géométriques maintenant bien connus.

Tout dans ce domaine était à découvrir y compris le vocabulaire technique et les fonctionnalités même de ce que l'on voulait réaliser.

La phase 1 de mes recherches a consisté à réaliser avec des stagiaires un multiplieur matriciel pour effectuer les rotations d'objets dans l'espace de façon rapide; Ce projet a abouti à la réalisation d'une maquette mais la technologie de l'époque nécessitait le volume d'un coffret 19 pouces alors qu'aujourd'hui un boîtier 40 broches en technologie VLSI suffit.

La phase 2 a consisté à étudier des algorithmes d'élimination des parties cachées pour la représentation réaliste 3D. (ou Rendering Problem).

J'ai donc étudié et implémenté les algorithmes de rendu disponibles dans la littérature scientifique aux USA. En particulier l'algorithme de WATKINS.

La conclusion de mon étude fut que cet algorithme était irréalisable sur le plan matériel et que la clé de l'image de synthèse se trouverait dans une étroite imbrication entre un matériel et un logiciel, sachant que cette technique exigeait des puissances de calcul que l'on ne peut encore aujourd'hui atteindre. (Il faut encore plusieurs heures de calcul aux matériels actuellement les plus performants pour obtenir une image de synthèse comportant tous les effets spéciaux...).

En fonction de cette expérience, j'ai étudié la voie des algorithmes dits à priorité, bien adaptés à la synthèse d'images pour la simulation, puisque reposant sur le fait que les objets à représenter sont fixes les uns par rapport aux autres.

La méthode et le logiciel que j'ai obtenu permettait de visualiser des solides tels que des avions, des chars, de l'ordre de 120 facettes en rendu réaliste avec ombrage lisse.

Ce logiciel reposait sur l'idée suivante: Au lieu de subdiviser les objets à l'aide de plans séparateurs, je subdivisais l'espace en sous-régions définies par des angles solides jusqu'à ce qu'un graphe de priorités non cyclique de polygones soit calculable à l'intérieur de cette région. En cas d'échec, (c'est à dire si le graphe est encore cyclique)

on subdivise à nouveau l'espace. Dans la phase temps réel de visualisation, il ne reste plus qu'à rechercher dans quelle zone de l'espace on se trouve afin d'accéder directement au graphe de priorités correspondant.

Sur ces idées je lançais la réalisation d'un système de visualisation que je baptisais GSI (ou Générateur pour la Synthèse d'Images).

Cette machine fut construite en 79 du point de vue du matériel par la société INFOREL et par l'antenne de RENNES de SUPELEC sur contrat du CELAR pour l'écriture du logiciel assembleur: dès cette époque elle permettait la représentation en surfaces ombrées d'un ensemble de 400 facettes en temps réel soit 10000 facettes par secondes (Les meilleures machines actuelles en 88 soit près de 10 ans plus tard génèrent au maximum 50000 facettes par seconde )

Cette machine était constituée:

- D'une mémoire d'image 512\*512\*8 bits
- D'une table de couleurs 8 bits vers 3\*8 bits RVB
- D'un générateur de facettes câblé pour le remplissage des facettes. Cette carte électronique remplissant automatiquement en mémoire d'image des polygones ombrés.
- D'une carte dotée d'un processeur AMD 2900 effectuait les opérations de pré-traitement des images (rotations, clipping, etc...)
- Du logiciel à base de priorités par régions de l'espace défini plus haut.

Ce matériel a été exposé en divers endroits (dont une présentation à PHOENIX, Arizona en 81). Il a été repris ensuite dans ses principes par la société SOGITEC qui en a extrapolé sa gamme de matériels GI 500 puis GI 1000. J'ai ensuite reçu de la Direction Générale pour l'Armement (DGA) des royalties pour cette invention.

Simultanément cette machine suscita un certain intérêt non seulement de la part des militaires pour la simulation de vol, mais également de la part de branches très diverses de l'industrie, et en particulier l'audiovisuel, la CAO, les arts Graphiques, et les Télécommunications.

#### **7-4/ MES RECHERCHES AU CCETT**

C'est ainsi que le CCETT me demanda en 81 de prendre la direction d'un projet de création d'une nouvelle machine de synthèse d'image cette fois-ci orientée audiovisuel 3D graphique.

J'ai donc entrepris en 1981 avec l'aide d'une équipe de recherche la réalisation d'un prototype de synthèse d'images polyvalent que j'ai dénommé CUBI 7 et qui fut industrialisé plus tard par la société TELMAT.

Les originalités introduites dans cette machine sont nombreuses et certaines, malgré l'évolution rapide des techniques, encore sans équivalent dans le monde. Ces concepts sont les suivants:

- Mode de désignation directe d'écran (décrit plus haut )
- Choix du Z-BUFFER (ce qui était à l'époque la première machine de ce type)

*NOTA: La GI 10000 de SOGITEC dont j'ai conçu les principes avant de quitter le CELAR est sortie plus tard et comporte aussi un Z-BUFFER.*

*-Possibilités de mixage d'images dites multimodes, c'est à dire que la machine peut fonctionner selon des modes de représentation des images différents (typiquement: Z-BUFFER, RAY-TRACING et IMAGE RASTER issue d'une source extérieure donnant en plus l'information Z ) et mixer les différentes images produites avec élimination des parties cachées. (aucun autre matériel du commerce ne dispose actuellement de cette possibilité).*

*-Recours à l'APL puis au langage NIAL (ou LISP-APL) pour :*

La modélisation des objets.

La définition des animations avec recours aux techniques des langages orientés objets.

**Possibilités non implémentée à l'heure actuelle sur cette machine mais potentiellement extensibles:**

*-Sculpture sur écran. (Voir chapitre 3)*

*-Proposition d'architecture de machine à base de TRANSPUTERS (chapitre 4) .*

Conjointement à ces recherches j'ai conçu toujours à partir de l'APL et de son langage dérivé le NIAL, des algorithmes venant en support des idées décrites ci-dessus:

-COMPILATEUR APL permettant de transformer toute phrase APL en un code C ou FORTRAN. Ce compilateur repose sur un automate à états finis; Il peut se compiler lui-même (fonctionnement en compilateur de compilateur), permettant ainsi de le porter sur toute autre machine et de s'exécuter lui-même en C.

Le but d'un tel logiciel est le suivant:

La productivité d'un programmeur moyen se situe, selon les normes IBM aux alentours de 10 lignes de code écrites, documentées et testées par jour. La conclusion immédiate que l'on peut tirer de ce chiffre est que l'efficacité des lignes écrites par le programmeur doit sa performance à la complexité des actions qu'elle engendre. Il importe donc que cette complexité soit maximale.

Prenons un exemple:

Si l'on écrit la ligne de code suivante:

$$A = B + C \quad (1)$$

en basic, en FORTRAN, en C et en APL:

En basic (en FORTRAN ou en C) A, B ou C ne peuvent représenter que des nombres.

En APL, au contraire, A, B ou C peuvent représenter non seulement des nombres, mais également des matrices ou des tableaux généralisés (matrices de matrices).

Le temps nécessaire à l'écriture de la formule (1) étant le même dans les deux langages, l'on constate que l'efficacité de (1) en APL sera supérieure à nombre de ligne égal, puisque l'opération effectuée est dans un cas une simple addition de deux nombres, et dans l'autre une addition matricielle.

Cependant APL est un langage interprété et non compilé, c'est à dire qu'il interprète et exécute directement les instructions du programme. Nous pouvons faire ici deux remarques:

- a/ *Plus la complexité du code déroulé est grande par rapport à la complexité de l'instruction source, plus la différence entre langage compilé et interprété tend à s'estomper. En effet, le temps d'interprétation de:  $A = B + C$  devient négligeable devant son temps d'exécution dans le cas où A, B et C sont des matrices.*
- b/ *De plus, les fonctions encapsulées dans le langage APL sont très optimisées. Cependant, il est clair que dans les cas où la phrase APL fait intervenir des fonctions élémentaires avec des opérandes de type scalaire, l'on se rapproche des pénalisations des langages interprétés de type BASIC.*

Or ce cas se rencontre fréquemment dans un programme, où il n'est évidemment pas question de se passer de ces grandeurs. De plus, malgré l'élégance de la programmation APL, le recours à des boucles se produit parfois, pénalisant aussi l'efficacité du langage.

