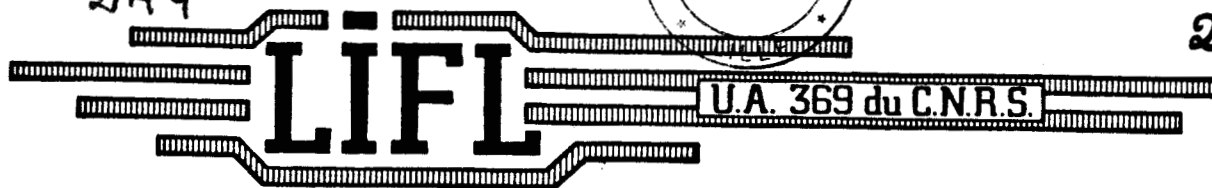


50376  
1988  
249



50376  
1988  
249

W  
R



**LABORATOIRE D'INFORMATIQUE FONDAMENTALE DE LILLE**

N° d'ordre : 425

# THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le titre de

**DOCTEUR INGENIEUR**

**en Informatique**

par

**OULD BRAHIM Djamel**

**ETUDE D'UN NOUVEAU MODELE DE PRIMITIVE  
POUR LA DESCRIPTION DES IMAGES  
NUMERIQUES : LE MACRO- PIXEL**

Thèse soutenue le 20 Sept 1988 devant la Commission d'Examen

Membres du Jury

V. CORDONNIER  
M. MERIAUX  
Ph. DURIF  
M. PELERIN  
Ph. DAUBRESSE

Président  
Rapporteur  
Examineur  
Examineur  
Examineur

UNIVERSITE DES SCIENCES  
ET TECHNIQUES DE LILLE  
FLANDRES ARTOIS

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M. H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT  
ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER,  
DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF,  
LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL,  
PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PARREAU, J. LOMBARD, M. MIGEON, J. CORTOIS.

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES  
DE LILLE FLANDRES ARTOIS

M. A. DUBRULLE.

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CONSTANT Eugène  
M. FOURET René  
M. GABILLARD Robert  
M. MONTREUIL Jean  
M. PARREAU Michel  
M. TRIDOT Gabriel

Electronique  
Physique du solide  
Electronique  
Biochimie  
Analyse  
Chimie appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre	Astronomie
M. BIAYS Pierre	Géographie
M. BILLARD Jean	Physique du solide
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean Pierre	Chimie-Physique
M. BOSCOQ Denis	Probabilités
M. BOUGHON Pierre	Algèbre
M. BOURIQUET Robert	Biologie végétale
M. BREZINSKI Claude	Analyse numérique
M. BRIDOUX Michel	Chimie-Physique
M. CARREZ Christian	Informatique
M. CELET Paul	Géologie générale
M. CHAMLEY Hervé	Géotechnique
M. COEURE Gérard	Analyse
M. CORDONNIER Vincent	Informatique
M. DEBOURSE Jean Pierre	Gestion des entreprises
M. DHAINAUT André	Biologie animale
M. DOUKHAN Jean Claude	Physique du solide
M. DYMMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du solide
M. FAURE Robert	Mécanique
M. FOCT Jacques	Métallurgie
M. FRONTIER Serge	Ecologie numérique
M. GRANELLE Jean Jacques	Sciences Economiques
M. GRUSON Laurent	Algèbre
M. GUILLAUME Jean	Microbiologie
M. HECTOR Joseph	Géométrie
M. LABLACHE-COMBIER Alain	Chimie organique
M. LACOSTE Louis	Biologie végétale
M. LAVEINE Jean Pierre	Paléontologie
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique atomique et moléculaire
M. LEROY Jean Marie	Spectrochimie
M. LHONNE Jean	Chimie organique biologique
M. LOMBARD Jacques	Sociologie
M. LOUCHEUX Claude	Chimie physique
M. LUCQUIN Michel	Chimie physique
M. MACKE Bruno	Physique moléculaire et rayonnements atmosphériques
M. MIGEON Michel	E.U.D.I.L.
M. PAQUET Jacques	Géologie générale
M. PETIT Francis	Chimie organique
M. POUZET Pierre	Modélisation - Calcul scientifique
M. PROUVOST Jean	Minéralogie
M. RACZY Ladislas	Electronique
M. SALMER Georges	Electronique
M. SCHAMPS Joel	Spectroscopie moléculaire
M. SEGUIER Guy	Electrotechnique
M. SIMON Michel	Sociologie
Mle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. TILLIEU Jacques	Physique théorique
M. TOULOTTE Jean Marc	Automatique
M. VIDAL Pierre	Automatique
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS - 2ème CLASSE

M. ALLAMANDO Etienne	Composants électroniques
M. ANDRIES Jean Claude	Biologie des organismes
M. ANTOINE Philippe	Analyse
M. BART André	Biologie animale
M. BASSERY Louis	Génie des procédés et réactions chimiques
Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mécanique
M. BELLET Jean	Physique atomique et moléculaire
M. BERTRAND Hugues	Sciences Economiques et Sociales
M. BERZIN Robert	Analyse
M. BKOUCHE Rudolphe	Algèbre
M. BODARD Marcel	Biologie végétale
M. BOIS Pierre	Mécanique
M. BOISSIER Daniel	Génie civil
M. BOIVIN Jean Claude	Spectrochimie
M. BOUQUELET Stéphane	Biologie appliquée aux enzymes
M. BOUQUIN Henri	Gestion
M. BRASSELET Jean Paul	Géométrie et topologie
M. BRUYELLE Pierre	Géographie
M. CAPURON Alfred	Biologie animale
M. CATTEAU Jean pierre	Chimie organique
M. CAYATTE Jean Louis	Sciences Economiques
M. CHAPOTON Alain	Electronique
M. CHARET Pierre	Biochimie structurale
M. CHIVE Maurice	Composants électroniques optiques
M. COMYN Gérard	Informatique théorique
M. COQUERY Jean Marie	Psychophysiologie
M. CORLAT Benjamin	Sciences Economiques et Sociales
Mme CORSIN Paule	Paléontologie
M. CORTOIS Jean	Physique nucléaire et corpusculaire
M. COUTURIER Daniel	Chimie organique
M. CRAMPON Norbert	Tectonique Géodynamique
M. CROSNIER Yves	Electronique
M. CURGY Jean jacques	Biologie
Mle DACHARRY Monique	Géographie
M. DAUCHET Max	Informatique
M. DEBRABANT Pierre	Géologie appliquée
M. DEGAUQUE Pierre	Electronique
M. DEJAEGER Roger	Electrochimie et Cinétique
M. DELORME Pierre	Physiologie animale
M. DELORME Robert	Sciences Economiques
M. DEMUNTER Paul	Sociologie
M. DENEL Jacques	Informatique
M. DE PARIS Jean Claude	Analyse
M. DEPREZ Gilbert	Physique du solide - Cristallographie
M. DERIEUX Jean Claude	Microbiologie
Mle DESSAUX Odile	Spectroscopie de la réactivité chimique
M. DEVRAINNE Pierre	Chimie minérale
Mme DHAINAUT Nicole	Biologie animale
M. DHAMELINCOURT Paul	Chimie physique
M. DORMARD Serge	Sciences Economiques
M. DUBOIS Henri	Spectroscopie hertzienne
M. DUBRULLE Alain	Spectroscopie hertzienne
M. DUBUS Jean Paul	Spectrométrie des solides

M. DUPONT Christophe	Vie de la firme (I.A.E.)
Mme EVRARD Micheline	Génie des procédés et réactions chimiques
M. FAKIR Sabah	Algèbre
M. FAUQUEMBERGUE Renaud	Composants électroniques
M. FONTAINE Hubert	Dynamique des cristaux
M. FOUQUART Yves	Optique atmosphérique
M. FOURNET Bernard	Biochimie structurale
M. GAMBLIN André	Géographie urbaine, industrielle et démographie
M. GLORIEUX Pierre	Physique moléculaire et rayonnements atmosphériques
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GOUDMAND Pierre	Chimie physique
M. GOURIEROUX Christian	Probabilités et statistiques
M. GREGORY Pierre	I.A.E.
M. GRENY Jean Paul	Sociologie
M. GREVET Patrice	Sciences Economiques
M. GRIMBLOT Jean	Chimie organique
M. GUILBAULT Pierre	Physiologie animale
M. HENRY Jean Pierre	Génie mécanique
M. HERMAN Maurice	Physique spatiale
M. HOUDART René	Physique atomique
M. JACOB Gérard	Informatique
M. JACOB Pierre	Probabilités et statistiques
M. JEAN Raymond	Biologie des populations végétales
M. JOFFRE Patrick	Vie de la firme (I.A.E.)
M. JOURNEL Gérard	Spectroscopie hertzienne
M. KREMBEL Jean	Biochimie
M. LANCRAND Claude	Probabilités et statistiques
M. LATTEUX Michel	Informatique
Mme LECLERCQ Ginette	Catalyse
M. LEFEBVRE Jacques	Physique
M. LEFEBVRE Christian	Pétrologie
Mlle LEGRAND Denise	Algèbre
Mlle LEGRAND Solange	Algèbre
M. LEGRAND Pierre	Chimie
Mme LEHMANN Josiane	Analyse
M. LEMAIRE Jean	Spectroscopie hertzienne
M. LE MAROIS Henri	Vie de la firme (I.A.E.)
M. LEROY Yves	Composants électroniques
M. LESENNE Jacques	Systèmes électroniques
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique théorique
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU Jean Marie	Optique - Physique atomique
M. MAIZIERES Christian	Automatique
M. MAURISSON Patrick	Sciences Economiques et Sociales
M. MESMACQUE Gérard	Génie Mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie organique
M. MORTREUX André	Chimie organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
M. NICOLE Jacques	Spectrochimie
M. NOTELET Francis	Systèmes électroniques
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée

M. PERTUZON Emile	Physiologie animale
M. PONSOLLE Louis	Chimie physique
M. PORCHET Maurice	Biologie animale
M. POSTAIRE Jack	Informatique industrielle
M. POVY Lucien	Automatique
M. RICHARD Alain	Biologie animale
M. RIETSCH François	Physique des polymères
M. ROBINET Jean Claude	EUDIL
M. ROGALSKI Marc	Analyse
M. ROY Jean Claude	Psychophysiologie
Mme SCHWARZBACH Yvette	Géométrie
M. SLIWA Henri	Chimie organique
M. SONME Jean	Géographie
M. STAROSWIECKI Marcel	Informatique
M. STERBOUL François	Informatique
M. TAILLIEZ Roger	Génie alimentaire
M. THERY Pierre	Systèmes électroniques
M. THIEBAULT François	Sciences de la terre
M. THUMERELLE Pierre	Démographie - Géographie Humaine
Mme TJOTTA Jacqueline	Mathématiques
M. TOURSEL Bernard	Informatique
M. TREANTON Jean René	Sociologie du Travail
M. TURREL Georges	Spectrochimie infrarouge et Raman
M. VANDORPE Bernard	Chimie minérale
M. VASSEUR Christian	Automatique
M. VAST Pierre	Chimie inorganique
M. VERBERT André	Biochimie
M. VERNET Philippe	Génétique
M. WACRENIER Jean Marie	Electronique
M. WALLART Francis	Spectrochimie infrarouge et Raman
M. WARTEL Michel	Chimie inorganique
M. WATERLOT Michel	Géologie générale
M. WEINSTEIN Olivier	Analyse économique de la recherche et développement
M. WERNER Georges	Informatique théorique
M. WOZNIAK Michel	Spectrochimie
Mme ZINN JUSTIN Nicole	Algèbre

J'exprime ma respectueuse gratitude à Monsieur le professeur V. CORDONNIER d'avoir bien voulu accepter la présidence du jury, et pour l'intérêt qu'il n'a cessé de porter à l'évolution de ce travail.

Qu'il me soit permis d'exprimer ma sincère reconnaissance à M<sup>r</sup> M. MERIAUX, chargé de recherche CNRS, qui a suivi et guidé avec compétence tous les développements de ma recherche.

Je remercie vivement M<sup>r</sup> Ph. DURIF, Assistant à Lille I, Docteur 3<sup>ième</sup> cycle, d'avoir accepté de faire partie de ce jury.

Je suis très reconnaissant à M<sup>r</sup> M. PELERIN, directeur de la société INFOGRAPHIX, et à M<sup>r</sup> Ph. DAUBRESSE, Directeur technique de la société METADESIGN, d'avoir bien voulu accepter de juger ce travail.

Mes plus vifs remerciements vont à toutes les personnes qui m'ont aidé dans la réalisation de ce travail, ainsi qu'à tous les membres du laboratoire d'informatique qui trouveront ici mention du plaisir que j'ai eu à travailler avec eux.

J'ai sincèrement apprécié la collaboration de M<sup>me</sup> C. LAVERDISSE. Je la remercie pour l'efficacité et la diligence qu'elle a apporté à la frappe de ce document.

Enfin, je remercie M<sup>r</sup> H. GLANC qui a assuré avec beaucoup de soin la préparation matérielle de ce mémoire.