Or s'il est clair que tout programme APL peut se réécrire en C, il n'en demeure pas moins que le taux d'expansion moyen est de 1 à 10 entre un programme APL et un programme C, ce qui pénalise lourdement, d'après les considérations précédentes les temps de développement au profit d'APL.

Il m'est apparu en 1980 qu'une solution permettant de concilier les deux, à savoir:  
:développement rapide  
:code exécutable rapide  
était possible en COMPILANT le code APL:

J'ai appris récemment qu'une société française avait réalisé depuis peu un compilateur APL, gagnant des facteurs d'exécution de 10 au moins par rapport à une version interprétée.

Les résultats de mesures montraient que les temps d'exécution obtenus avoisinaient ceux réalisés en codant directement en C, donc avec un temps d'écriture du programme 10 fois plus long environ.

## **LA REALISATION DU COMPILATEUR APL:**

Le compilateur analyse chaque phrase APL de la même façon que l'interpréteur APL lui-même, c'est à dire de la droite vers la gauche.

Il met en pile les parenthèses pour ne s'intéresser qu'aux instructions les plus internes qu'il remplace par des appels aux fonctions de base d'APL, qu'il génère explicitement dans son code objet.



Les crochets des matrices sont également traités; Il accepte toutes les combinaisons d'opérateurs autorisées par APL.

Il déclare les constantes au début de chaque sous-programme et initialise les variables à utiliser dans chaque sous-programme, ainsi que les variables globales en début de programme.

Ce programme est entièrement réalisé et testé. Il génère du code C assimilable par tout compilateur de ce langage. Il reste cependant à écrire les différentes fonctions élémentaires de l'APL en C. Cet ultime travail étant plus fastidieux que délicat à mener à bien.

#### *CONCLUSION sur la compilation APL:*

Outre le gain de temps que cette méthode procure, elle ouvre une voie nouvelle à l'un des problèmes majeurs de l'informatique actuelle, à savoir la programmation parallèle:

En effet, l'esprit de la programmation en APL consiste à travailler sans boucle et surtout le plus possible avec des grandeurs complexes telles que des tableaux, ou des arborescences.

Il va de soi que contrairement aux langages de bas niveau tels que le C ou le Pascal, la programmation en APL permet d'écrire directement des programmes parallèles. Autrement dit, la sémantique du parallélisme que connaît seul le programmeur est conservée dans son programme APL. Reprenons à cet effet une variante de l'exemple ci-dessus:

$$A = (B + C) * (D + E) \quad (2)$$

Dans le cas où A, B, C, D et E sont des matrices.

Le programme en APL aura exactement la forme de (2)

En C ou en Pascal, le programmeur devra soit écrire:

```
for(i=0; i<N; i++)
{
    for(j=0; j<P; j++)
    {
        A[j] = (B[i][j]+C[i][j])*(D[i][j]+E[i][j])
    }
}
```

L'on voit difficilement comment un automate tendant à détecter des processus parallélisables pourrait déduire de cette écriture la formule (2).

Tout au plus pourrait-il optimiser la boucle interne en faisant calculer séparément  $B[i][j] + C[i][j]$  sur un processeur et  $D[i][j] + E[i][j]$  sur un autre.

L'inconvénient majeur réside dans le fait que chaque processeur aurait une charge très courte de calcul par rapport aux tâches de gestion du parallélisme. (comme c'est d'ailleurs le cas dans la plupart des compilateurs vectoriels qui ne peuvent remonter à la structure sémantique du programmeur et ne parallélisent que de courtes expressions de très bas niveau).

Il est connu, en particulier, que les performances vectorielles d'un ordinateur s'effondrent lorsque le programmeur n'utilise pas spécialement le mode vectoriel, alors que ce mode est déjà intrinsèque dans APL.

L'on peut objecter que le programmeur pourrait aussi définir à l'aide de sous-programmes l'addition matricielle ainsi que la multiplication matricielle. Ceci suppose en fait qu'il aurait à réécrire la majeure partie des fonctions de base d'APL (qui seraient donc des fonctions évoluées en C).

Il semble plus logique de partir directement des fonctions mathématiques d'APL, qui forment un ensemble structuré et cohérent.

La méthode que je préconise, à partir d'un compilateur APL permettrait de résoudre de façon élégante le problème de la programmation parallèle de la façon suivante:

Le programmeur utiliserait les primitives APL, introduisant du même coup la sémantique du parallélisme.

A ce stade, un programme APL très simple pourrait répartir la charge de calcul entre plusieurs processeurs parallèles.

L'intérêt majeur de ce système est que la charge en question serait beaucoup plus importante pour chaque processeur (étant donné la structure matricielle des instructions de base d'APL) que les temps consacrés aux échanges de données et au séquençement. Ils n'induiraient donc pas de surcoût de complexité.

#### **-Algorithme de routage pour la CAO écrit en APL:**

Cet algorithme recherche le chemin optimum pour relier, compte-tenu d'obstacles donnés, 2 points d'un circuit imprimé.

Il repose sur la méthode du minimum de distance entre un point à relier et un autre. Il teste différents trajets possibles, et s'appelle de façon récursive.

#### **-MACRO-GENERATEUR DE MICROCODES pour MICROPROCESSEUR en TRANCHES écrit en APL**

Ce langage permet de générer du code machine à partir d'instructions en langage évolué interprétées en APL. A partir d'un jeu d'instructions dont les champs ont été définis au préalable par le concepteur du système, il est possible de rentrer chacune de ces instructions à l'aide de champs de bits dont le contenu peut être soit une constante, soit un paramètre de la fonction, soit une fonction APL.

Enfin, à l'époque actuelle, j'oriente mes recherches vers l'analyse et la reconstruction par synthèse de scènes naturelles pour lesquelles des résultats encourageants sont en cours d'élaboration; Le but fixé consiste à pouvoir extraire des scènes naturelles, quelle que soit leur complexité, leur base de donnée 3D, en vue d'un codage à très fort taux de compression:

Pour donner un ordre de grandeur, les algorithmes de codage-compression classiques utilisés actuellement atteignent des taux de compression de l'ordre de 50 en images fixes et de 20 en images animées; Le but visé consiste à tenter d'améliorer ces taux tout en conservant une qualité d'image et une résolution d'écran satisfaisantes.

De plus, les objets extraits de la scène d'origine (dont on a déduit la base de données ) pourraient être transmis séparément avec uniquement leurs paramètres

directeurs de mouvement durant une séquence d'animation; Ce qui permettrait d'améliorer les taux de compression puisque seuls les mouvements codés seraient transmis durant la phase dynamique et non les bases de données géométriques.

**ANNEXE 7: FORMULATION MATHÉMATIQUE DE L'ÉQUATION ÉCONOMIQUE DES MACHINES GRAPHIQUES**

L'on peut écrire empiriquement que si C représente le coût d'un matériel, T le temps de calcul d'une image et A un paramètre lié à la complexité algorithmique utilisée :

$$(C * T)/A = Cte$$

Cette formule repose sur une observation de l'offre industrielle en synthèse d'image: Elle signifie que si T est faible ( de l'ordre de 40 mS pour le temps réel) c'est le coût du matériel qui augmente si l'on souhaite améliorer la qualité algorithmique de l'image produite. C'est le cas des simulateurs de vol, où l'image doit être générée en temps réel: l'on sait que, par exemple, le seul fait de demander de la texture sur un simulateur fait doubler le temps de calcul (ou le volume d'électronique à temps de calcul constant).

A l'inverse, lorsque le temps réel n'est pas indispensable, C peut rester constant, et la qualité algorithmique augmenter: L'on atteint dans le cas d'images de très haute qualité des temps de calcul de l'ordre de plusieurs jours pour une seule image. L'on est très loin, dans ce cas, du temps réel. D'ou la formule liant les variables ci-dessus.

## **ANNEXE 8: LES TECHNIQUES NEURO-MIMÉTIQUES POUR LA SAISIE et la SYNTHÈSE DES FORMES 3D et des MOUVEMENTS**

Cette thèse ne serait pas complète si elle omettait d'envisager ce que sera demain le quotidien de la saisie des objets, des mouvements, leur synthèse et quelles architectures de machines en assureront la bonne marche, compte-tenu des considérables progrès des techniques neuronales: Cette annexe ne peut bien entendu s'appuyer sur des réalisations complètes, mais sur quelques recherches de pointe menées actuellement dans quelques laboratoires de par le monde.

Au cours de ma carrière vécue dans le domaine de la synthèse des images, j'ai de nombreuses fois entendu le type de question suivante émise par des visiteurs:

Serait-il possible, à partir d'une image de caméra, voire d'un couple d'images stéréoscopiques, de saisir en 3D une scène (voire des mouvements) afin de les synthétiser: Cette forme de saisie aurait un avantage évident: la simplicité, adjointe à un logiciel classique de modélisation, qui jouerait le rôle d'éditeur d'objets.

Cette question, maintes fois répétée, reste encore dans une large mesure sans réponse matérielle même au niveau de la recherche: Il faut toutefois souligner les études menées par A. GAGALOWITCH à l'INRIA, et plus généralement les techniques de "SHAPE FROM SHADING", c'est à dire l'extraction de la FORME à partir de l'ombrage: En effet, la synthèse d'image consiste fondamentalement à trouver un ombrage à chaque pixel de l'écran (information ayant perdu tout contenu sémantique, sauf pour l'oeil, qui fait naturellement l'opération de saisie ou d'extraction de l'information sémantique) à partir d'une base de données sémantiquement pertinente.

L'ensemble oeil-cerveau, en effet, réalise en permanence cette fonction, et génère au niveau cervical, l'information sémantique structurée de la scène observée: Autrement dit, l'oeil est en fait le plus puissant modèleur de saisie et de mouvements d'objets 3D, bien supérieur à toute machine existante. Malheureusement, mis à part quelques peintres et graphistes de grand talent, il est très difficile de faire l'opération de synthèse par l'intermédiaire de nos seules mains. Quand à retrouver, dans le cerveau, la base de données 3D relève actuellement de la plus pure utopie.

Cependant, l'introduction récente des réseaux neuronaux, ou neuromimétiques permet d'entrevoir des solutions à moyen ou long terme: Un réseau Neuronal est en effet un ensemble de cellules logicielles ou matérielles interconnectées, dont l'algorithme de calcul est une tentative de simulation des cellules neuronales du cerveau.

Ces réseaux ont 2 phases de fonctionnement distinctes:

Une phase d'apprentissage, durant laquelle on fournit d'une part un vecteur d'information (ou stimuli) en entrée, et dont on impose à la sortie un vecteur

correspondant à l'entrée: Par exemple, dans le cas de la reconnaissance de caractères, l'on fournit à l'entrée du réseau neuronal les valeurs d'ombrage de pixels du caractère observé, et en sortie, l'on donne au réseau la valeur octale de ce caractère.

Ensuite, durant la phase dite de reconnaissance, l'on fournit uniquement en entrée du réseau les valeurs d'ombrages de pixels, et celui-ci donne, à sa sortie, le code ASCII du caractère observé. Un tel système, étendu à des formes quelconques bi-dimensionnelles, a été réalisé sur un micro-ordinateur MAC-INTOSH ou ATARI. (LER 90)

Au contraire de l'approche de l'INRIA, ce système comporte un algorithme d'APPRENTISSAGE. Il est donc envisageable d'"APPRENDRE" à cet automate tout type de "PROTOTYPE" ou vecteur en entrée, ce vecteur pouvant être: Une forme 3D à reconnaître parmi un ensemble de formes, un mouvement, ou ce que nous pouvons appeler des invariants:

En effet, lorsque l'on apprend à un enfant l'objet "CRAYON", celui-ci peut le reconnaître dans une position quelconque: c'est à dire que le stockage de l'objet "CRAYON" dans le cerveau n'est pas -tant s'en faut- une simple mémoire d'image: Le code -ou plutôt la réponse comportementale du cerveau à l'objet "CRAYON"- est insensible aux translations et rotations 3D: Ce qui n'est pas vrai dans tous les cas: l'invariant est en fait étroitement lié au type d'objet stocké, et fait partie de ce stockage: Il est en effet très difficile de reconnaître un visage ou un caractère à l'envers, tout simplement parceque le cerveau n'est que très rarement entraîné à ce type de reconnaissance. Le cerveau ne stocke donc pas l'invariant correspondant à la rotation, mais seulement celui de la translation.

Dans un avenir relativement proche, (des réseaux neuronaux commencent déjà à le faire) il sera possible de reconnaître des bibliothèques d'objets 3D stockés sous forme de réponse comportementale dans des machines à neurones. Déjà, pour les besoins de compression d'images en vidéo communications, des approches 3D sont à l'étude, la base de donnée étant extraite à partir d'une bibliothèque de formes déjà apprises.

A ce stade, synthèse et saisie seront deux fonctions totalement complémentaires, et ne poseront plus aux infographistes les laborieux problèmes qu'ils rencontrent aujourd'hui. Les formes brutes seront saisies par ce moyen, les intentions des opérateurs captées par reconnaissance de parole, tant pour les couleurs que pour les formes ou le mouvement, après apprentissage. Cette approche diffère fondamentalement de l'Intelligence Artificielle, en ce sens que la machine apprend par l'EXEMPLE, et non pas à partir de règles, dont la saisie reste toujours très lourde, et les effets de bord mal contrôlés. L'architecture de synthèse des images travaillant avec une boucle de retour constituée du système d'analyse. (Voir figure 9: Machine d'analyse-synthèse neuronale).

*L'image finale étant constamment relue par l'analyseur et confrontée à l'ensemble des données stockées dans ses liens synaptiques à l'instar de l'ensemble oeil-cerveau-main-pinceau.*

Voilà, en quelques lignes, le futur très vraisemblable de la saisie des objets et de mouvements 3D.

Quant aux architectures, il est clair que les modèles cellulaires sont actuellement les plus prometteurs dans la recherche actuelle en infographie; Presque chaque mois voit apparaître un nouveau processeur de signal, cellule élémentaire communicante, raccordable à 4 voisins ou plus. Des réseaux Neuronaux numériques, analogiques, optiques, voire bio-moléculaires apparaissent également à l'heure actuelle, permettant de fabriquer des ensemble capables de réaliser les fonctions décrites plus haut. Cette orientation devrait conduire à une amélioration significative de la saisie-synthèse-codage-production des images.

## ANNEXE 9: UTILISATION DES RESEAUX NEURO-MIMETIQUES POUR L'ANALYSE et LA SYNTHÈSE DES IMAGES

### 9-1/DEFINITION DES SYTEMES NEURO-MIMETIQUES

Depuis 1988 environ, l'étude des systèmes neuro-mimétiques prend un essor tout à fait considérable, après quelques essais dans les années 60, et un oubli de quelque 30 années, consécutif aux déceptions engendrées par le PERCEPTRON, à une époque où la technologie du silicium était quasiment embryonnaire.

#### DEFINITION

Les techniques Neuro-Mimétiques peuvent se définir comme une tentative de modélisation des structures neuronales du cerveau, d'où leur nom de systèmes Neuro-Mimétiques.

*Un système peut être qualifié de neuro-mimétique lorsque:*

- a/ Il comporte un parallélisme massif de cellules élémentaires effectuant toutes la même opération. Ce parallélisme est aussi appelé "connectionniste".*
- b/ Sa programmation se fait exclusivement par l'exemple, au cours de phases d'apprentissage.*

Elles diffèrent fondamentalement des architectures de machines dites parallèles, en ce sens que chaque entité élémentaire - ou "NeuroMime"- ou "NEURONE", n'effectue qu'un seul type d'opération, cadencée ou non par des horloges extérieures. Cette opération est la suivante: Soit un neurone  $i$ , connecté au neurone  $j$  de valeur  $U_j$  par un coefficient pondérateur  $a_{ij}$ : l'équation d'état du neurone s'écrit:

$$U_i = F(\sum_j a_{ij} \cdot u_j) \quad (1)$$

$F$  étant une fonction de seuillage de type sigmoïde. Cette fonction constitue une simulation simple de la fonction neuronale. Elle est réalisée durant la phase dite de "*reconnaissance*". Le calcul des coefficients synaptiques  $a_{ij}$  s'effectue durant la phase dite d'*apprentissage*. Ces phases sont définies de la façon suivante:

- **Phase 1: Apprentissage:** L'on envoie au système Neuronal un vecteur d'entrée, et l'on force un vecteur à sa sortie: Ce vecteur constitue le modèle de réponse que l'on souhaite entrer dans le réseau pour un prototype d'entrée donné: Ex: Si l'on veut apprendre la reconnaissance de caractères à un réseau neuronal, le vecteur d'entrée sera par exemple constitué par la matrice de points du caractère à apprendre, et le vecteur de sortie sera le code ASCII de ce même caractère.