# TABLE DES MATIERES

## O - INTRODUCTION GENERALE

## I - GENERALITES SUR LES SYSTEMES GRAPHIQUES

- I.1 - Notions de systèmes graphiques
  - I.1.1 - Les systèmes de traitement et d'analyse d'images
  - I.1.2 - Les systèmes de synthèse d'images
- I.2 - Notions d'images
  - I.2.1 - Modèle de Bertin
  - I.2.2 - Modèle de Martinez
- I.3 - Modèles de systèmes graphiques de synthèse
  - I.3.1 - Le modèle en couches
  - I.3.2 - Le modèle fonctionnel
- I.4 - La modélisation
  - I.4.1 - L'approche fil de fer
  - I.4.2 - Le modèle surfacique
  - I.4.3 - L'approche volumique
- I.5 - Exemples de techniques de modélisation des solides
  - I.5.1 - L'approche CSG
  - I.5.1 - L'approche BRep
  - I.5.3 - L'approche par facettes
- I.6 - Conclusion



## **II - LA MODELISATION PIXELS : SES LIMITES**

- II.1 - Introduction
- II.2 - Les dispositifs de visualisation
  - II.2.1 - La mémoire de rafraîchissement
  - II.2.2 - Systèmes à écran plat
  - II.2.3 - Résumé
- II.3 - Le modèle pixel en traitement d'images
  - II.3.1 - La compression numérique des images
    - II.3.1.1- Compression basée sur la théorie de l'information
    - II.3.1.2- Compression basée sur un modèle Contour-Texture
  - II.3.2 - La segmentation d'images
  - II.3.3 - Résumé
- II.4 - Le modèle pixel en synthèse d'images
  - II.4.1 - Codage des contours
    - II.4.1.1- Codage de Freeman
    - II.4.1.2- Codage par transformation par axe médian
  - II.4.2 - Le phénomène d'aliassage
- II.5 - Conclusion

## **III - LA MODELISATION MACRO-PIXELS**

- III.1 - Introduction
- III.2 - Le modèle macro-pixels
  - III.2.1 - La description macro-pixel
  - III.2.2 - Caractéristiques générales du macro-pixel
  - III.2.3 - Définition générale d'un pixel
  - III.2.4 - Définition de l'objet macro-pixel

- III.3 - Aspect géomorphologique du macro-pixel
  - III.3.1 - Approche par décomposition d'objets
  - III.3.2 - Approche par courbure de contour
  - III.3.3 - Les classes proposées
    - III.3.3.1 - Le choix
    - III.3.3.2 - Critères de sélection
- III.4 - Aspect visuel du macro-pixel
  - III.4.1 - Propriétés de la lumière
  - III.4.2 - Le modèle proposé
  - III.4.3 - Adéquation des classes de macro-pixels au modèle visuel
- III.5 - Les opérateurs de combinaison
  - III.5.1 - Les opérateurs de contours
    - III.5.1.1 - L'opérateur d'union
    - III.5.1.2 - L'opérateur de différence
    - III.5.1.3 - L'opérateur d'intersection
  - III.5.2 - Les opérateurs de texture
    - III.5.2.1 - Propriétés des surfaces
    - III.5.2.2 - Application aux macro-pixels
- III.6 - Les attributs des macro-pixels
  - III.6.1 - Les attributs du macro-pixel
  - III.6.2 - Les attributs des sous-objets
- III.7 - Conclusion

#### **IV - PROPOSITION D'ARCHITECTURE POUR UNE MACHINE MACRO-PIXEL**

- IV.1 - Introduction
- IV.2 - Exploitation du parallélisme dans les systèmes graphiques
  - IV.2.1 - Par découpage fonctionnel
  - IV.2.2 - Par partitionnement de l'espace image
  - IV.2.3 - Par partitionnement de l'espace objet

- IV.3 - Exemples de machines graphiques à traitement parallèle
  - IV.3.1 - Machine-action
    - IV.3.1.1 - Le Geometry Engine
  - IV.3.2 - Machine-pixels
    - IV.3.2.1 - MAP ( Multiprocesseur Associatif Parallèle)
    - IV.3.2.2 - Pixel-planes
  - IV.3.3 - Machine-objet
    - IV.3.3.1 - ARTEMIS
    - IV.3.3.2 - La machine sans mémoire de trame
- IV.4 - Adaptation au modèle macro-pixel
  - IV.4.1 - Type de partitionnement
  - IV.4.2 - Les contraintes de réalisation
  - IV.4.3 - La solution proposée
    - IV.4.3.1 - Le processeur d'interface
    - IV.4.3.2 - Le processeur-élément
    - IV.4.3.3 - Le processeur d'unification
- IV.5 - Conclusion

## **CONCLUSION GENERALE**

## **BIBLIOGRAPHIE**

## **ANNEXE**

A - algorithme de triangularisation de polygones quelconques

## **INTRODUCTION GENERALE**

## INTRODUCTION GENERALE

On assiste depuis des années à un essor considérable de l'informatique graphique, qui offre de larges possibilités d'utilisations dans presque tous les domaines où l'image existe, aussi bien scientifiques qu'industriels ou artistiques.

En effet, l'image par sa puissance suggestive est un outil tout désigné pour la communication à grande échelle de données, de symboles et autres graphismes, sans qu'il y ait lieu de réaliser un interface quelconque entre cette image et sa perception visuelle.

Généralement les dispositifs de visualisation des systèmes graphiques utilisent le modèle "pixel", modèle dans lequel une image est représentée comme une juxtaposition de points images (appelés pixels pour "picture elements") repartis dans une grille cartésienne. Chaque pixel détermine dans ce cas une valeur d'intensité ou de couleur calculée au même emplacement du monde réel ou d'une description de celui-ci.

Cependant, aussi simple qu'elle soit, une image véhicule une multitude d'informations qui n'apparaissent pas forcément dans les valeurs des pixels, tel les contours, les textures ou la transparence des objets. En effet, mise à part leur position, rien ne relie un pixel de son voisin. Cette absence de relations rend difficile la prise en compte du contenu sémantique de l'image dans ce modèle.

C'est dans ce cadre que nous nous proposons d'étudier un nouveau modèle de description d'images numériques, permettant de conserver autant que faire se peut le contenu sémantique de l'image, et ainsi de répondre efficacement aux besoins actuels de l'imagerie informatique.

Le but recherché est alors d'offrir à l'utilisateur un outil général permettant de décrire de manière simple, globale et concise tout objet graphique atténuant ainsi les redondances et les imperfections du modèle classique.

Après un bref rappel de quelques notions sur les systèmes graphiques, nous étudierons donc au chapitre II les problèmes inhérents à l'utilisation du modèle pixel, tant du point de vue

traitement que de synthèse d'images. A la lumière de cette analyse, le chapitre III sera consacré à l'étude d'un nouveau modèle de représentation d'images numériques basé non plus sur une description fine (pixel) de l'image, mais plutôt sur une description globale et "macroscopique" de celle-ci. Ainsi seront définies la notion de "macro-pixel", élément de base du modèle, et les opérateurs nécessaires à la construction d'objets graphiques à partir des macro-pixels. Enfin, nous proposons dans le chapitre IV une implémentation matérielle possible d'une "machine macro-pixel" dans le cadre d'une "architecture-objets". On trouvera de plus en annexe différents algorithmes que nous avons développés à l'occasion de ce travail.

## **CHAPITRE I**

### **GENERALITES SUR LES SYSTEMES GRAPHIQUES**

Le but de ce chapitre est de préciser quelques notions spécifiques à l'informatique graphique en général et à la synthèse d'images en particulier. Nous nous limiterons aux domaines qui nous concerneront pour la suite de cette étude. Signalons, cependant, l'existence d'une bibliographie très riche, on pourra consulter à cet effet [MORV 76] [NEW 79] [FOL 82] [PERO 88].

### **I.1 - Notions de système graphique**

L'imagerie informatique représente tout ce qui a trait à la production et au traitement d'images à l'aide d'ordinateurs.

Un système graphique est donc constitué par l'ensemble des moyens informatiques mis en oeuvre afin d'assurer le traitement, la communication, le stockage et la visualisation d'informations graphiques, au sens le plus large, liées à l'image.

Dépendant du rôle assigné, on peut distinguer deux types de systèmes graphiques :

- les systèmes graphiques dont le but est la création d'images à partir de données structurées, réalisant ainsi la synthèse d'images.
- les systèmes graphiques dont la fonction est le traitement d'images existantes.

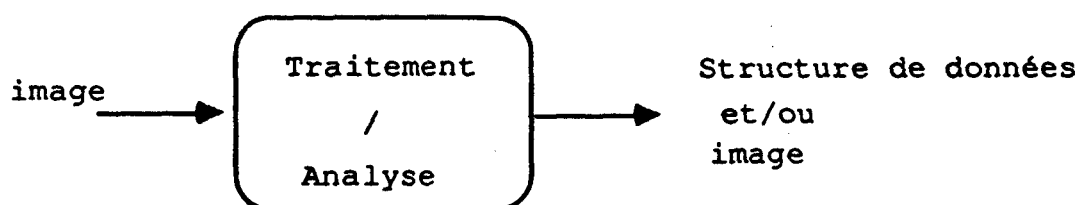
Cette distinction ne considère que les aspects fonctionnels de ces systèmes. Elle n'est considérée comme complète qu'avec la prise en compte de la composante temps. Ainsi, peuvent être distingués les systèmes graphiques temps réel et les systèmes où l'image constitue une finalité.



### **I.1.1 - Les systèmes de traitement et d'analyse d'images**

Le traitement d'images peut être défini comme étant la transformation d'images existantes , le plus souvent dans le but d'en extraire un contenu plus synthétique.

Le schéma fonctionnel du traitement d'images est le suivant :



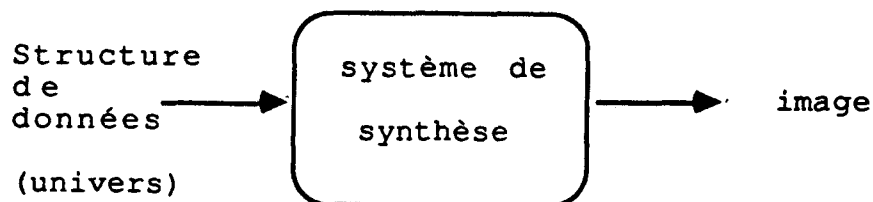
Ce type de systèmes trouve généralement son domaine d'application dans l'imagerie satellite, scanner...On distingue principalement trois types de traitement

- 1 - la restauration/amélioration d'images afin d'éliminer les défauts dus aux capteurs, mais qui par contre, demande une puissance de calcul très importante.
- 2 - l'analyse d'images dans le but d'extraire des informations d'images ou primitives.
- 3 - la reconnaissance de formes et l'analyse de scènes. Dans ce dernier cas on doit pouvoir restituer la troisième dimension afin de reconnaître les objets en amont et en aval.

### **I.1.2 - Les systèmes de synthèse d'images**

La synthèse d'images est la production, sur un dispositif de visualisation, d'images ou de dessins à partir d'une structure de données calculée ou décrite, par exemple, données produites par un programme d'application ou commandes utilisateur.

Un système de synthèse peut se représenter de la façon suivante :



Traditionnellement, l'un des domaines d'application des systèmes de synthèse d'image est la conception assistée par ordinateur (C.A.O.).

Cependant, l'utilisation de ces systèmes ne s'est pas arrêtée là, et s'est étendue à d'autres domaines tels les jeux vidéo, le cinéma, les arts graphiques, la simulation, etc...

On peut distinguer deux sortes de systèmes de synthèse :

- les systèmes produisant des images en différé : l'image est dans ce cas produite au rythme du calculateur. Son affichage a lieu dans un deuxième temps à partir d'une unité de stockage rapide.
- les systèmes produisant des images en temps réel : dans ce cas l'image est produite au rythme de sa consommation. Ce procédé implique un fonctionnement très rapide du calculateur (ex : simulateur de vol...)

## I.2 - Notions d'image

L'image étant le principal élément des systèmes graphiques, il est intéressant de pouvoir en définir les notions. Cependant, il est pour le moins difficile de déterminer un modèle conceptuel général d'images, néanmoins nous distinguerons deux modèles.

### **I.2.1 - Modèle de Bertin [BER 77]**

Dans ce modèle on considère des images bidimensionnelles constituées d'éléments séparables ou tâches. Chaque tache est définie complètement par huit variables :

- la position (x,y)
- la taille t, par rapport à une taille de référence
- le grain (i.e texture)
- la couleur (i.e teinte et saturation)
- l'orientation, par rapport à un centre de rotation et à une orientation de référence
- la forme de la tache (i.e contour)
- la valeur (i.e l'intensité).

### **I.2.2 - Modèle de Martinez [MER 82]**

Dans ce cas une image contient six classes d'informations qui sont appelées attributs :

- l'Identification qui représente le nom d'un objet dans l'univers.
- la Morphologie qui décrit la forme de l'objet.
- l'Aspect qui contient les informations intrinsèques attachées à l'objet (couleur, transparence, texture...).
- la Géométrie définissant l'emplacement d'un objet dans l'univers.
- l'Eclairage définissant les caractéristiques de l'environnement des objets (sources lumineuses, lumière ambiante).
- La Structure qui permet d'établir les relations logiques existant entre les objets.

Bien plus complet que le modèle de Bertin, ce modèle permet l'identification des objets et rend compte des relations entre l'objet et le monde extérieur (géométrie, éclairage, structure).

### **I.3 - Modèles de systèmes graphiques de synthèse**

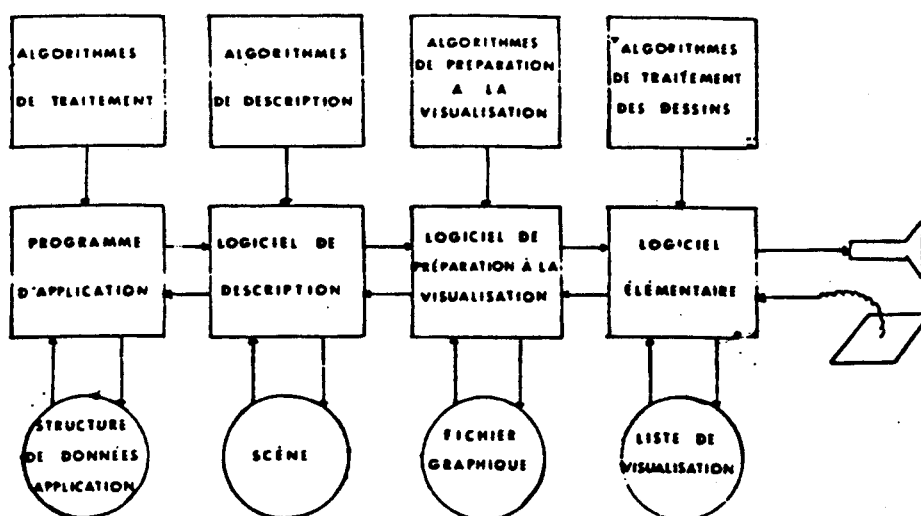
Plusieurs modèles ont été introduits, nous en présentons deux, le modèle en couche de LUCAS et le modèle fonctionnel de MARTINEZ.