- **Phase 2: Reconnaissance:** Après la phase d'apprentissage, le système Neuronal doit être capable de redonner le vecteur de sortie appris pour le vecteur d'entrée correspondant même si le vecteur d'entrée est légèrement bruité ou différent du vecteur présenté durant la phase d'apprentissage.

Durant la phase 2, l'opération effectuée par chaque neurone est la somme pondérée de chacune des valeurs contenues dans les axones des neurones qui lui sont directement connectés. Voir Formule (1).

En général l'on applique une fonction de type sigmoïde à  $U_i$ , afin que les neurones ne se déclenchent que pour un seuil donné en fonction de la somme pondérés des entrées.

Ces techniques diffèrent fondamentalement des techniques d'architecture parallèle, en ce sens que chaque neurone effectue une fonction (1) bien définie, au contraire des calculateurs parallèles dont chaque noeud est une petite unité centrale de type microprocesseur pouvant effectuer tout type de calcul au même titre qu'un calculateur classique monotâche.

La formule (1) est généralement effectuée par l'ensemble des systèmes neuronaux, en revanche, le processus de la phase (1) (apprentissage) est loin d'être universel: De très nombreux travaux font état d'études sur des systèmes d'apprentissage de natures très diverses.

### LIEN AVEC LES ARCHITECTURES DE MACHINES DE SYNTHÈSE D'IMAGE

Les techniques neuronales relèvent de la théorie des automates, définie en particulier dans le modèle des machines de Boltzmann, qui diffèrent dans leur principe des machines de Van-Neumann, à la base des calculateurs que nous connaissons actuellement.

L'intérêt fondamental de ces machines est qu'elles rendent caduques la notion même de programmation: Un apprentissage par l'exemple est seul nécessaire pour assurer leur fonctionnement: Il est donc tout à fait concevable d'apprendre à de telles machines les techniques de synthèse d'images, essentiellement:

- Les rendus de surfaces: qui pourraient être saisies à partir d'objets réels que l'on aurait préalablement présenté devant une caméra connectée à l'architecture neuronale: ex du bois, du fer, du plastique, etc...
- Les transformations géométriques: Une machine neuronale peut tout aussi bien "apprendre" les techniques géométriques fondamentales utilisées en synthèse:
  - Rotations
  - Clipping
  - Transformation perspective
  - Remplissage des facettes

Sans donner explicitement l'algorithme mathématique correspondant, mais un certain nombre d'exemples dans lesquels il est appliqué.

L'opération inverse, c'est à dire la saisie des scènes réelles, (partie 2 de cette thèse) serait alors effectuée de façon automatique.

Prenons un exemple: Lorsque l'on enseigne à un enfant une table de multiplication, celui-ci ne calcule pas *effectivement* les opérations, mais stocke dans ses structures neuronales l'association entre les vecteurs d'entrée présentés (ex: 2 + 3) et le résultat: 5.

De même, lorsqu'une personne se déplace dans un couloir, elle ne suit pas d'algorithme défini d'optimisation de trajectoire, mais applique le résultat comportemental de ses structures neuronales, entraînées à partir d'autres exemples de marche appris dans son enfance. Des travaux tout récents aux Etats-Unis ont montré que des machines peuvent être entraînées à effectuer l'opération : "Tracer un cercle" SANS rentrer explicitement l'équation:

$$x^2 + y^2 = R^2$$

Lorsqu'un homme trace un cercle, il n'applique pas la formule précédente.

De même, l'on peut entraîner un réseau neuronal par exemple à piloter un camion semi-remorque, sans explicitement donner l'algorithme de pilotage, mais en l'entraînant dans un nombre limité de conditions, comme d'ailleurs pour un pilote humain, qui utilise en permanence la réponse comportementale de son cerveau entraîné dans des circonstances antérieures, et recevant les informations de la situation présente.

La structure neuronale possède alors un avantage considérable: l'association comportementale est toujours réversible, en ce sens qu'elle peut reconstituer l'entrée d'un vecteur, générant une sortie donnée. (ce que ne peut faire une machine purement algorithmique qui part de données et obtient un résultat: A moins que l'on puisse trouver la formule réciproque, ( ce qui est très rare) la machine algorithmique ne peut effectuer d'opérations réciproques.

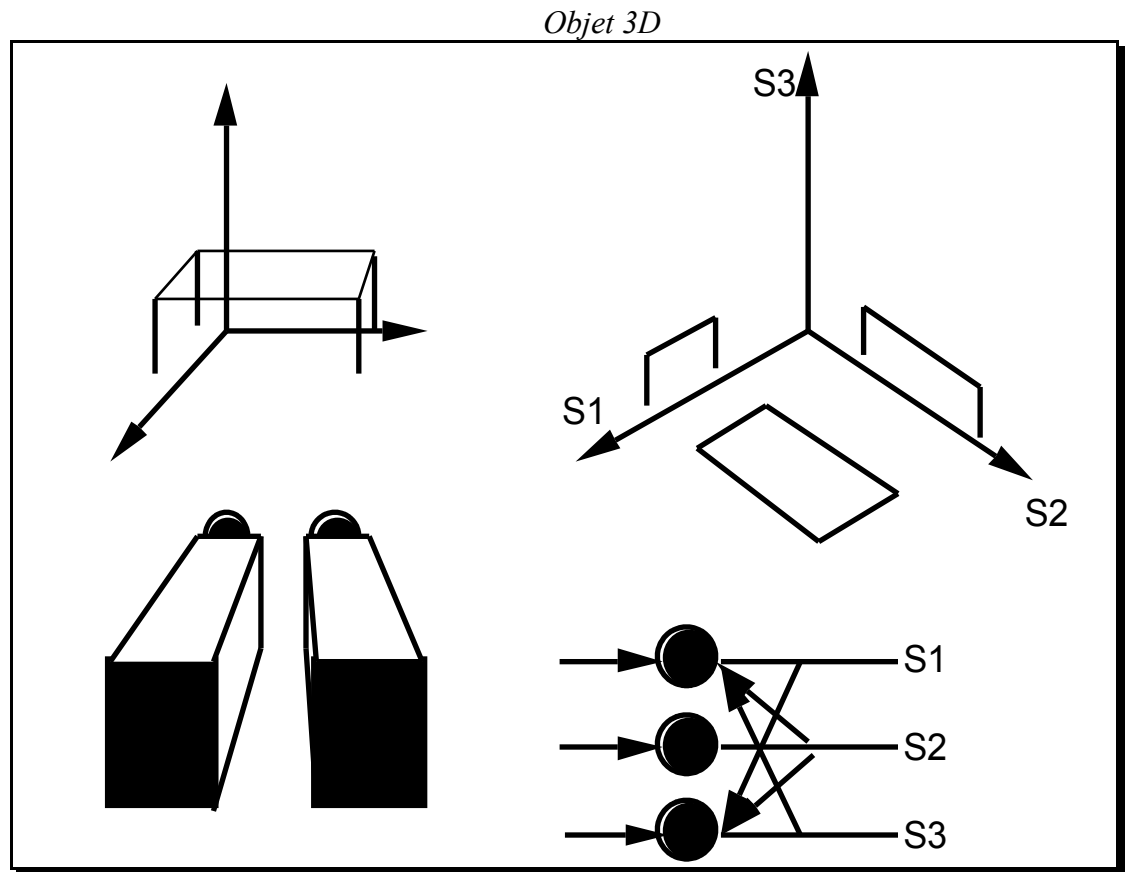
Par contre, l'opération d'analyse 3D permanente que fait le couple: oeil-cerveau pourra être traité par une machine neuronale, APRES apprentissage des formes qu'elle est susceptible de rencontrer, ainsi que des lois géométriques et de rendu de surface des solides 3D.

Il est ainsi saisissant de constater combien sont proches, vu sous l'angle prometteur mais complexe du neuronal, le système de saisie de l'information 3D, et le système de représentation, ces fonctions étant dans le cerveau humain étroitement imbriquées.