#### **I.3.1 - Le modèle en couche [LUC 77] [CEI 82] [KIL 81]**

Dans ce modèle, tout système graphique est découpé en quatre couches fonctionnelles :

- le logiciel d'application constitue la première couche. Déterminant les besoins de l'utilisateur, ce logiciel manipule une structure de données qui peut ne pas être spécifiquement graphique.
- les logiciels de description permettant alors de construire l'univers graphique à partir de la couche application.
- la troisième couche est le logiciel de préparation à la visualisation. Elle permet de déterminer la sous-scène destinée à être affichée.
- Enfin, les logiciels élémentaires transcendent les informations de la sous-scène en un ensemble d'informations affichables (par exemple commandes de faisceau d'électrons sur un écran TRC). C'est cette couche qui est liée aux caractéristiques matérielles du terminal.

Le schéma est donc le suivant :



La synthèse se fait de la couche application à la couche "élémentaire." L'interaction se fait quant à elle, dans l'autre sens (de la couche élémentaire à la couche application).

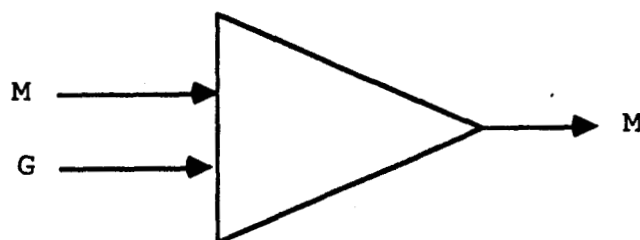
A noter que les représentations intermédiaires ne rendent pas forcément compte de la nature des objets manipulés, ce qui rend la désignation d'objets à partir des tâches produites sur le dispositif de visualisation difficile à résoudre.

### I.3.2 - Le modèle fonctionnel [MAR 82]

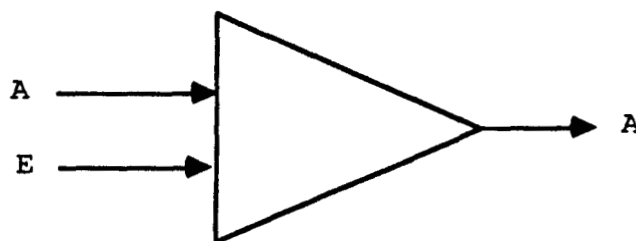
Dans ce cas, la modélisation des systèmes graphiques se fait par composition des classes d'informations d'une image (déjà décrites en I.2.2) à l'aide d'opérateurs élémentaires de synthèse. Cette méthode permet de visualiser les caractéristiques propres à un système sans avoir à distinguer logiciel et matériel.

Le traitement réalisé par chacun des opérateurs dépend des attributs sur lesquels il travaille ainsi que de la manière dont sont agencés ces opérateurs. Quelques opérateurs simples ont été décrits, parmi eux :

- la synthèse M.G qui permet de modifier la morphologie d'un objet en fonction des attributs géométriques. Le résultat étant une nouvelle morphologie (par exemple translation, rotation...)



- la synthèse A.E qui modifie les attributs d'aspects A liés à l'objet en fonction de l'éclairage ambiant E.



A noter que de telles considérations ont donné lieu à la machine HELIOS [FER 81] [MAR 82] et que des travaux sont en cours vers la définition d'un "bus" capable de supporter tout type d'opérateur.

Cependant, si les résultats dépendent de la puissance et des possibilités du système, un plus grand réalisme de l'image peut être obtenu suivant la manière dont sont décrits initialement les objets de la scène ou univers.

#### I.4 - La modélisation [PLE 88] [GRO 87] [PERO 88]

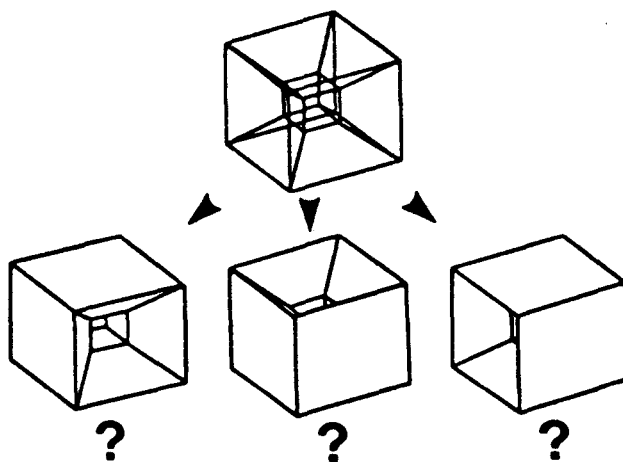
La modélisation est l'étude de la structuration des informations nécessaires à la définition de la scène. Ces informations relatives aux objets constituant la scène, doivent décrire le plus précisément possible les caractéristiques intrinsèques de ces derniers, ainsi différents modèles ont été proposés et utilisés. Ces systèmes présentent des capacités très différentes et inégales.

Les premiers modèles introduits furent du type fil de fer, dans lesquels les objets étaient mémorisés sous forme de suite de segments 3D. Cependant les inconvénients de ce modèle ont conduit d'abord à l'approche surfacique permettant de tenir compte des propriétés géométriques des objets, ensuite au développement de modèles volumiques solides, où les propriétés tant géométriques que topologiques sont prises en compte.

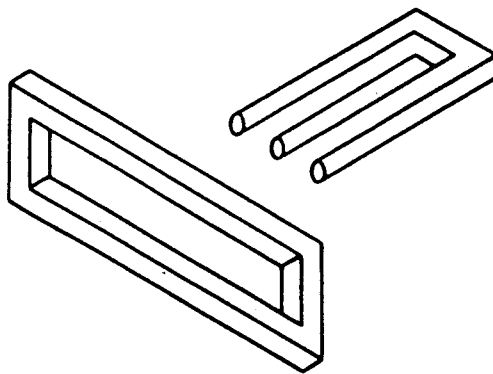
#### I.4.1 - L'approche fil de fer

Dans ce modèle les arêtes de chaque objet sont mémorisées en tant que suite de segments 3D et d'éléments circulaires. Ces représentations sont les plus simples à créer et ont l'avantage d'utiliser un minimum de puissance et de mémoire informatique. Cependant, cette modélisation présente trop d'inconvénients pour être utilisée souvent.

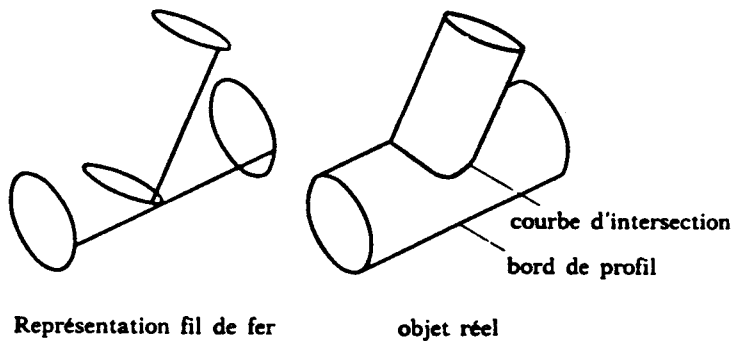
Tout d'abord, l'absence d'identification du vide ou du plein rend parfois ambiguë l'interprétation de l'objet comme le montre la figure suivante



Par ailleurs, les systèmes fil de fer tolèrent l'existence de "non-sens" dans la conception.

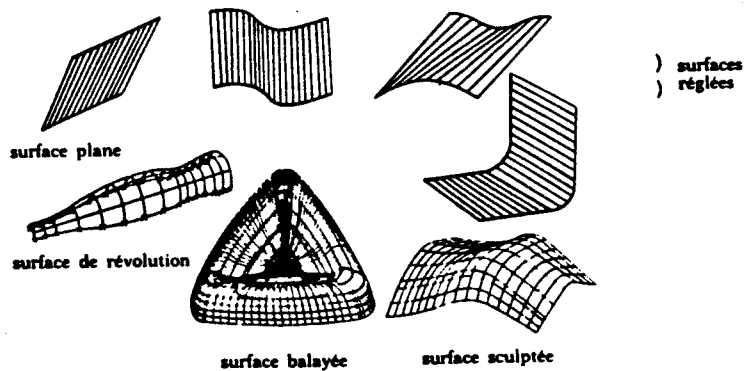


De plus, l'absence de définition des surfaces rend impossible le calcul des courbes d'intersection.



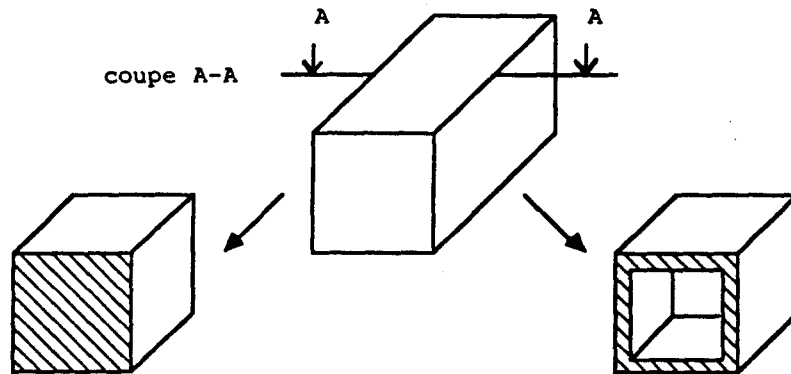
#### I.4.2 - Le modèle surfacique [VERO 87] [BEZIER 86]

Les nombreuses ambiguïtés inhérentes à l'approche fil de fer sont évitées par les modéleurs surfaciques. En effet, les modèles d'objets sont créés en associant des éléments de surface à des contours délimités par les arêtes. Plusieurs types de surfaces peuvent être utilisés : les surfaces planes, les surfaces réglées, les surfaces gauches, les surfaces de Bézier, les surfaces de révolution...





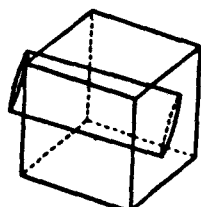
Bien que supérieur à l'approche fil de fer, l'approche surfacique présente certaines faiblesses. Ainsi des problèmes d'intégrité et de cohérence du modèle peuvent se poser étant donné que chaque surface d'objet est décrite indépendamment des autres. De plus, un tel modèle ne permet pas de réaliser des coupes automatiques de l'objet.



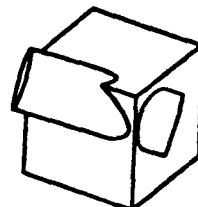
Les risques d'erreurs demeurent donc assez importants et la compréhension difficile.

#### I.4.3 - L'approche volumique

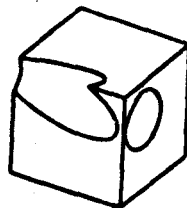
Cette approche permet de décrire et de visualiser sans ambiguïté des objets solides comme, par exemple, des pièces ou des systèmes mécaniques. La méthode de construction de ces modèles consiste à décrire la forme finale à partir de volumes plus simples et à réaliser des opérations d'addition, soustraction ou intersection entre ces pièces selon une logique booléenne.



Bloc et cylindre



Union



Différence



Intersection

Ainsi, la plupart des limites propres aux techniques précédentes disparaissent. On peut, par exemple, réaliser des vérifications de collisions, des calculs automatiques des propriétés mécaniques, fournir des vues complètes ou en coupe avec élimination des lignes cachées.

Néanmoins, il paraît difficile dans ces modèles, de représenter des objets naturels comme les arbres, le feu, les nuages. Ce qui a montré les limites de ces modèles, et a permis le développement d'autres modèles plus complexes tels les modèles récursifs (fractals et graftals) [SMIT 84], les modèles stochastiques [REEV 83] [REEV 85].

### **I.5 - Exemples de techniques de modélisation des solides [REQ 80] [REQ 85]**

Plusieurs techniques de modélisation des solides ont été proposées.

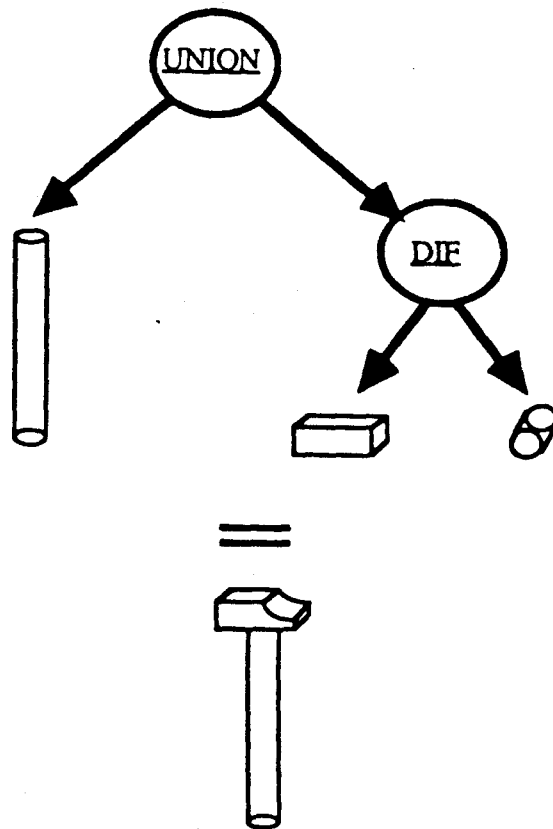
Nous en distinguons trois, les plus utilisées :

- l'approche CSG (Constructive Solid Geometric)
- l'approche BRep (Boundary Representation)
- l'approche par facettes.

#### **I.5.1 - L'approche CSG [LAID 86] [GOLD 86]**

Ce modèle permet de décrire un solide à l'aide d'un ensemble d'opérateurs de composition (union, différence, intersection) et d'un ensemble de volumes de base simples tels que : sphères, blocs, cylindres, cônes, tores...

L'objet est donc représenté par un arbre binaire dont les noeuds sont des opérateurs de composition et les feuilles des volumes de base, éventuellement accompagnées d'une matrice de transformation. La figure suivante nous donne un exemple de représentation.

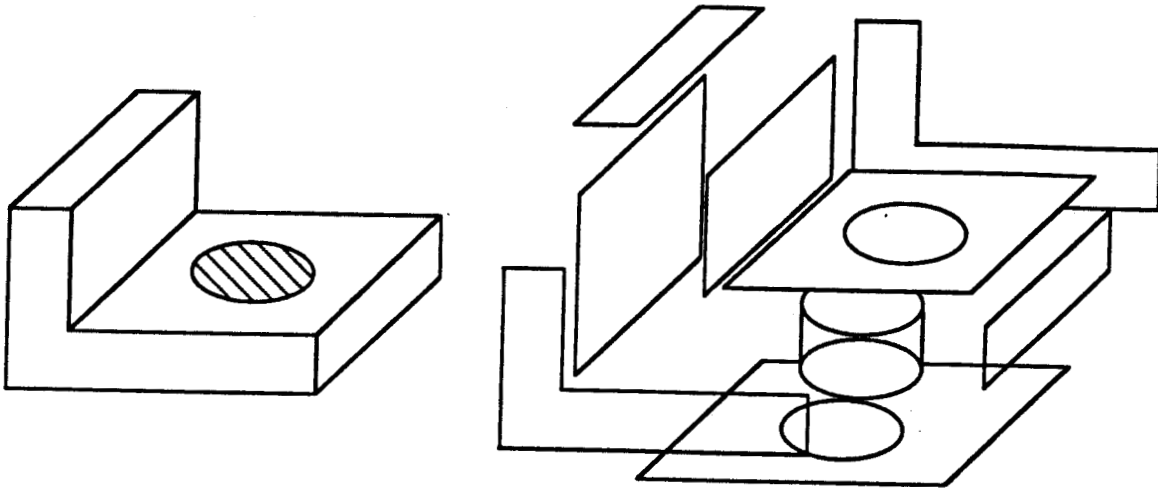


L'arbre CSG de représentation d'un objet a l'avantage de fournir une description compacte du modèle, facilitant le stockage. Cependant, un tel modèle est peu adapté à la description des visages, des plantes. Des extensions ont été développées résolvant partiellement ce problème comme les déformations par exemple [SEDE 86].

A noter une bibliographie très riche concernant l'arbre de construction CSG.

### I.5.2 - L'approche par représentation par les bords ou BRep [MICH 87]

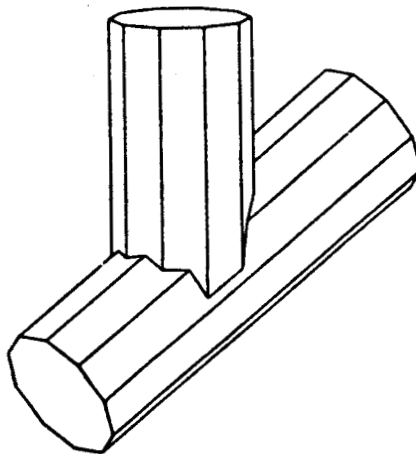
Ce modèle représente un volume par sa surface extérieure, elle-même représentée par un ensemble de faces ou de morceaux de surfaces. Ainsi, ce modèle permet une définition explicite des arêtes, des faces, de l'orientation de ces faces et de leur connectivité. Un exemple de modèle est donné par la figure suivante :



Une telle description est naturelle et facilite les transformations géométriques, car elles peuvent être appliquées aux sommets des faces ou des morceaux de surfaces. Cependant, il est pour le moins difficile de réaliser des opérations logiques (union, intersection, différence) entre deux volumes comportant un grand nombre de faces.

### I.5.3 - L'approche par facettes

Le principe de cette méthode consiste à approcher la forme réelle d'un objet par un ensemble de surfaces planes (facettes).



Le nombre de facettes utilisées permet de déterminer le degré d'approximation. Dans ce cas, si le nombre de facettes est faible, l'opération d'élimination des faces cachées est rapide, par contre le résultat visuel est parfois grossier et le calcul des propriétés mécaniques risque d'être très approché. Néanmoins, l'occupation mémoire reste faible. Dans le cas d'un grand nombre de facettes, la précision du modèle peut être acceptable, bien qu'elle n'atteint pas celle des modèles CSG et BRep. Cependant, la génération des vues devient extrêmement longue et exige des capacités mémoire très importantes.

A noter, que les algorithmes de visualisation acceptant directement ces modèles, en particulier le CSG, utilisent la technique de Lancer de rayon [ROTH 82], qui permet un grand réalisme de l'image en simulant le cheminement de la lumière. Cependant, le temps de réponse peut être prohibitif eu égard au grand nombre de traitements à effectuer.

## 1.6 - Conclusion

Nous avons présenté un bref tour d'horizon des systèmes graphiques et de leurs fonctionnalités. Nous avons particulièrement développé les techniques de modélisation de la scène ou univers. Cette étape constitue la première partie de la synthèse d'images. Par la suite un certain nombre de techniques sont applicables afin de déterminer l'aspect final de l'image. Des techniques diverses existent tel le lancer de rayon ou le calcul d'ombrage.

En traitement d'images, par contre, c'est à partir de l'image visualisée que sont réalisées les fonctions de reconnaissance de formes, d'analyse de scène ou tout simplement de compression numérique des images en vue de leur stockage ou de leur transmission.

Ainsi l'image visualisée ou sa représentation immédiate constitue un élément fondamental de ces systèmes. De ce fait, la manière dont sont alors codés les objets dans l'image détermine à ce stade les performances du modèle de visualisation.

Généralement, le modèle le plus adapté aux dispositifs de visualisation actuels est le modèle pixels. Cependant, un tel modèle possède néanmoins ces limites, c'est ce que nous allons voir dans le chapitre suivant.

## **CHAPITRE II**

### **LA MODELISATION PIXEL : SES LIMITES**

## II.1 - introduction

D'une manière générale, une image permet de représenter et de véhiculer un certain nombre d'objets synthétiques, artistiques ou autres, pouvant avoir ou non une signification sémantique à l'échelle humaine. Ces différents objets sont regroupés sous le terme d'objets graphiques.

Pour être transmise, et donc perçue, une image doit être alors adaptée aux supports de transmission et de visualisation. On parle dans ce cas d'images numériques. L'aspect essentiel de cette adaptation réside dans le choix d'une structure de données appropriée à la représentation de l'information à véhiculer et/ou à manipuler.

Cette structure doit alors refléter de façon aussi précise que possible la nature des paramètres que l'image doit supporter, i.e les différents aspects visuels et sémantiques des objets graphiques de l'image.

Une manière classique et conventionnelle de représenter une telle image numérique est de la considérer comme une juxtaposition de points images (appelés aussi pixels, pour picture elements), ayant chacun comme caractéristique un niveau de gris ou de couleur uniforme relevé à l'emplacement correspondant dans l'image réelle, ou calculé à partir d'une description interne de la scène à représenter.

On parle alors de représentation sous forme canonique d'une image numérique (Fig. II.1).

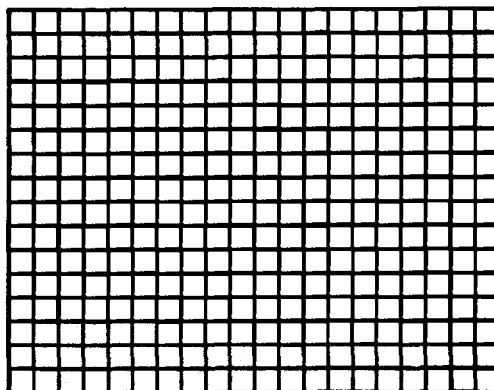


Fig.II.1 - Représentation canonique d'une image numérique



L'existence de dispositifs de visualisation adaptés à un tel modèle de représentation, tels que écran de télévision ou autres équivalents, milite vers son utilisation dans les systèmes de synthèse et de traitement d'images.

Cependant, il va sans dire que l'utilisation d'un tel modèle ne va pas sans poser certains problèmes spécifiques que ce soit en synthèse ou en traitement d'images, qui constituent les deux grands domaines de l'infographie.

De plus, il est pour le moins difficile dans ce modèle de représenter et de désigner les différentes caractéristiques propres aux objets, tel la continuité des formes ou leurs aspects. En effet, mis à part leur positionnement dans la grille cartésienne, rien ne relie un pixel de son voisin ou d'un autre pixel. La seule caractéristique possible qu'on peut considérer est la quantité d'information apportée par le pixel, qui est d'ailleurs théoriquement répartie de manière identique sur tous les pixels formant l'image. Et même dans ce cas, il apparaît que cette seule information n'est pas suffisante [KOCH 83].

Dans ce contexte, un certain nombre de travaux ont été menés dont le but final est d'offrir aux différents utilisateurs de l'imagerie informatique - pour reprendre un terme en vogue actuellement - d'autres moyens de représentation d'objets permettant de prendre en compte quelques aspects des divers objets visibles ou à visualiser dans une image. Nous citerons entre autres : l'étude réalisée par [MER 79] pour la réalisation d'un terminal graphique couleur fonctionnant par taches, l'architecture du système graphique d'ARTEMIS [GRU-CAT 81] basée sur un grand nombre de "processeurs-triangle" sous forme d'un pipe-line, l'étude sur la compression d'images numériques basée sur un modèle contour-texture [KOCH 83], ou celle plus récente de [ALT 88] étudiant d'autres motifs de pixels sur de nouvelles technologies d'écrans.

Ainsi se distinguent deux grands axes, guidés par

- 1) Les raisons technologiques, particulièrement pour les dispositifs de visualisation tels les écrans à cristaux liquides, à plasma.
- 2) Les exigences de la représentation interne et des applications
  - compatibilité géométrique
  - aptitude de l'image à la manipulation ou au calcul.
  - interactivité de l'image et les possibilités de désignation d'objets (crayons optiques, souris).
  - capacités de stockage et de transmission (compression).

Ces inconvénients nous ont amenés naturellement à étudier un nouveau modèle de représentation d'images basé non plus sur une description pixel d'une image, mais plutôt comme nous le verrons plus loin, sur une description plus globale, macroscopique de celle-ci.

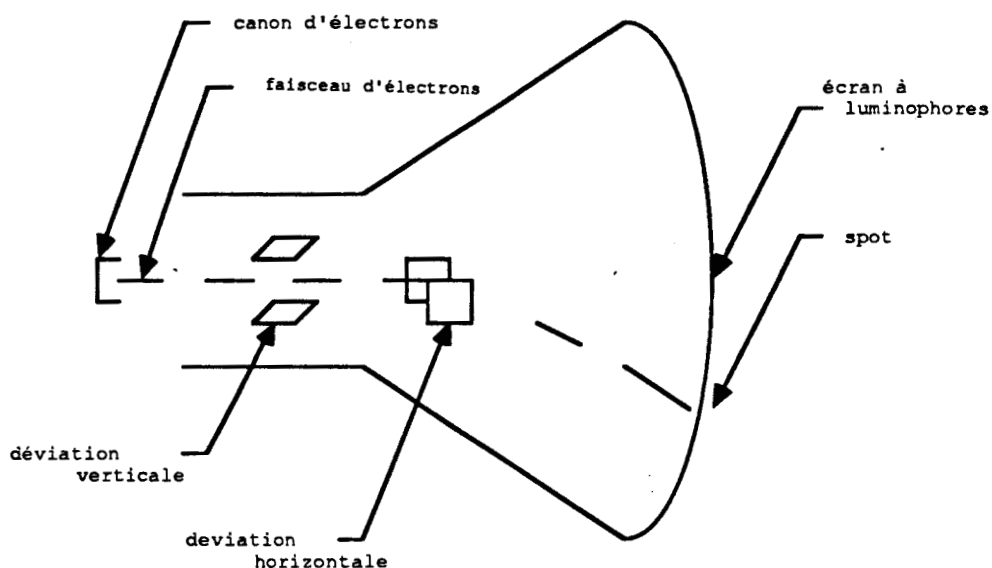
Pour cela, après un bref rappel des systèmes de visualisation, particulièrement les tubes à rayons cathodiques (TRC) équipant à présent tous les dispositifs de visualisation, nous verrons dans les paragraphes 3 et 4 les conséquences - avantages et inconvénients - du modèle canonique d'une image dans les systèmes de synthèse et de traitement d'images.

## **II.2 - Les dispositifs de visualisation**

Parmi les dispositifs d'affichage d'images, les écrans TRC sont actuellement les plus répandus et dominant largement le marché.

Ce type d'écran (Fig. II.2) permet le contrôle :

- de l'intensité lumineuse, grâce au canon à électrons et à la couche de luminophores de la surface d'affichage.
- des coordonnées (x,y) du point qu'on veut atteindre, grâce aux deux systèmes de déviation horizontal et vertical.



**Fig. II.2 - Principe d'un tube à rayons cathodiques (TRC)**

Cependant, la lumière émise par les luminophores perd rapidement de son intensité. Ce qui impose, pour assurer la présence de l'image sur l'écran, un mode de maintien de celle-ci à une fréquence déterminée dépendant de la persistance lumineuse des luminophores. On distingue dans ce cas deux modes de balayage :

- le balayage cavalier, l'intensité du faisceau est alors commandée en fonction du mode de tracé : canon éteint, canon traçant. L'image est généralement constituée de traits continus.

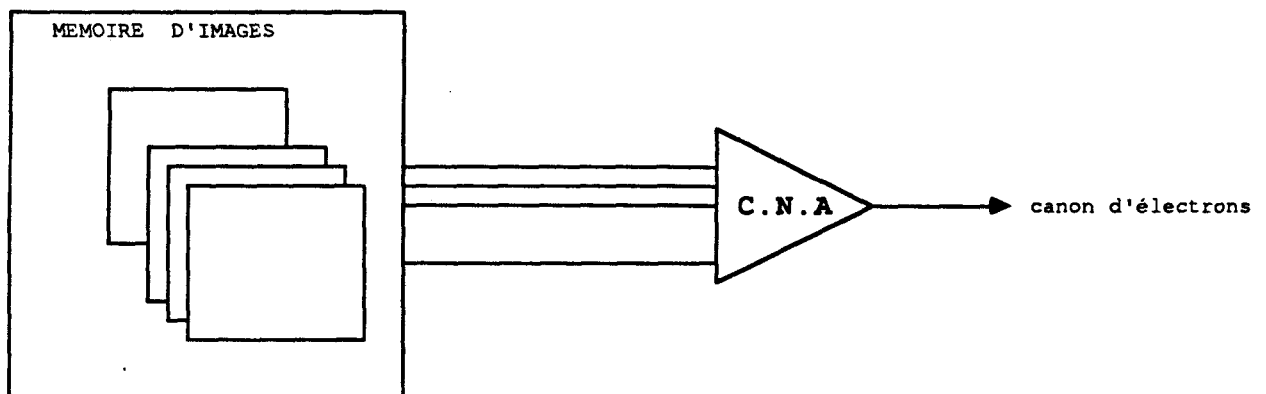
- le balayage de trame ou reccurrent, et dans ce cas, le faisceau est dirigé pour parcourir en permanence l'écran de haut en bas à une fréquence  $F$  (50 à 60 Hz, selon le pays, en télévision).