**9-2/EXTRAIT DU TEXTE DE L'EXPOSÉ PRÉSENTÉ PAR J. HERAULT (HER 89) aux Journées d'électronique 89 de l'Ecole Polytechnique Fédérale de LAUSANNE:**

Dans cet exposé, J. HERAULT esquisse une possibilité d'application des réseaux neuronaux à l'analyse d'objets 3D, et à leur classification en sous-ensembles, après un rappel des possibilités d'utilisation des réseaux neuronaux pour le traitement d'image classique en filtrage, en particulier pour effectuer des opérations de lissage, de détection d'orientations et de coutours, ou d'augmentation de contraste.

Cet exposé préfigure les nouvelles possibilités de saisie 3D des systèmes neuro-mimétiques.



*Fig 16: Modélisation Neuronale d'objets 3D*

## ANNEXE 10: VISION PROSPECTIVE DE LA SAISIE et de la SYNTHÈSE D'IMAGE EN L'AN 2000

Il est probable que les années 90 verront se confirmer les tendances suivantes dans le domaine de la synthèse des images:

- *Disparition progressive des super-calculateurs de grande puissance et d'usage général pour la synthèse d'image* ( Ces calculateurs de type CRAY ou CONVEX ne sont pas dotés de composants matériels spécialisés 3D ).
- *Développement des stations de travail.*
- *Emergence de stations intégrant de plus en plus de fonctions graphiques 3D câblées sur silicium.*
- *Réduction des coûts à performances égales:* Cf. le développement des stations de travail personnelles telles que MAC-INTOSH, ATARI ou AMIGA en graphique 3D).
- Modélisation à partir de petites stations de travail à bas coût tels que MAC-INTOSH, PC ou autres.
- Emergence d'accélérateurs (ou architectures spécialisées de rendu 3D) connectés sur réseau à des grappes de stations personnelles type MAC ou PC.
- Emergence de protocoles d'interfaces entre modeleurs et architectures de rendu, permettant une interconnection aisée de l'ensemble dans le cadre de réseaux locaux ou publics liant:
  - Des stations de modélisation bas coût.
  - Des ACCELERATEURS de calcul d'image (rendu) de très grande puissance
  - Des systèmes de stockage des images avec compression des données.
  - Des protocoles d'échanges de bases de données graphiques 3D entre stations de travail et accélérateurs de rendu.  
(RENDERMAN de Pixar-NEXT en est la préfiguration.)
- Banalisation de la logique câblée pour les effets de rendu les plus demandés: Contours (aliasing) et surfaces lisses, textures, ombres portées, possibilité d'utilisation économique du lancer de rayons et des techniques de radiosité.
- Amélioration considérable des systèmes de saisie, par acquisition directe des objets et des mouvements 3D par de simples caméras associées à des réseaux neuronaux.
- an 2000: FIN DE L'ERE DE LA PROGRAMMATION: les machines seront alors entièrement programmées par l'EXEMPLE qu'il soit vocal, visuel ou gestuel (par saisie d. La notion de fichier et de programme disparaîtra alors entièrement, grâce à l'émergence des **réseaux neuronaux**: Prémices de cette évolution, les

réalisations dès 89 de simulateurs neuronaux aptes à reconnaître en une fraction de seconde une forme quelconque, actuellement en 2D sur une station personnelle. (LER 89).

- Application des réseaux neuronaux à la saisie directe de formes et de mouvements 3D. Ces réseaux prendront la forme actuellement insoupçonnée de chaînes biomoléculaires, qui seront "cultivées" sur des substrats quartz par exemple, à l'instar peut-être de ce qui se fait déjà au Japon. Ces réseaux pourront intégrer:

- *Des rétines biologiques*, disposant d'une résolution comparable à celle de l'œil humain

- *Des réseaux neuronaux d'analyse d'image* effectuant les traitements bas niveaux de type oculaires: détection de contours, de mouvement, du relief, des textures;

- *Des réseaux effectuant les tâches de vision cervicale haut niveau*: Mises en correspondance des résultats des étages bas niveau, et des réponses comportementales apprises.

- La réponse perceptible de ces systèmes se fera sous forme de synthèse des résultats permettant la commande d'actions (ex: en robotique, pilotage d'engins, etc...), et éventuellement dessin de la scène vue (cas de la Télévision et du codage). Ce re-dessin de contrôle permettant également de mesurer le degré de compréhension de la "machine". A ce stade, analyse et synthèse d'image se feront de façon quasiment indifférenciée, à l'exemple du couple œil-cerveau.

*NOTA: Beaucoup de personnes ne se rendent pas compte qu'elles effectuent en permanence grâce au couple œil-cerveau l'analyse de la scène qu'elles observent: témoin cette question, maintes fois posée par des visiteurs à l'issue de la visite de mon laboratoire de synthèse d'image: "Pourquoi n'est-il pas possible de saisir les objets d'une scène à l'aide d'une caméra afin de les rentrer dans un système de synthèse d'image, et de les restituer, après traitement (rotation, changement de couleur, de texture, par exemple), sur l'écran?" Ce type de question montre à l'évidence la sous-évaluation de la complexité de la tâche d'analyse d'image permanente effectuée par l'homme. A l'exemple du graphiste, il est clair que le système neuronal pourra à l'avenir non seulement effectuer cette analyse, mais également restituer une image synthétique sur écran TV. Cette image pouvant constituer une précieuse mesure du degré de compréhension du système pour une scène donnée.*

**ANNEXE 11:NOTES CONTRIBUTIVES sur le NEURONAL**

---

*Opus citatum extraits de:*

**LA RECHERCHE SUR LES RESEAUX NEURONAUX AU JAPON  
ETUDE CNRS**

*-Octobre 89-*

Le Professeur AMARI, de l'Université de TOKYO, oriente le projet New Information processing Technology du MITI vers la recherche théorique et vers la **biologie**.

La NHK effectue des recherches sur le système visuel, et en particulier sur les liaisons corticothalamiques, le rôle de l'aire 19, la reconnaissance des formes au niveau du cortex et le traitement du mouvement par les aires MT et MST. Ces études sont effectuées à l'aide d'images de synthèse couleur de 400 \* 600 pixels à 60 Hertz et 120 Hertz pour la vision en relief.

NTT Basic Research Laboratories à Musashino:

*Effectue des **cultures de neurones à croissance contrôlée microgravés sur substrat quartz**. Les neurones utilisés pour cette culture sont issus de la dissection de ganglions dorsaux d'embryons de poulets ou de souris adultes. L'activité de ces réseaux est mesurée par la concentration en calcium à l'aide de batteries de 100 photodiodes.*

NTT a également conçu un préprocesseur de type cochlée.

NTT produit la machine ELIS (de 1 à 3 millions de yens 16 à 128 MOctets). La carte mère est disponible sous forme de carte d'extension sur MAC II d'APPLE.

**LES GRANDS PROGRAMMES NEURONAUX JAPONAIS:**

- Le programme FRONTIER RESEARCH ON TELECOMMUNICATIONS du Ministère des Postes et des Télécommunications fait une part pour les problèmes de *compression d'image TV*.
  - Le programme HUMAN FRONTIER consacre un volet à la neurobiologie
  - Le programme ERATO de l'AST soutient le projet BIOHOLONICS
  - Le programme New Information Processing Technology du MITI: 2 milliards de yens sur 8 à 10 ans.
-

## ANNEXE 12: ANALYSE STATISTIQUE D'IMAGE PAR FONCTIONS PROBABILISTES ITERATIVES FRACTALES

Notre propos n'est pas ici d'introduire la théorie du chaos et des fractales, mais d'avancer deux hypothèses:

-a/ Il existe un lien entre Neurones et courbes ou surfaces Fractales

-b/ Les fonctions probabilistes itératives pourraient permettre de modéliser des objets 3D du monde environnant.

### 12-1/ RAPPEL SUR LA THEORIE DU CHAOS ET DES FRACTALES-LIEN AVEC LES RESEAUX NEURONAUX

Depuis l'année 1950, les scientifiques ont intensifié les recherches sur un champ de phénomènes qui dans la décade suivante fut recouvert sous le vocable de théorie du chaos ou théorie des catastrophes.