Le maintien de l'image sur l'écran implique nécessairement un dispositif de stockage permettant le réaffichage de l'image.

### II.2.1- La mémoire de rafraîchissement

Cette mémoire a une structure différente selon qu'on utilise un balayage cavalier ou de trame. Dans le premier cas, cette mémoire contient la liste des coordonnées des segments constituant l'image : c'est la liste de

visualisation. Le faisceau d'électrons semble ainsi se diriger de manière aléatoire sur l'écran. Dans le second cas, la mémoire de rafraîchissement, appelée aussi mémoire d'image, est structurée comme un tableau à deux indices, avec un ou plusieurs plans (Fig. II.3). Chaque élément de cette mémoire correspond à un pixel de l'image numérique et sa valeur est une information relative à la couleur de ce pixel. Cette information est généralement codée sur un mot de 1 à 12 bits et même parfois 24 bits, dépendant du nombre de couleurs voulues.



**Fig. II.3 - Mémoire de rafraîchissement à balayage de trame**

Les écrans à TRC à rafraîchissement utilisant un balayage de trame sont actuellement les plus utilisés. Cependant, des contraintes d'ordre technologique tel que l'accès rapide à la mémoire d'image, la taille de cette mémoire ou un temps court pour effectuer les conversions numériques-analogiques, limite la définition d'affichage (i.e, le nombre de pixels affichés) pour une image de bonne qualité.

En effet, si chaque information du pixel est codée sur 24 bits, la taille de la mémoire d'image devra être de 192 koctets pour une image moyenne de 256 x 256 pixels, et de 768 koctets pour une image de 512 x 512 pixels.

De plus, des phénomènes d'aliassage (effets de marche d'escalier sur l'image) ou des problèmes de distorsion géométrique, s'accroissant avec la taille de l'écran, apparaissent.

## **II.2.2 - Systèmes à écran plat**

Les inconvénients des écrans à TRC ont abouti à l'apparition des écrans plats. Leur caractéristique principale est que la surface de l'écran est plate et leur profondeur est constante et indépendante de la surface. Selon le mode d'affichage, on distingue principalement deux catégories d'écrans plats :

- Ecrans plats émetteurs, tels les écrans à plasma ou à diodes électroluminescentes (LED).
- Ecrans plats non émetteurs, tels les écrans à cristaux liquides.

Cependant, les écrans plats sont coûteux, posent encore un certain nombre de problèmes d'ordre technologique et surtout le bon rendu des couleurs s'obtient difficilement.

### **II.2.2.1 - Ecrans à plasma**

Un écran à plasma peut être considéré comme un dispositif constitué de deux plaques de verre entourant des cellules minces (d'une épaisseur de l'ordre de 1/10 de millimètre) remplies de néon. Chaque plaque contient un réseau de conducteurs. Les deux plaques déterminent ainsi deux nappes orthogonales de fils. La coïncidence de deux signaux sur deux conducteurs vertical et horizontal, provoque l'ionisation locale du néon et permet ainsi de sélectionner une cellule donnée.

### **II.2.2.2 - Ecrans à cristaux liquides**

A l'état initial, les écrans à cristaux liquides sont transparents et réfléchissent la lumière. Lorsqu'ils sont soumis à une tension électrique, ils deviennent opaques, ce qui permet d'utiliser toutes sortes de figures à

la place du pixel. Les cristaux liquides ont une faible consommation d'énergie et sont donc bien adaptés à des dispositifs portatifs tels que les instruments de mesure ou les montres.

### II.2.3 - Résumé

Le tableau II.1 regroupe brièvement les avantages et les inconvénients des écrans à TRC, principalement à balayage de trame, et des écrans plats.

**Tableau II.1 - Récapitulatif des avantages et inconvénients des différents systèmes de visualisation**

type d'écrans	avantages	inconvénients
Ecrans TRC	<ul style="list-style-type: none"> <li>- simplicité de mise en oeuvre.</li> <li>- adapté au modèle pixel d'une image.</li> </ul>	<ul style="list-style-type: none"> <li>- taille de la mémoire d'image peut être grande.</li> <li>- problèmes d'accès rapide à cette mémoire.</li> <li>- taille de l'écran faible</li> <li>- résolution moyenne.</li> </ul>
Ecrans plats	<ul style="list-style-type: none"> <li>- surface d'écran plate.</li> <li>- permet de résoudre les problèmes de distorsion géométrique des TRC.</li> </ul>	<ul style="list-style-type: none"> <li>- coûteux</li> <li>- rendu des couleurs difficile.</li> </ul>

Après avoir vu les dispositifs de visualisation, et pour bien situer notre étude, il nous a paru intéressant de déterminer les limites du modèle pixel dans les domaines de traitement et de synthèse d'images. En effet,

l'utilisation massive des images a créée de nouveaux besoins de plus en plus pressants. Ces exigences aboutissent indéniablement à un surcoût, qui parfois, est non négligeable dans le traitement ou la synthèse de l'image. Ce qui a montré les limites du modèle canonique, par les effets indésirables qui apparaissent parfois dans l'analyse d'une image ou dans les effets de marches d'escaliers (aliassage) et de calcul d'ombrage par exemple.

Dans les deux paragraphes qui vont suivre, on essaiera d'évaluer les difficultés rencontrées, par l'utilisation de la représentation pixel, dans certaines applications du traitement et de la synthèse d'images.

### **II.3 - Le modèle pixel en traitement d'images**

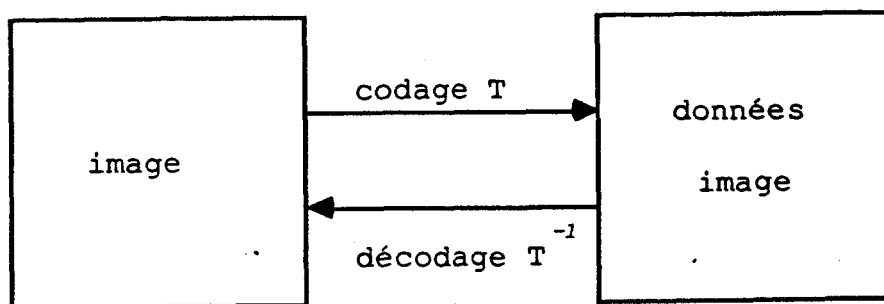
Le traitement d'images regroupe tout ce qui est analyse par extraction de primitives ou d'attributs, reconnaissance de formes, compression d'images pour la transmission et le stockage...

Parmi les différents problèmes qu'on peut rencontrer dans ce domaine, et afin de nous limiter dans le cadre de notre étude, nous avons retenu ceux qui peuvent se poser à la compression numérique des images et à la segmentation d'images, afin de regrouper certains attributs ne se trouvant pas dans les pixels isolés de celle-ci.

Ce choix a été fait dans le sens qu'un codage adéquat d'une image permet d'atténuer le problème de la taille de la mémoire d'images, qui comme nous l'avons vu précédemment constitue un inconvénient majeur des dispositifs de visualisation lié au modèle pixel d'une image. De plus, une image devant véhiculer des objets, il nous a paru intéressant de voir les problèmes pouvant se poser à l'extraction d'informations relatives à ces objets, à partir du modèle pixel d'une image. D'où l'intérêt accordé à la segmentation d'images, domaine clé pour la reconnaissance des formes.

#### **II.3.1 - La compression numérique des images**

On peut définir le but de la compression, ou plus généralement du codage d'images, comme étant la réduction du plus grand nombre possible de bits nécessaire à représenter et à reconstituer une réplique fidèle de l'image originale (Fig. II.4).



**Fig. II.4 - Codage d'images**

Dans les différentes approches du problème de la compression des images numériques, on distingue deux grands axes :

- l'un basé sur les résultats de la théorie de l'information en exploitant les corrélations existant entre les pixels
- l'autre basé sur un modèle d'étude des mécanismes du système visuel humain [MARR 80].

### **II.3.1.1 - Compression basée sur la théorie de l'information**

[JAIN 81] [NETRA 80] [OULD 87].

Une première analyse des images numériques montre que des points images adjacents possèdent généralement des niveaux de gris similaires ou proches, montrant ainsi une corrélation spatiale importante. En exploitant cette corrélation, il est alors possible de réduire les "données brutes" d'une image.

Un grand nombre de méthodes basées sur cette approche ont été développées. On peut les regrouper en trois grandes catégories :

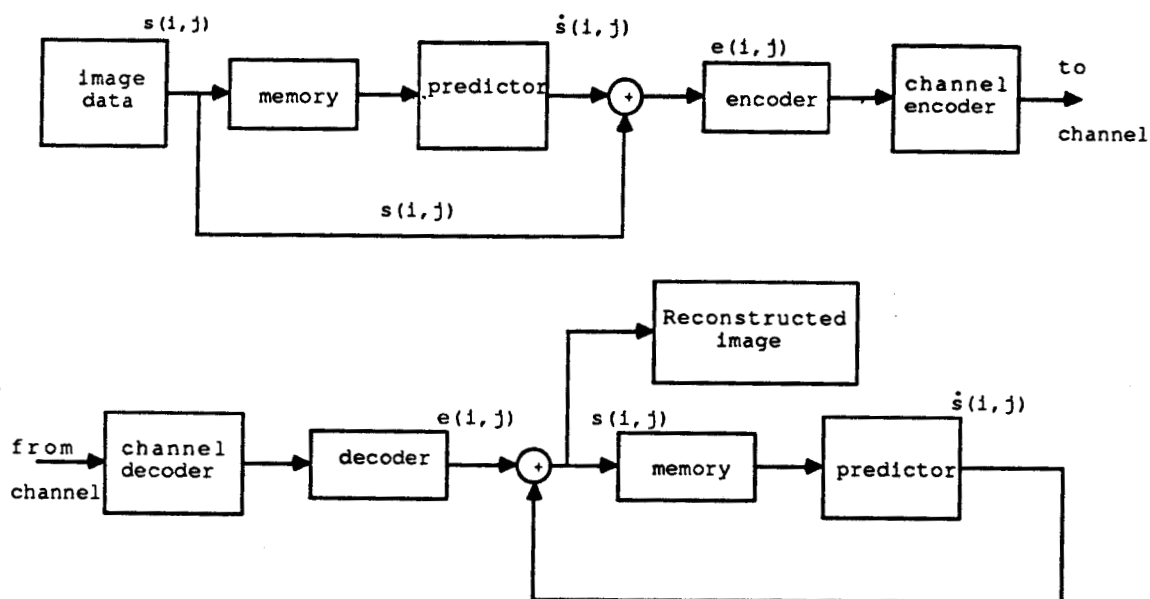
#### **II.3.1.1.1 - Méthodes prédictives**

Ces méthodes de compression ont été les plus développées et sont surtout utilisées dans la transmission d'images (TV).

Dans ce cas, la valeur de chaque pixel est prédite à partir des pixels précédemment codés et transmis, et seule la valeur de l'erreur (ou écart) entre la valeur prédite et la valeur réelle est quantifiée puis codée et transmise. La figure II.5, nous donne un exemple de système de compression



avec un codage par prédiction. Dans ce contexte la méthode de prédiction linéaire est la plus intéressante, dont le DPCM (Differential Pulse-Code Modulation) est un cas particulier et d'ailleurs le plus utilisé [KOBA 74].



**Fig. II.5 - Système de compression d'images utilisant un codage prédictif [KOBA 74].**

Les méthodes prédictives permettent une mise en oeuvre facile et conduisent à un taux de compression allant de 2 à 4 [KUNT 84]. Cependant, elles n'atteignent pas le taux théorique de compression décrit par Shanon en théorie de l'information. Ceci vient du fait qu'il est difficile de trouver un modèle mathématique adéquat de la source d'informations, alors que le calcul de la distribution statistique des pixels est rendu impossible par le grand nombre d'images possibles, et ne peut être fait que dans un cadre local ou restreint. D'autre part, ces méthodes sont inefficaces dans le traitement de l'information contour d'une image.

Ces insuffisances ont amené les chercheurs à s'intéresser à d'autres méthodes utilisées dans le traitement des signaux telle la transformé de Fourier.

### II.3.1.1.2 - Méthodes par transformation orthogonale

Ce type de codage est basé sur le fait que la distribution de l'énergie dans le domaine transformé d'une image est plus adéquate pour le codage, et par cela pour la compression, que sa représentation spatiale.

En effet, l'utilisation des transformations orthogonales connues telles les transformés de Fourier, Walsh, Hadamard, Slant, Haar, Cosinus, Karhunen-Loève..., permet le passage d'un ensemble de pixels, statistiquement dépendant, à un ensemble de coefficients plus ou moins indépendants [WINTZ 72]. L'élimination ou le codage grossier des coefficients jugés non significatifs, permet alors un codage efficace du point de vue de la compression. Les coefficients restant sont alors quantifiés et codés en vue de leur transmission ou de leur stockage. Pour retrouver l'intensité des pixels, on applique alors sur ces coefficients la transformation inverse. Le schéma global regroupant toutes ces opérations est donné par la figure II.6. A noter que dans ces méthodes, l'image est divisée en plusieurs blocs (ou sous-images) et chaque bloc est alors transformé en un ensemble de coefficients.

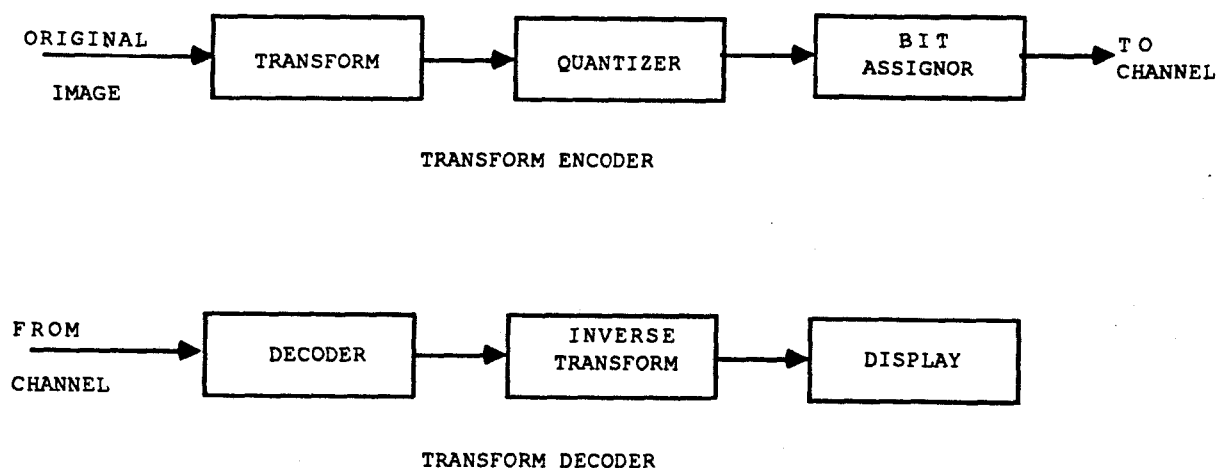


Fig.II.6 - Schéma bloc d'un codage/décodage par transformé ([NETRA 80])

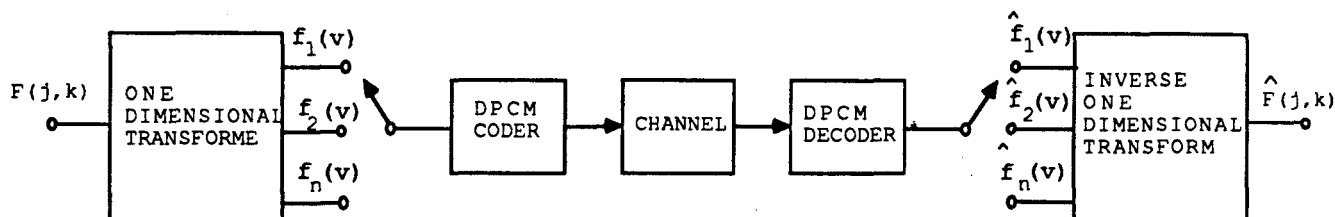
Sans détailler les aspects théoriques des transformations (pour de plus amples détails se référer à [HAR 72] [BEAU 84]), en définitive pour une sous-image  $N \times N$  pixels, chacune des transformations nous donne un ensemble de  $N^2$  coefficients, chacun étant une combinaison linéaire des valeurs des  $N^2$  pixels du bloc considéré. Deux principales techniques sont alors applicables pour la sélection des coefficients à coder afin d'achever la compression [PRAT 78] :

- **le codage par zone** : technique non adaptative consistant à choisir un nombre  $n$  de coefficients possédant une large variance. Les coefficients à écarter sont alors mis à zéro.
- **le codage par seuil** : Dans cette technique adaptative, on ne code que les coefficients dont l'amplitude dépasse un seuil fixé au préalable et sont alors quantifiés avec un nombre fixe de niveaux de quantification. Il est alors nécessaire de coder la position de chaque coefficient.

En résumé, quoique assez performantes, les méthodes par transformation n'arrivent pas à dépasser nettement les taux de compression atteints par les méthodes prédictives. Cependant, contrairement à ces dernières, les méthodes par transformation distribuent les dégradations uniformément sur toute l'image.

#### II.3.1.1.3 - Le codage hybride

Ce type de codage permet de combiner les techniques prédictives, comme le DPCM, et transformationnelles afin de bénéficier de la simplicité de l'un et de l'efficacité de l'autre. Un système de codage hybride transformation/DPCM d'une image monochrome est donné par la figure II.7.



**Fig. II.7 - Système de codage hybride transformation/DPCM  
d'une image monochrome [PRAT 78].**

L'intérêt accordé aux méthodes hybrides est que d'une part dans les systèmes de codage par transformation on néglige les redondances existant entre les blocs, d'autre part la transformation de larges blocs, quoique plus intéressante du point de vue théorique, rend l'implémentation plus complexe. Par contre, dans les méthodes hybrides cette redondance est alors exploitée par l'utilisation du DPCM. De plus, la transformation de petits blocs est une opération simple à réaliser.

En résumé, le tableau II.2 [KOCH 83] nous donne le taux de compression atteint par les différentes méthodes qu'on vient de voir.

Tableau II.2 - Taux de compression par méthode de compression

méthode	nombre moyen de bits/pixel
MIC (PCM)	6-8
Codage prédictif	
- MIC différentiel (DPCM)	2-3
- Modulation delta	1
codage par transformation	
- non adaptatif	1 - 1.5
- adaptatif	0.5 - 1
codage hybride	
- non adaptatif	1-2
- adaptatif	0.5 - 1.5

### II.3.1.2 - Compression basée sur un modèle Contour-Texture

[KOCH 83] [KUNT 84] [OULD 88]

Les méthodes de compression basées sur les résultats de la théorie de l'information, réalisent un codage d'images en ne tenant compte que des seules valeurs des pixels, disposées dans la grille cartésienne.

Cependant, les problèmes posés par la modélisation de la source d'informations, ainsi que de certaines données supportées par l'image mais non contenues de manière explicite dans les valeurs isolées des pixels, telles que contours, textures, micro-textures, ont limité les performances attendues de ces méthodes. Il apparaît donc, à ce stade, qu'une prise en compte de ce type d'information (contour, texture, micro-texture ou type d'objet) permettra une compression efficace de l'image.

En effet, si l'on considère que le but du codage est de réduire l'information effective d'une image sans en altérer l'information "sémantique", définie ici comme étant l'information nécessaire à la reconnaissance d'une image, la connaissance de cette dernière permettrait alors un codage adéquat et compact de l'image.

Or, les résultats de neurophysiologie et de psychologie de la vision montrent que dans la perception d'une scène, un observateur humain ne traite pas de manière quantitative les luminances ou les valeurs de chaque pixel. Il apparaît que l'observateur cherche à distinguer les formes contenues dans l'image telles que textures et surtout contours, et parfois il approxime mentalement les traits perçus à un groupe reconnaissable [JERN 81] [MARR 80].

Ces résultats permettent de distinguer, du moins à ce stade, deux classes de points images sémantiquement différentes :

- La classe des points contours qui, du point de vue sémantique, est la plus importante.
- La classe des points texture qui, par contre, est relativement faible en apport sémantique.

On peut classer les méthodes de ce groupe en deux catégories. La première permet généralement d'utiliser ou d'adapter les méthodes classiques au contenu et surtout à l'aspect contour d'une image [WIL 83] [BURT 83].

La seconde catégorie préconise d'utiliser directement une modélisation Contour-Texture par extraction puis compression de ces deux entités. Une expérience de cette catégorie, réalisée par [KOCH 83] conduit à représenter une image non plus comme un ensemble de points placés de manière homogène sur une grille cartésienne, mais plutôt comme une mosaïque de régions connexes, de formes diverses et dont le contenu suit un certain critère d'approximation. Le taux atteint par cette méthode est de l'ordre de 50 à 1, ce qui a permis de dépasser nettement ceux obtenus jusque là par les autres méthodes.

Néanmoins, l'inconvénient majeur de ce groupe de méthodes est qu'au stade actuel des développements, il est pratiquement impossible de réaliser une application en temps réel, sans préjuger des problèmes de coût et de rentabilité de la méthode.

On voit dès lors, que les difficultés d'extraction d'information sémantique, dans le sens de la reconnaissance d'une image, à partir de la représentation pixel de celle-ci limitent l'utilisation des nouvelles méthodes de codage d'image très performantes, qui ne peuvent être réalisées que par un surcoût non négligeable.

Nous allons voir maintenant une autre technique utilisée en traitement d'images, en l'occurrence la segmentation et les problèmes y afférant.

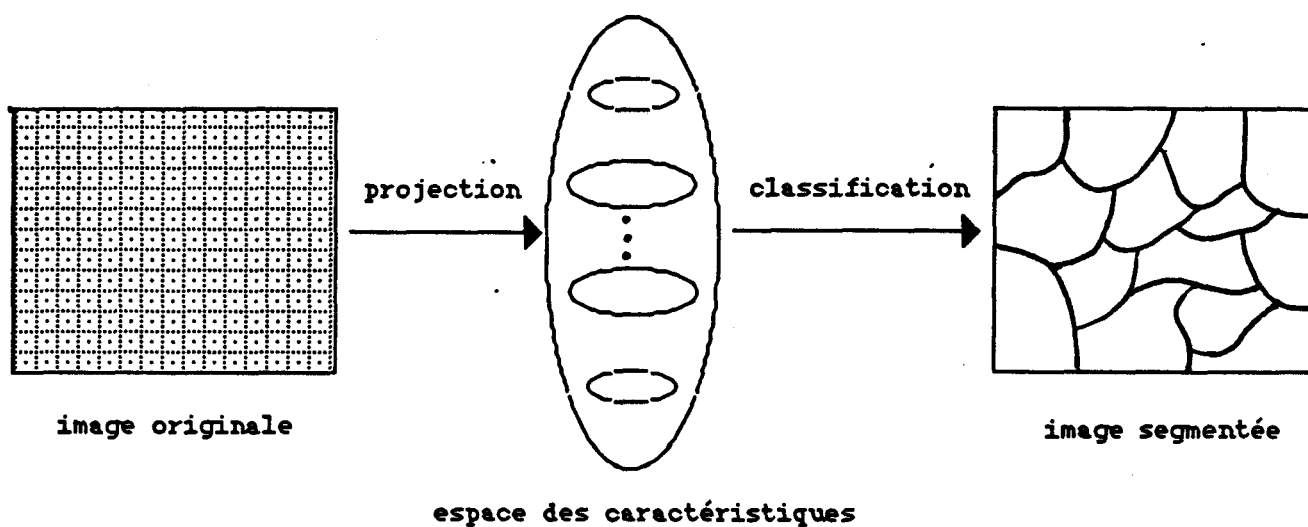
### II.3.2 - La segmentation d'images [FU 81]

La segmentation est la technique par laquelle sont regroupés l'ensemble des points d'une image numérique qui possèdent des attributs de visibilité similaire.

Par conséquent, l'opération de segmentation peut être vue comme une classification des points d'une image en un ensemble fini de classes (ou de régions) disjointes. Cette classification doit être surjective dans le sens où un point quelconque de l'image ne peut appartenir qu'à une et une seule classe. Chacune des classes étant caractérisée par certaines propriétés ou caractéristiques, représentant des concepts présents dans l'image, que l'on voudrait isoler. Par exemple, la couleur dans le cas d'une segmentation d'images en couleur ou le niveau de gris dans une fenêtre dans le cas d'une segmentation d'images noir et blanc.

On crée alors un espace vectoriel, appelé espace des caractéristiques (feature space en anglais), en assignant à chaque caractéristique extraite de l'image un vecteur de base de cet espace. La dimension de celui-ci étant alors égale au nombre de caractéristiques que l'on dispose afin de segmenter l'image.

Après projection de tous les points de l'image dans cet espace, en évaluant la valeur de chacune des caractéristiques en chaque point, on réalise l'opération de classification. Cette opération, appelée **clustering** (en anglais), consiste à regrouper les points de l'espace des caractéristiques en fonction d'un certain critère de voisinage. La figure II.8 regroupe globalement les différentes étapes de la segmentation d'une image.



**Fig. II.8 - Segmentation d'une image, considérée comme un cas particulier de classification dans un espace de caractéristiques**

On voit dès lors, que le paramètre clé de la segmentation est le choix de la dimension de l'espace des caractéristiques. Pour des raisons de simplicité on essaye de choisir ce nombre aussi petit que possible. Dans le cas unidimensionnel, on peut formaliser l'opération de segmentation par l'équation suivante :

$$S(x,y) = k \quad \text{si } T_{k-1} < g(x,y) < T_k \quad \text{pour } k = 0, \dots, m$$

où

$g(x,y)$  = valeur du niveau de gris du point  $(x,y)$

$T_0 \dots T_m$  = valeurs ordonnées des différents seuils

$m$  = nombre de classes

$S(\dots)$  = image segmentée

Dépendant de la dimension de l'espace des caractéristiques, ainsi que des propriétés choisies, on peut trouver plusieurs techniques de segmentation qu'on peut classer en trois groupes :



- **Techniques de segmentation unidimensionnelle**, telles que la segmentation par seuillage d'histogramme ou par extraction de contours
- **Techniques de segmentation séquentielle**, où le traitement effectué sur un point  $p$  de l'image, afin de décider s'il appartient ou non à une région, dépend de la classification des autres points. On peut distinguer, dans ce groupe, la segmentation par croissance de régions et la segmentation par division et rassemblement (split and merge).
- **Techniques de segmentation itérative**, qui s'apparente aux méthodes de relaxation vu sa ressemblance à cette classe de méthodes numériques itératives.

Afin de ne pas nous étendre longuement sur la segmentation d'image, étant donné l'objet de notre étude, nous avons regroupé, dans le tableau II.3, les principales caractéristiques des différentes méthodes de segmentation déjà citées.

**Tableau II.3 - Caractéristiques principales de certaines méthodes de segmentation**

Technique de segmentation	méthodologie	complexité de mise en oeuvre	inconvénients
seuillage d'histogramme,	établissement d'un histogramme à partir des valeurs des pixels de l'image.	simple	<ul style="list-style-type: none"> <li>- difficulté de choix des seuils</li> <li>- problème d'identification des pics de l'histogramme aux objets recherchés.</li> </ul>
extraction de contours	filtrage bidimensionnel passe-haut.	simple	<ul style="list-style-type: none"> <li>- sensibilité aux bruits dans l'image.</li> <li>- contours non fermés.</li> </ul>
croissance de régions	regroupement des points connexe partageant la même propriété	moyenne	<ul style="list-style-type: none"> <li>- choix de la propriété:</li> <li>- les contours ne correspondent pas forcément aux objets de l'image.</li> <li>- nécessité d'un traitement complémentaire pour les faux contours</li> </ul>

.../...