Cette théorie a depuis permis de modéliser de nombreux phénomènes physiques, en particulier dans le domaine de l'optique, de la géologie, et également pour modéliser des comportements issus de groupes humains tels que les fluctuations boursières. ( Per 90-Gud 90)

En parallèle, les études sur les fractales menées par Benoît Mandelbrot chez IBM et par Carpenter ont permis de générer des êtres mathématiques nouveaux, dont la dimension, d'après leurs auteurs, est fractionnaire: En particulier, un fractal est un être mathématique intermédiaire entre la courbe et la surface: Sa dimension est donc comprise entre celle de la ligne (dimension 1) et celle de la surface (dimension 2).

Perez, chercheur chez IBM à Montpellier a défini la notion de neurone fractal (PER 90). Cette structure neuronale a été présentée pour la première fois au congrès international sur les Réseaux Neuronaux en juin 90 à San Diego (USA).

### 12-2/ LES FONCTIONS ITERATIVES

Ces fonctions ont été définies par Barnsley, professeur à l'Université d'Atlanta, à partir d'un ensemble de transformations affines  $w_i$  dotées d'un taux de probabilités  $p_i$  de la façon suivante:

$$\begin{aligned}x' &= a*x + b*y + c \\y' &= a'*x + b'*y + c'\end{aligned}$$

Un ensemble IFS ou Iterative Function System est un ensemble  $\mathbf{W}$  de fonctions  $w_i$ :

$$\mathbf{W} = \{w_1, w_2, \dots, w_i, \dots, w_n\}$$

chaque matrice de  $w_i$  ayant des valeurs propres inférieures à 1;

$\mathbf{W}$  est associé à un ensemble de probabilités:

$$\mathbf{P} = \{p_1, \dots, p_n\}$$

**Théorème du collage ou de la contraction:**



Soit  $(W,P)$  un IFS, un attracteur  $A$  sera défini de la façon suivante:

$$A = \bigcap_{n=0}^{\infty} U_{i=1}^n w_i(L)$$

Soit un ensemble de contractions  $w_i: K \rightarrow K, i = (1,2,\dots,n)$  tel que  
 $h(L, U_{i=1}^n w_i(L)) < \epsilon$  pour  $\epsilon > 0$   
 alors  $h(L,A) < \epsilon/(1-s)$   
 où:  
 $A$  est l'attracteur de l'IFS  
 $h$  est la distance de Hausdorff:  
 $h(B,C) = \max[\max_{x \in B} \min_{y \in C} (d(x,y)), \max_{y \in C} \min_{x \in B} (d(x,y))]$   
 $0 < s < 1/d(w_i(x), w_i(y)) \leq s(d(x,y)) \quad \forall x,y \in K$

**Il s'ensuit que pour trouver un ensemble adéquat de fonctions de recouvrement pour reconstruire  $A$ , nous avons seulement besoin de faire un collage d'éléments formés de copies contractées de l'ensemble  $A$  lui-même.**

L'expérimentation de ces fonctions a conduit à la génération de formes fractales approximant une forme donnée: ex: formes de plantes (Fougères)

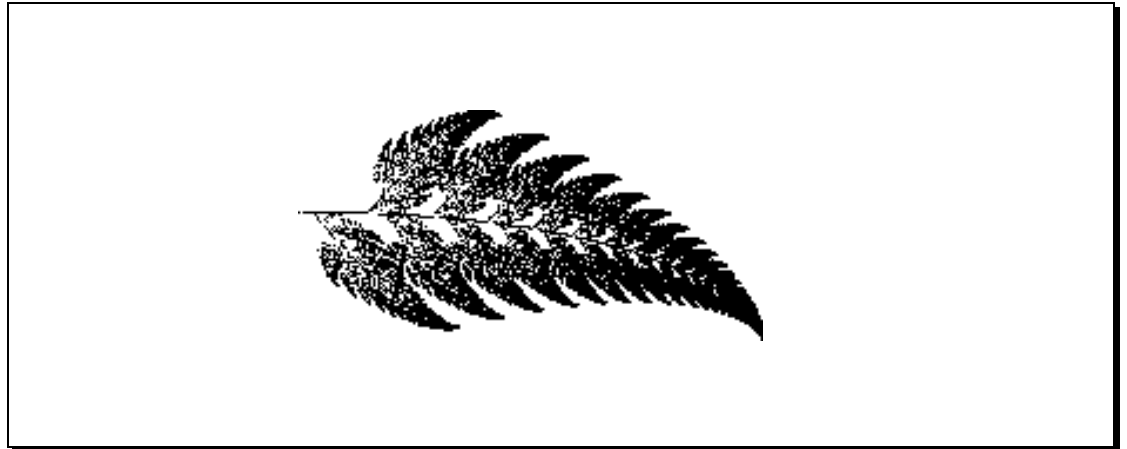
Le professeur Barnsley et son équipe affirment pouvoir approximer toute forme d'objets sur une image donnée.

L'objectif visé est la compression d'images TV, avec des taux très supérieurs à ce que l'on obtient à l'heure actuelle par les moyens classiques. L'on entend par compression d'image tout algorithme permettant de réduire la taille de stockage de cette image: Corrélativement, un algorithme de décompression recrée l'image originale, avec apparition éventuelle d'un certain nombre d'artefacts, c'est à dire de différences par rapport à l'image originale. L'appréciation de l'importance des défauts est essentiellement subjective, l'oeil humain seul pouvant juger de la qualité du résultat obtenu. Des images ont ainsi été compressées de façon très récente avec des taux tout à fait exceptionnels de plus de 1000; C'est à dire qu'une image comportant 24 bits par points peut être stockée à raison de 0.024 bits par point. Les meilleurs systèmes de compression actuels ne dépassant pas 0.5 bits par point.

*Nota: Il est toujours possible de demander un taux de compression supérieur, mais les dégradations deviennent alors visibles et rapidement inacceptables lors de tests subjectifs.*

**Conclusion (provisoire) sur les approches fractales:**

**Cette technique toute récente est expérimentée de par le monde par quelques chercheurs seulement; cependant, elle semble très prometteuse, et semble pouvoir s'associer à des techniques neuronales, en particulier pour la reconnaissance rapide des coefficients fractals en fonction des images et des objets à analyser. Notamment, les correspondances de points liées à la transformation  $w_i$  n'est pas sans rappeler les liens qui s'établissent dans un réseau neuronal entre un neurone  $i$  et son correspondant  $j$  avec le poids synaptique  $w_{ij}$ .**



*Fig 17: Exemple de fonction probabiliste itérative IFS:  
Modélisation d'une feuille de fougère (voir listing ci-dessous)*

<b>LISTING APL DE LA FONCTION IFS FRACTALE ITERATIVE et PROBABILISTE POUR LA FEUILLE DE FOUGERE</b>
---

```

[ 0] _FERN
[ 1] A1_2 2μ0 0 .16
[ 2] A2_2 2μ.2 @.26 .23 .22
[ 3] A3_2 2μ@.15 .28 .26 .24
[ 4] A4_2 2μ.85 .04 @.04 .85
[ 5] B1_2 1μ0 0
[ 6] B2_2 1μ0 1.6
[ 7] B3_2 1μ0 .44
[ 8] B4_2 1μ0 1.6
[ 9] P_4μ.01 .08 .15 .1
[ 10] _IFS

[ 0] _IFS
[ 1] XY_2 1μ0 0
[ 2] _TCFF
[ 3] LOOPÿ
[ 4] K_(?1000)÷1000.
[ 5] XY_K_W XY
[ 6] MOVETO OFFSET+_XY_20
[ 7] LINETO OFFSET+_XY_20
[ 8] ,LOOP

[ 0] Z_K_W XY
[ 1] ÿCALCUL DE LA FONCTION AFFINE ITERATIVEÿ
[ 2] ,(K_P[1])/W1
[ 3] ,(K_P[2])/W2
[ 4] ,(K_P[3])/W3
[ 5] ,W4
[ 6] W1ÿZ_B1+A1+_XY`,0
[ 7] W2ÿZ_B2+A2+_XY`,0
[ 8] W3ÿZ_B3+A3+_XY`,0
[ 9] W4ÿZ_B4+A4+_XY`,0

```

<b>LISTING DU LOGICIEL NEURONAL EN APL POUR L'ANALYSE ET LA RECONNAISSANCE DE FORMES</b>
--

```

™ACTION
[1] ADD FEATURE
[2] ADD CLASS
[3] LEARN\APPREND\Œ
[4] RECOGNIZE\RECONNAIT\Œ
[5]

™ADDFEATURE
[1] _TCFF
[2] FEATURE_FEATURE BOX '
[3] A__EX 'FEATURES'
[4] 8 SETMENU FEATURE
[5] DRAWMENUBAR

™ANIM
[1] PENPAT WHITEPAT
[2] _TCFF
[3] BOUÿ
[4] (R_PICTFRAME NANA1)DRAWPICTURE NANA1
[5] PAINTRECT R
[6] (R_PICTFRAME NANA2)DRAWPICTURE NANA2
[7] PAINTRECT R
[8] ,BOU

™APPREND
[1] 'ÿRESULÿ TABLEAU DES RESULTATS
[2] 'ÿRESUL DEPEND DU TEMPS ET DE L'INDEX S DE LA VALEUR A RECONNAITRE
[3] BOUÿ
[4] P_0 2µ0
[5] P_TRACE
[6] ,(0=(µP)[1])/0
[7]
[8] 'NUMERO DU PROTOTYPE'
[9] EPREUVE
[10]
[11] 'VALEUR DU SYMBOLE ENTRE?'
[12] S_S BOX '
[13] C_C,[1]NªEXT P
[14] EPREUVE_EPREUVE+1
[15] ,BOU

™APPRENDANG
[1] 'ÿRESULÿ TABLEAU DES RESULTATS
[2] 'ÿRESUL DEPEND DU TEMPS ET DE L'INDEX S DE LA VALEUR A RECONNAITRE
[3] EPREUVE_1
[4] S_(µA)µ0
[5] BOUÿ
[6] P_TRACE
[7]
[8]
[9] C[EPREUVE;]_(µA)ªEXTRACT P

```

```

[10]
[11] 'NUMERO DE L EPREUVE?'
[12] EPREUVE
[13]
[14] 'VALEUR DU SYMBOLE ENTRE?'
[15] S[EPREUVE]__
[16] EPREUVE_EPREUVE+1
[17] ,BOU

```

## ™Z\_A BOX B

```

[1] 'ÿAJOUTE B AU TABLEAU A
[2] 'ÿSI µB>µA ON REDIMENTIONNE A
[3] B_,B
[4] ,((µB)_ (µA)[2])/FIN
[5] A_((µA)[1],µB)aA
[6] FINÿB_(µA)[2]aB
[7] Z_A,[1]B
[8]
[9]

```

## ™Z\_CADRE P

```

[1] 'ÿCADRAGE DE P SUR 0 2 0 1
[2] Z1_@1 0_P
[3] Z2_1 0_P
[4] CURVE_(Z2-Z1)*2
[5] CURVE_CURVE[;1]+CURVE[;2]
[6] 'ÿEXTRAIT LES COORDONNEES RECADREES DE P
[7] MINY_@/,P[;1]
[8] MINX_@/,P[;2]
[9] MAXY_/_/,P[;1]
[10] MAXX_/_/,P[;2]
[11] P[;1]_2_(P[;1]-MINY)÷MAXY-MINY
[12] P[;2]_(P[;2]-MINX)÷MAXX-MINX
[13] Z_P
[14]

```

## ™CHANGEPAT NEWPAT

```

[1] PENPAT NEWPAT
[2] PGM_PGM,'PENPAT ',NEWPAT,_TCNL

```

## ™CONTINUE

```

[1] _CP_( _PH-2),0
[2] ' 'HIT THE MOUSE BUTTON TO CONTINUE '
[3] ' "
[4] _CP_( _PH-1),( _PW-1)
[5] ' "
[6] WAITMOUSEDOWN
[7] WAITMOUSEUP

```

## ™CROIX XY;DIM

```

[1] DIM_10
[2] DRAWLINE (XY-DIM,0),(XY+DIM,0)
[3] DRAWLINE (XY-0,DIM),(XY+0,DIM)

```

## ™DISPLAY

```

[1] _TCFF
[2] I_1
[3] BOUÿ
[4]
[5] ((10 5)µC[!;])_PUTBITS 0 10+(0,20_(I-1))

```

[6] ,(EPREUVE>I\_+1)/BOU

™EXENEU

[1] A\_N^A  
 [2] A1\_(μC)μA  
 [3] A\_+|A1-C  
 [4] MIN\_A'@/A  
 [5] ((10 5)μC[MIN;]) \_PUTBITS 0 100

™Z\_EXT P

[1] P\_@5\_CADRE P  
 [2] PENSIZ 1 1  
 [3] MOVETO P[1;]  
 [4] LINETO 1 0\_P  
 [5]  
 [6] Z,\_GETBITS 0 0 10 5

™Z\_EXTRACT A;Z2

[1] 'yEXTRAIT LES COORDONNEES SIGNIFICATIVES DE A  
 [2] 'yON CALCULE D'ABORD LES COORDONNEES DES VECTEURS DE A  
 [3] Z1\_@1 0\_A-(1 0\_A),[1]0 0  
 [4]  
 [5] 'yON NORMALISE CES VECTEURS/  
 [6] Z2\_NORME Z1  
 [7] 'yON CALCULE LES PRODUITS VECTORIELS DE CES VECTEURS 2 A 2y  
 [8] Z\_(V1\_(@1 0)\_Z2) VECT (V2\_1 0\_Z2)  
 [9] Z1\_@2+V1 PSCAL V2  
 [10] Z\_N^a(((Z1)\_2)/Z1

™Z\_EXTRACTNEW P

[1] Z1\_@1 0\_P  
 [2] Z2\_1 0\_P  
 [3] CURVE\_(Z2-Z1)\*2  
 [4] CURVE\_CURVE[;1]+CURVE[;2]  
 [5] 'yEXTRAIT LES COORDONNEES RECADREES DE P  
 [6] MINY\_@/,P[;1]  
 [7] MINX\_@/,P[;2]  
 [8] MAXY\_/\_/P[;1]  
 [9] MAXX\_/\_/P[;2]  
 [10] P[;1]\_(P[;1]-MINY)÷MAXY-MINY  
 [11] P[;2]\_(P[;2]-MINX)÷MAXX-MINX  
 [12] Z\_N^a,P  
 [13]  
 [14]

™Z\_EXTRACTV P

[1] 'yEXTRAIT LES COORDONNEES SIGNIFICATIVES DE A  
 [2] MINY\_@/P[;1]  
 [3] MINX\_@/P[;2]  
 [4] MAXY\_/\_/P[;1]  
 [5] MAXX\_/\_/P[;2]  
 [6] P[;1]\_(P[;1]-MINY)÷MAXY-MINY  
 [7] P[;2]\_(P[;2]-MINX)÷MAXX-MINX  
 [8] Z\_@P[;1]...P[;2]°. \*N

™FIX;DMY

[1] DMY\_\_FX PGM

™FLUSHKEY

[1] ,(0\_μ\_GETKEY)μ1

™HELP

[1] 'MODIFÿ MODIFIE UN PATTERN DE TYPE CARACTERE MANUSCRIT'  
 [2] 'ININEUÿ INITIALISE LE RESEAU NEURONAL'  
 [3] 'EXENEUÿ ITERE LE CALCUL DU RESEAU NEURONAL'  
 [4]

™ININEU

[1] 'ÿNÿNOMBRE DE NEURONES  
 [2] 'ÿAÿVALEURS DES NEURONES  
 [3] 'ÿCÿCOEFFICIENTS NEURONAU.  
 [4]  
 [5] N\_100  
 [6] EPREUVE\_1  
 [7] A\_Nμ0  
 [8] C\_(0,N)μ0  
 [9] RESUL\_10 10μ0  
 [10] PATTERNS\_10 10 50 2μ0  
 [11] S\_0 0μ"

™INIT

[1] MAKEHOTSPOTS  
 [2] CURR\_PAT\_BLACKPAT  
 [3] SELECTION\_XLINE  
 [4] PGM\_'MYPICTURE',\_TCNL,'\_TCFF ` PENPAT BLACKPAT',\_TCNL  
 [5] DRAWSCREEN  
 [6] \_CP\_( \_PH,\_PW)-1  
 [7] \_"

™MENU

[1] PENNORMAL  
 [2] 6 SETMENU \_CR 'TRACER'  
 [3] 7 SETMENU \_CR 'ACTION'  
 [4] 8 SETMENU FEATURES\_1 8μ'FEATURES'  
 [5] TRACER\_FIN\_2 6 1  
 [6]  
 [7] DRAWMENUBAR