<p>division et rassemblement</p>	<p>subdivision de l'image en sous-quadrants à caractéristique homogène</p>	<p>moyenne</p>	<ul style="list-style-type: none"> <li>- nécessité de rassembler les quadrants adjacents de même propriété en régions.</li> <li>- frontière des régions en forme d'escaliers.</li> </ul>
<p>itérative</p>	<p>application des méthodes de relaxation à la segmentation d'images</p>	<p>grande</p>	<ul style="list-style-type: none"> <li>- connaissance à priori d'un modèle d'images.</li> <li>- connaissance à priori des différentes classes d'images.</li> <li>- n'est utilisée que pour un nombre restreint de classes d'images.</li> </ul>

On remarquera, dans ce tableau, que les deux premières méthodes de segmentation ne sont intéressantes que pour un nombre restreint ou distinct d'objets dans l'image. De plus, ce tableau montre que la détermination d'informations complémentaires, des différents objets constituant l'image, par segmentation s'accompagne d'un accroissement de la complexité de mise en oeuvre.

Ce résultat est intéressant et montre la limite de la représentation pixel pour l'extraction d'informations non contenues explicitement dans les valeurs des pixels, mais supportées par l'image.

### II.3.3 - Résumé

Loin de nous d'aborder tous les problèmes du traitement d'images en général et de la compression et la segmentation d'images en particulier. On peut dire néanmoins que la représentation pixel d'une image, malgré la simplicité d'utilisation et l'existence de dispositifs matériels peu coûteux adaptés à ce genre de représentation, ne répond que difficilement aux besoins actuels dans le domaine du traitement d'images.

En effet, nous avons vu dans un premier temps, en compression numérique des images, que l'information sémantique d'une image est difficilement quantifiable à partir de la représentation pixel. Ce qui a quelque peu limité les taux de compression atteints jusque-là. En second lieu, pour les techniques de segmentation, il est évident que l'extraction d'informations complémentaires non contenues de manière explicite dans les valeurs des pixels isolés, mais véhiculées par l'image, s'obtient difficilement et s'accompagne d'un accroissement de la complexité de mise en oeuvre.

On voit dès lors, les limitations de la représentation pixel d'une image pour le développement des méthodes efficaces, aptes à répondre aux besoins actuels en traitement d'images. D'où la nécessité d'avoir un autre modèle de représentation d'image aidant cette recherche d'informations (sémantiques).

Nous allons voir maintenant ce qu'il en est en synthèse d'images.

## II.4 - Le modèle pixel en synthèse d'images

Dans le paragraphe précédent, nous avons vu certains problèmes pouvant être posés par la représentation pixel en traitement d'images. Nous avons alors considéré déjà que l'image était présente et complètement digitalisée ou numérisée.

En synthèse d'images, il en va autrement. En effet, il s'agit de produire, sur un dispositif d'affichage en particulier l'écran TRC, des images ou des dessins à partir d'une structure de données calculées ou décrites. Par exemple, données produites par un programme d'application ou commande utilisateur.

Bien que d'une autre nature, l'utilisation de la représentation pixel, particulièrement au stade de la visualisation, pose un certain nombre de problèmes difficilement résolus jusqu'à présent. Prenons un exemple : la figure II.9 regroupe schématiquement, et de manière succincte, les différentes étapes de la génération d'une image dans un système de synthèse.

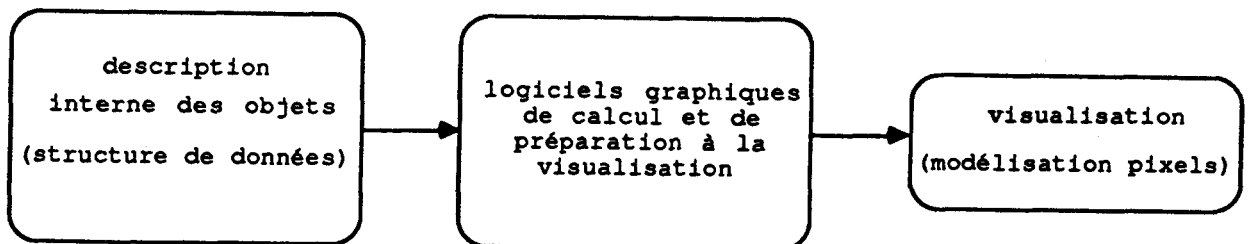


Fig. II.9 - Description globale de la génération d'une image.

Chaque objet composant l'image à visualiser, peut être modélisé dans un espace continu (voir les techniques de modélisation d'objets graphiques) [PERO 88] particulièrement pour les objets 3D. La visualisation de ces objets posera entre autre le problème de discrétisation, c'est-à-dire du passage du monde continu (description interne des objets) au monde discret de l'image (représentation pixel). Dans ce cas, il est habituel de rencontrer des problèmes d'aliassage et de disparition de certains détails de la scène, par exemple [PERO 88] [GUPT 81] [CRON 77]. Par ailleurs, d'autres problèmes de nature différente, mais qui peuvent se poser, tel l'accès rapide et la taille de la mémoire de rafraîchissement, la désignation d'objets directement de l'image (la correspondance pixel  $\rightarrow$  objet est difficile à établir), le remplissage de taches, le codage des contours "points"...

Parmi ces différents problèmes, conséquence de la représentation pixel d'une image nous avons choisi, par souci de simplicité et de brièveté, de développer les problèmes inhérents aux codages des contours et au phénomène d'aliassage. Même si d'autres problèmes, particulièrement de désignation d'objets et de remplissage de taches entre autres, mériteraient tout aussi bien quelques développements.

#### II.4.1 - Codage des contours

On distingue deux types de codage de contours :

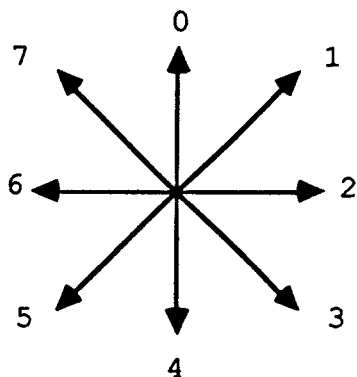
- le **codage des contours points**, représentant l'ensemble des points de la matrice pixel composant le contour
- le **codage de contours par primitives**, décrivant un contour à l'aide d'algorithmes de tracé d'éléments géométriques simples tels que segments, arcs de cercle, polygones...

Le second type de codage est difficile à cerner, dépendant des possibilités offertes par le processeur graphique utilisé. De plus, on peut considérer que ce type de codage est d'une manière générale assez compact, il n'est donc pas nécessaire d'aller plus loin. Par contre, il l'est beaucoup moins (compact) pour le premier. C'est ce que nous nous proposons de développer.

### II.4.1.1 - Codage de Freeman [FRE 61]

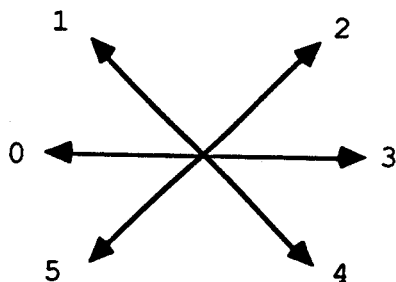
Une des méthodes exactes de description de la forme d'un contour est le code de Freeman. Dans cette méthode, les contours des objets de l'image sont supposés définis. On choisit alors pour un contour donné, un point de départ  $(x_0, y_0)$  ainsi que le sens (droite ou gauche) dans lequel on va parcourir la figure à partir de ce point. On passe alors d'un point du contour au suivant à l'aide d'un code de direction dépendant de la connectivité des points. Ainsi les directions peuvent être représentées par :

- 8 directions, pour des points de connectivité 8 (réseau rectangulaire plan)

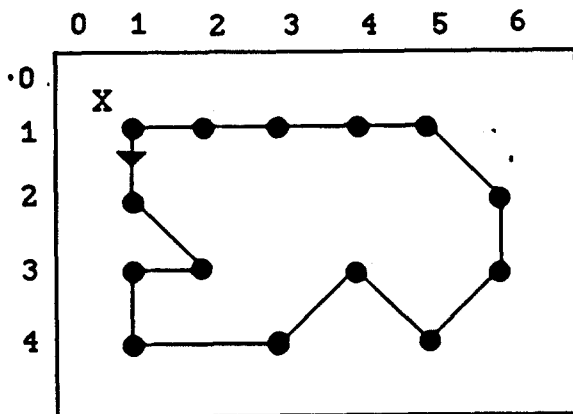


(codage à 3 bits)

- 6 directions, pour des points de connectivité 6 (réseau hexagonal plan)



La figure II.10 nous donne un exemple de codage d'un contour donné par code de Freeman, pour des points de connectivité 8



suivi du contour

X = point de départ (1,1)

C = { 4, 3, 6, 4, 2, 1, 3,  
1, 0, 7, 6, 6, 6, 6 }

code correspondant

Fig. II.10 - Codage de Freeman d'un contour points à connectivité 8.

L'équation suivante indique le nombre N de bits nécessaires pour décrire la forme géométrique d'un contour constitué de n points de connectivité k situés sur une grille cartésienne de M x M points

$$N = 2 \log_2 (M) + (n - 1) \log_2 (k)$$

Dans cette équation  $2\log_2(M)$  bits sont nécessaires pour décrire la position du premier point et  $(n - 1) \log_2 (k)$  bits sont nécessaires pour décrire la position des  $(n - 1)$  points suivants.

Même si cette description de contour permet de structurer facilement les données graphiques, de les manipuler et de les adapter à certaines opérations (translation, rotation particulières...), elle est par contre relativement coûteuse, du fait que le nombre de bits qu'elle nécessite est proportionnel au nombre de point du contour. Ceci est dû au fait qu'elle ne tient pas compte de la redondance présente dans le contour à coder.

Cette technique de codage est parfois appelée technique "externe" par le fait qu'elle est basée uniquement sur l'analyse des points frontière de la région.

La technique suivante utilise, par contre, les points intérieurs à la région considérée.



### II.4.1.2 - Codage par transformation par axe median [SAM 83] [WU 86]

Un exemple typique de méthode de codage de contours utilisant une technique dite "interne" est la transformation par axe median (Medial Axis Transform ou MAT en anglais). Dans cette approche, une région R composée de l'ensemble de ses points intérieurs  $I = \{x_1, \dots, x_n\}$ , est transformée en un ensemble de points et de valeurs associées appelé squelette. La détermination du squelette se fait de la manière suivante.

A partir de chaque point  $x_i$  de I on étend une figure géométrique simple centrée sur ce point de manière que la forme étendue obtenue soit maximale et complètement incluse dans la région. Pour des raisons de simplicité, on choisit généralement un carré comme figure géométrique, bien que d'autres formes puissent tout aussi bien être utilisées comme le cercle par exemple. L'ensemble des points  $x_i$  dont la forme étendue n'appartient ou n'est pas incluse complètement dans aucune autre forme étendue centrée sur un autre point  $x_j$ ,  $i \neq j$ , constitue alors le squelette. La figure II.11 illustre l'application de la transformation par axes medians sur une région composée de 20 points contours à l'aide de carrés.

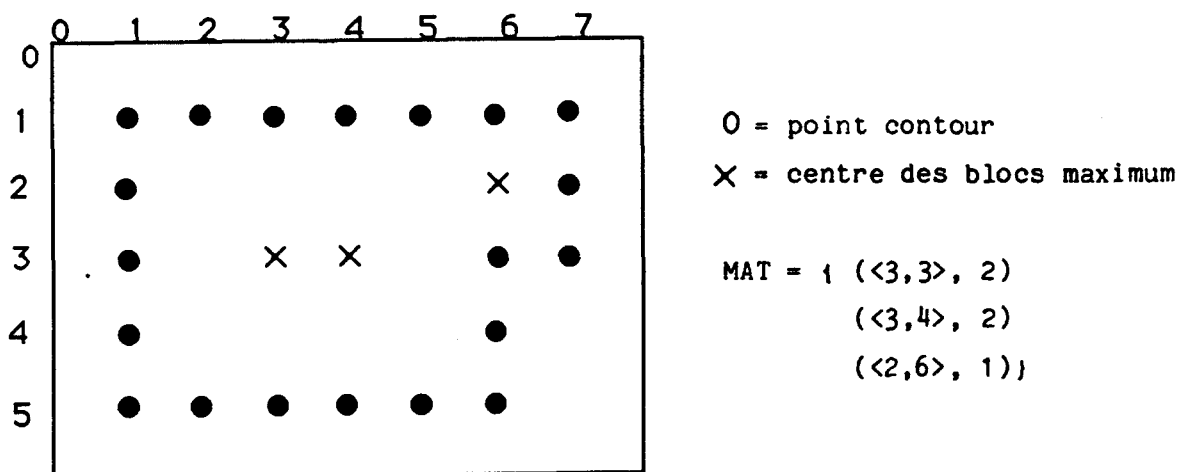


Fig II.11 - Application de la transformation par axe median

A noter que cette appellation provient du fait que tous les centres se situent en définitive à mi-chemin entre les deux bords opposés de la région.

#### II.4.2 - Le phénomène d'aliassage [CROW 77] [PITT 80] [GUPT 81] [WHIT 83]

Un obstacle majeur rencontré en synthèse d'images numériques, afin d'obtenir des images réalistes et de bonne qualité, est l'aliassage (aliasing). Ce phénomène se situe dans la transformation ou le passage du monde continu (modèles d'objets) au monde discret (modèle pixel), les objets sont décrits en continu alors que la mémoire d'images est discrète. On voit alors apparaître dans l'image des formes de marches d'escaliers irrégulières pour les frontières séparant deux zones de couleurs significativement différentes, de clignotement et/ou de disparition des petits objets, même partiellement, selon leur position sur la scène (images animées ou objets 3D). Les figures II.12 et II.13 montrent un exemple de ce phénomène d'aliassage pour chaque type de problème.