™MODIF P

[1] 'ÿMODIFICATION D'UN PATTERN DE POINTS Pÿ  
 [2] \_TCFF  
 [3] EPS\_10  
 [4] MOVETO P[1;]  
 [5] LINETO P  
 [6] BOUÿ  
 [7] WAITMOUSEDOWN  
 [8] XY\_(μP)μXY0\_GETMOUSE  
 [9] PMIN\_P-EPS  
 [10] PMAX\_P+EPS  
 [11] T\_(XY<PMAX)^(XY>PMIN)  
 [12] T\_T[;1]^T[;2]  
 [13] T\_T`1  
 [14] ,(T>(μP)[1])/BOU  
 [15] PENMODE 10  
 [16] BOU1ÿ

```
[17] NEW_GETMOUSE
[18] ,(~BUTTON)/FIN
[19] ,(NEW^.=OLD)/BOU1
[20] MOVETO P[1;]
[21] LINETO P
[22] P[T;]_NEW
[23] MOVETO P[1;]
[24] LINETO P
[25]
[26] ,BOU1
[27] FINÿPENNORMAL
[28] MOVETO P[1;]
[29] LINETO P
[30] ,BOU
```

```
™Z_NORME A;N
[1] 'ÿNORMALISATION D'UN ENSEMBLE DE VECTEURS N,2
[2] N_((A[;1]*2)+A[;2]*2)*.5
[3] N_((µA)[1],1)µN
[4] N_N,N
[5] Z_A÷N
```

```
™PASTE
[1] SCREEN_GETCLIP
[2] (50+PICTFRAME SCREEN) DRAWPICTURE SCREEN
```

```
™PRINT V;C
[1] C__AV ' V
[2] PUTCLIP _AV[TABFONT[C]]
[3]
```

```
™PRINTFON1
[1] SCREEN_0µ"
[2] _WSLIST 70 0__NL 3
[3] PRINT SCREEN
[4] SCREEN_0µ"
```

```
™Z_V1 PSCAL V2
[1] 'ÿPRODUIT SCALAIRE DE V1 AVEC V2
[2] Z_(V1[;1]_V2[;1])+V1[;2]_V2[;2]
[3]
```

```
™RECONNAIT
[1] BOUÿ
[2] P_0 2µ0
[3] P_TRACE
[4] ,(0=(µP)[1])/0
[5] A_EXT P
[6] EXENEU
[7]
[8] TEXTFACE BOLDSTYLE
[9] TEXTSIZE 30
[10] MOVETO 70 0
[11]
[12] DRAWTEXT "S[A'®/A;]
[13] WAITMOUSEDOWN
[14] ,BOU
```

```
™RECONNAITANG
```



```

[1] BOUÿ
[2] P_TRACE
[3] A_(µA)ªEXTRACT P
[4] A
[5] EXENEU
[6] S[A'_/A]
[7] '
[8] ,BOU

```

```

™SELECT SHAPE
[1] SELECTION_SHAPE

```

```

™SHOWPAT TS
[1] MOVETO PATTERNS[TS[1];TS[2];1;]
[2] PA_1 0_PATTERNS[TS[1];TS[2];:]
[3] TPA_PA=0
[4] PA1_(~(TPA[;1]^TPA[;2]))/[1]PA
[5] LINETO PA1
[6]

```

```

™TEST
[1] PENPAT BLACKPAT
[2] _TCFF
[3] PGM_'MYPICTURE',_TCNL,'_TCFF',_TCNL
[4] WAITMOUSEDOWN
[5] TRACE GETMOUSE
[6] ,4

```

```

™POINTS_TRACE;END;NEWEND;START
[1] ZONE_10 10
[2] POINTS_0 2µ0
[3] _TCFF
[4] PENSIZE 5 5
[5] TSTFINÿ
[6] KEY__GETKEY
[7]
[8] ,(3=µKEY)/T
[9] ,TSTFIN
[10] 'ÿON TRACE
[11] Tÿ
[12] TSTÿ,(KEY^.=TRACER_FIN)/0
[13] END_GETMOUSE
[14] MOVETO END
[15]
[16] PTS_POINTS_POINTS,[1]END
[17] LOOPÿ
[18] ,(~BUTTON)/TSTFIN
[19] NEWEND_GETMOUSE
[20] ,(END^.=NEWEND)/LOOP
[21] 'ÿON TESTE SI L'ON DEPASSE UN RECTANGLE AUTOURS DE NEWEND
[22] ,(NEWEND PTINRECT (END-ZONE),(END+ZONE))/LOOP
[23] LINETO END_NEWEND
[24] POINTS_POINTS,[1]END
[25] ,LOOP

```

```

™POINTS_TRACE1;END;NEWEND;START
[1] _TCFF
[2] WAITMOUSEDOWN
[3]
[4] END_GETMOUSE

```

```

[5] MOVETO END
[6] POINTS_1 2μEND
[7]
[8] LOOP1ÿ
[9] ,(~BUTTON)μOUT1
[10] NEWEND_GETMOUSE
[11] ,(END^.=NEWEND)/LOOP1
[12] LINETO END_NEWEND
[13] POINTS_POINTS,[1]END
[14] ,LOOP1
[15]
[16] OUT1ÿ

```

™TRACER

```

[1] FIN
[2] EFFACER

```

™TST

```

[1] BOUÿ_GETKEY
[2] ,BOU

```

™Z\_TSTV

```

[1] Z_(N,2)μ0
[2] Z[;2]_1_ 'N
[3] COEF_P[;1]...P[;2]°.*'N
[4] Z[;1]_(Z[;2]°.*'N)+._COEF
[5] MOVETO Z[1;]
[6] LINETO 1 0_Z

```

™Z\_A VECT B

```

[1] 'ÿPRODUIT VECTORIEL DE A PAR B
[2] Z_(A[;2]_B[;1])-A[;1]_B[;2]
[3]
[4]

```

™WAITMOUSEDOWN

```

[1] ,(~BUTTON)/1

```

™WAITMOUSEUP

```

[1] ,(BUTTON)/1

```

™WHERE

```

[1] WAITMOUSEDOWN
[2] GETMOUSE
[3] WAITMOUSEUP
[4] ,1

```

**EXEMPLES D'IMAGES DE SYNTHÈSE et ILLUSTRATIONS***Représentation d'un visage animé souriant et/ou parlant à partir d'une modélisation par facettes structurées*

a/ *Modélisation d'un visage humain décrit par facettes* par procédé classique à partir d'une numérisation par points et *rendu réaliste par synthèse d'image*. (Extraits d'Eurographics 88). Les expressions sont obtenues par structuration de la base de données des facettes et interpolations entre les positions de ces facettes dans les différentes expressions extrêmes du visage:

ex:    Positions à 0% de rire et à 100%  
         Positions à 0% de surprise et à 100%  
         etc...

          Noter la qualité de réalisme du rendu. La modélisation pourrait également être faite grâce aux méthodes proposées dans cette thèse.

b/ *Rendu en fausse couleur* (extrait de Systems International - Fév 90)

VU: Le Président du jury

VU: Les Membres du jury

Monsieur le Professeur J.E DUBOIS

Madame le Professeur J. POINSSAC

Monsieur le Professeur J. DONIO

VU et permis d'imprimer: le président de l'Université de Droit, d'Economie et de Sciences Sociales de  
PARIS (PARIS 2)

Georges DURRY

**NOTA :**

*La thèse ayant été tirée telle quelle avant les signatures,*

*Je rajoute ci-dessous les signatures officielles du Jury.*

*Jury dont le Président fut le professeur Dubois, d'où les inversions de signatures entre le Professeur  
Dubois et le Professeur Donio*

---

MODELISATION ET ARCHITECTURE DE MACHINES de SYNTHESE D'IMAGE 3D

Page 122

VU: Le Président du jury

VU: Les Membres du jury

Monsieur J. ~~DONNO~~ <sup>E. DUBOIS</sup>



Monsieur J.E. ~~DUBOIS~~ <sup>DONNO</sup>



Madame J. POINSSAC

VU et permis d'imprimer: le président de l'Université de Droit, d'Economie et de Sciences  
Sociales de PARIS (PARIS 2)



Georges DURRTY

---