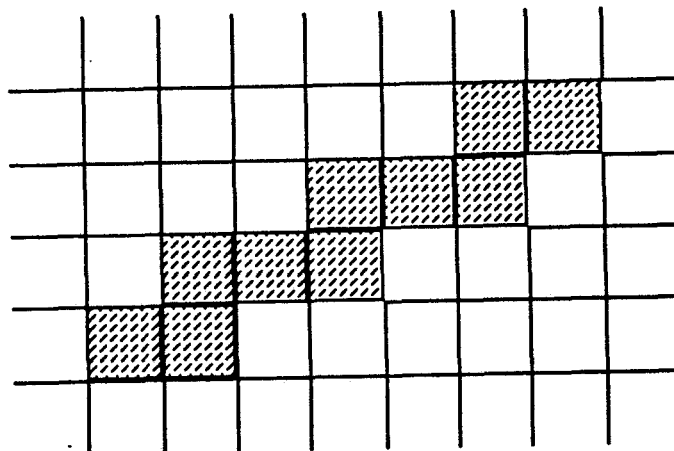


Fig. II.12 - Effets de marches d'escaliers (crénelage)

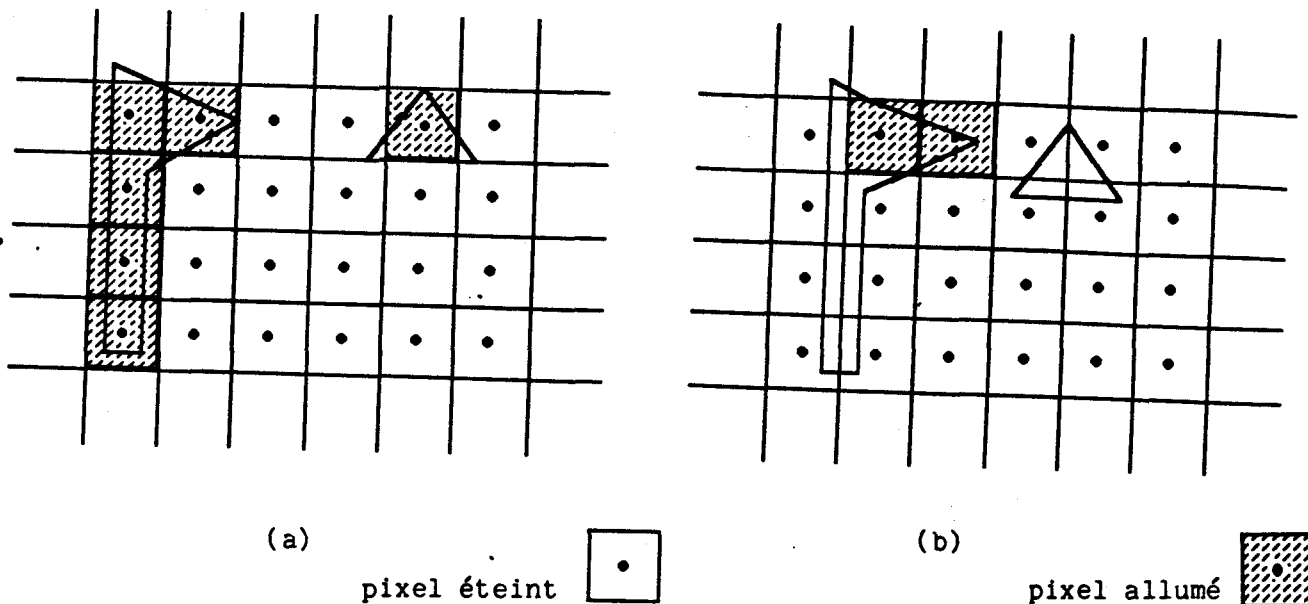


Fig. II.13 - Affichage de petits objets selon leur position dans la scène

Les principales causes de ce phénomène d'aliasage résident dans un échantillonnage inapproprié (sous-échantillonnage) de la scène à afficher, dues particulièrement à une définition insuffisante de la mémoire d'images numériques.

Une méthode globale d'antialiasage sera alors tout naturellement le sur-échantillonnage. Cette méthode est réalisée en deux étapes. Une première étape consistant à sur-échantillonner l'image à générer, par subdivision de chaque pixel en plusieurs sous-pixels de calcul. Le nombre de division d'un pixel suivant l'axe des x et l'axe des y nous donne les deux rapports de sur-échantillonnage de la scène. La seconde étape consiste à filtrer l'image sur-échantillonnée précédemment par un filtrage passe-bas antirepliement, puis à la rééchantillonner à la définition d'affichage.

Cette méthode générale est très intéressante et permet de traiter des scènes variées, quelles que soient les entités qui les composent : linéaires ou non linéaires. Néanmoins, les inconvénients majeurs de cette méthode sont l'espace mémoire très important, nécessaire pour le calcul de l'image sur-échantillonnée, et le temps et le coût de calcul de toute la scène, bien qu'il est possible, en localisant les zones à sur-échantillonner, de diminuer quelque peu ce temps de calcul.

D'autres méthodes d'antialiassage peuvent être rencontrées, telles les méthodes incrémentales d'antialiassage. Celles-ci sont locales et traitent les segments et bords de polygones en se basant sur les propriétés géométriques de ces éléments tel l'algorithme de tracé de segment de Bresenham [BRES 65]. Parmi ces méthodes, on peut citer la méthode de Pitteway et Watkinson [PITT 80] ou celle de Gupta et Sproull [GUPT 81]. Ces méthodes définissent de nouveaux paramètres de décision permettant, en plus du choix de la position d'un pixel, le calcul de son intensité suivant la contribution en intensités du polygone et du fond. On obtient alors, sur les bords du polygone ou du segment un certain dégradé qui atténue l'effet de marche d'escaliers. Cependant, certains problèmes spécifiques peuvent apparaître aux extrémités des segments ainsi qu'aux sommets des polygones.

En définitive, on voit encore que la représentation pixel, par sa faible définition de l'image, implique un certain nombre de problèmes d'antialiassage, qui restent difficiles à résoudre et demandent un temps de calcul supplémentaire afin d'aboutir à une image réaliste.

En fait, le problème ici vient du fait que le pixel est un objet élémentaire discret. Par conséquent, un nouveau modèle de représentation d'images doit être basé sur des objets élémentaires continus ou discrétisés avec antialiassage.

## II.5 CONCLUSION

Que ce soit en synthèse ou en traitement d'images, on a vu dans ce chapitre que la représentation pixel ou canonique d'une image ne répond que difficilement aux besoins actuels des utilisateurs de l'imagerie informatique. A noter que nous avons sciemment omis de parler de l'aspect architecture que nous nous réservons de développer au chapitre IV.

Néanmoins, les problèmes suivants, bien que non exhaustifs, permettent de nous éclairer sur les limites de la modélisation pixel d'une image :

- nécessité d'un dispositif de stockage, généralement important (mémoire d'image) pour l'affichage d'une image.
- difficulté d'extraction à partir du modèle pixel de l'information sémantique, définie comme étant l'information nécessaire à la reconnaissance d'une image par le système visuel humain, due à l'inexistence des relations "sémantiques" entre les pixels, mise à part leur position.
- difficulté de désignation d'objet directement de la matrice pixels (problème de correspondance pixel --> objet)
- nécessité d'un traitement complémentaire afin de diminuer les problèmes d'aliassage, qui peuvent apparaître et sont principalement dûs à l'aspect discret d'un pixel.

Il s'ensuit alors que l'objet élémentaire pixel d'une image est souvent trop limité :

- en surface
- en propriétés (position, couleur ou intensité)
- en relations avec l'application

C'est dans ce cadre que nous nous proposons d'étudier, dans le chapitre suivant, un nouveau modèle de représentation apte à répondre efficacement aux différents problèmes posés par le modèle pixel.

## **CHAPITRE III**

### **LA MODELISATION MACRO-PIXEL**

### III-1 - introduction

Dans ce chapitre, nous nous proposons d'étudier un nouveau modèle de représentation d'images apte à répondre efficacement aux besoins actuels de l'imagerie informatique, sans pour cela exclure le modèle pixel.

En effet, l'utilisation de la représentation pixel d'une image, malgré ses inconvénients (voir Chapitre II), est dictée surtout par l'existence de dispositifs matériels efficaces de visualisation et d'acquisition d'images basés sur ce modèle, à faible coût. En fait, le modèle pixel peut être considéré non pas comme un choix libre, mais plutôt comme un choix logique, imposé par la technologie finale. De ce fait, on peut assimiler le pixel pour les images à ce qu'est le bit pour la représentation interne des données, constituant ainsi une base élémentaire mais universelle pour la construction d'objets.

Dans ce cas, l'image pixels est obtenue par une traduction du modèle interne de représentation d'objets (monde continu), dans l'espace discret de visualisation (Fig. III.1).

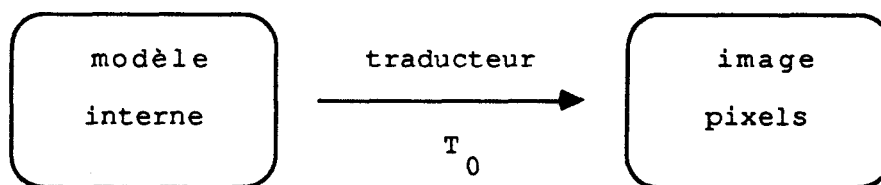


Fig.III.1 - Processus de génération d'une image pixels

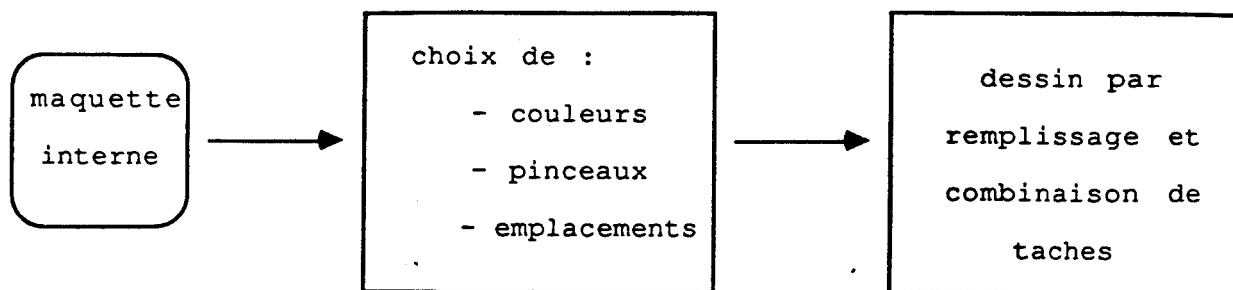
Cependant, cette traduction du monde continu en pixels, pose entre autres des problèmes :

- de détermination d'informations non contenues de manière explicite dans les attributs du pixel (intensité, position), mais supportées et véhiculées par l'image telles que contours ou textures, rendant ainsi difficile la compression numérique d'images ou l'extraction de primitives.
- de désignation d'objets directement de l'image, étant donnée la difficulté de réaliser la correspondance Pixel --> Objet.
- de relations avec l'application.
- et enfin du point de vue technologie de visualisation, de finesse de cette image, causant parfois des problèmes d'aliassage.

Ces inconvénients, liés à la représentation pixels d'une image, nous ont conduit à l'étude d'un nouveau modèle de représentation devant tenir compte de ces différents paramètres.

Pour cela prenons un exemple, simple mais révélateur du processus de génération d'une image, d'un peintre travaillant sa toile. Sans rentrer dans les détails des techniques de peinture, on peut dire qu'à partir d'une représentation visuelle interne des objets qu'il veut décrire, et pour chacun d'eux, le peintre choisit un emplacement dans la toile, l'ensemble des couleurs et le ou les types de pinceaux qu'il doit considérer. Par suite, il dessine ces objets par remplissage de certaines zones. En fait, on peut dire que le peintre travaille sa toile plutôt sous formes de taches et en combinant celles-ci il arrive à donner forme à ces objets et ainsi à leur donner une certaine signification (Fig. III.2).





**Fig.III.2 - Processus de génération d'une toile de peintre**

On en déduit dans ce cas, qu'une tache est pour le peintre l'objet élémentaire et que la production d'objets complexes se fait par combinaison de taches.

Ce processus diffère de la génération d'image en infographie, où l'aspect combinaison intelligente de taches est remplacé par l'aspect juxtaposition de points images. D'où une certaine perte d'information.

En effet dans ce cas, l'image est générée par projection des objets de la description interne (ou scène), directement dans l'espace discret de visualisation en associant à chaque objet un ensemble de "positions-points" et de couleurs associées.



Un objet est alors obtenu par juxtaposition d'un certain nombre de pixels affectés de leur attribut couleur. D'où une certaine perte et une dispersion du contenu sémantique de l'objet sur tous les pixels le composant.

D'autre part, un aspect important lié à la forme de l'objet n'apparaît pas distinctement de la représentation pixel. Cet aspect a été mis en évidence par les résultats de neurophysiologie et psychologie de la vision, qui montrent que dans la perception d'une scène un observateur humain ne traite pas de manière quantitative les luminances ou les valeurs de chaque

pixel. Il apparaît que l'observateur cherche à distinguer les formes contenues dans l'image et éventuellement à approximer mentalement les traits perçus à un groupe reconnaissable (voir annexe).

On voit, dès lors, l'importance des contours dans une image et par conséquent la nécessité d'avoir une information précise sur cet aspect.

En résumé, ces constatations nous ont amenées à considérer une image non plus sous forme de juxtaposition de pixels, mais plutôt sous forme de combinaison de taches.

Pour cela, nous allons dans le paragraphe suivant, déterminer les caractéristiques générales de ces objets-taches, définis ici comme entités élémentaires et regroupés sous le terme de "Macro-pixels", dans le sens où elles requièrent et contiennent plus d'informations que le pixel lui-même. On développera par la suite, aux paragraphes III.3 et III.4, leurs aspects géomorphologique et visuel.

La notion de combinaison de taches implique nécessairement la définition des opérateurs réalisant cette combinaison, ce qui fera l'objet du paragraphe III.5. Enfin, au paragraphe III.6, nous donnerons les différents attributs de la modélisation macro-pixels.

### **III.2 - Le modèle macro-pixels**

Nous avons vu de ce qui précède, par les faiblesses du modèle pixel, la nécessité de voir autrement une image que par juxtaposition de pixels, afin de tenir compte des différents paramètres permettant d'avoir une connaissance plus approfondie du contenu de l'image.

Nous allons essayer dans ce paragraphe de définir les caractéristiques générales de ce modèle. Pour cela, après avoir situé la description macro-pixels dans la chaîne aboutissant à l'image, nous donnerons les caractéristiques essentielles auxquelles un macro-pixel doit y répondre.

### III.2.1 - La description macro-pixel

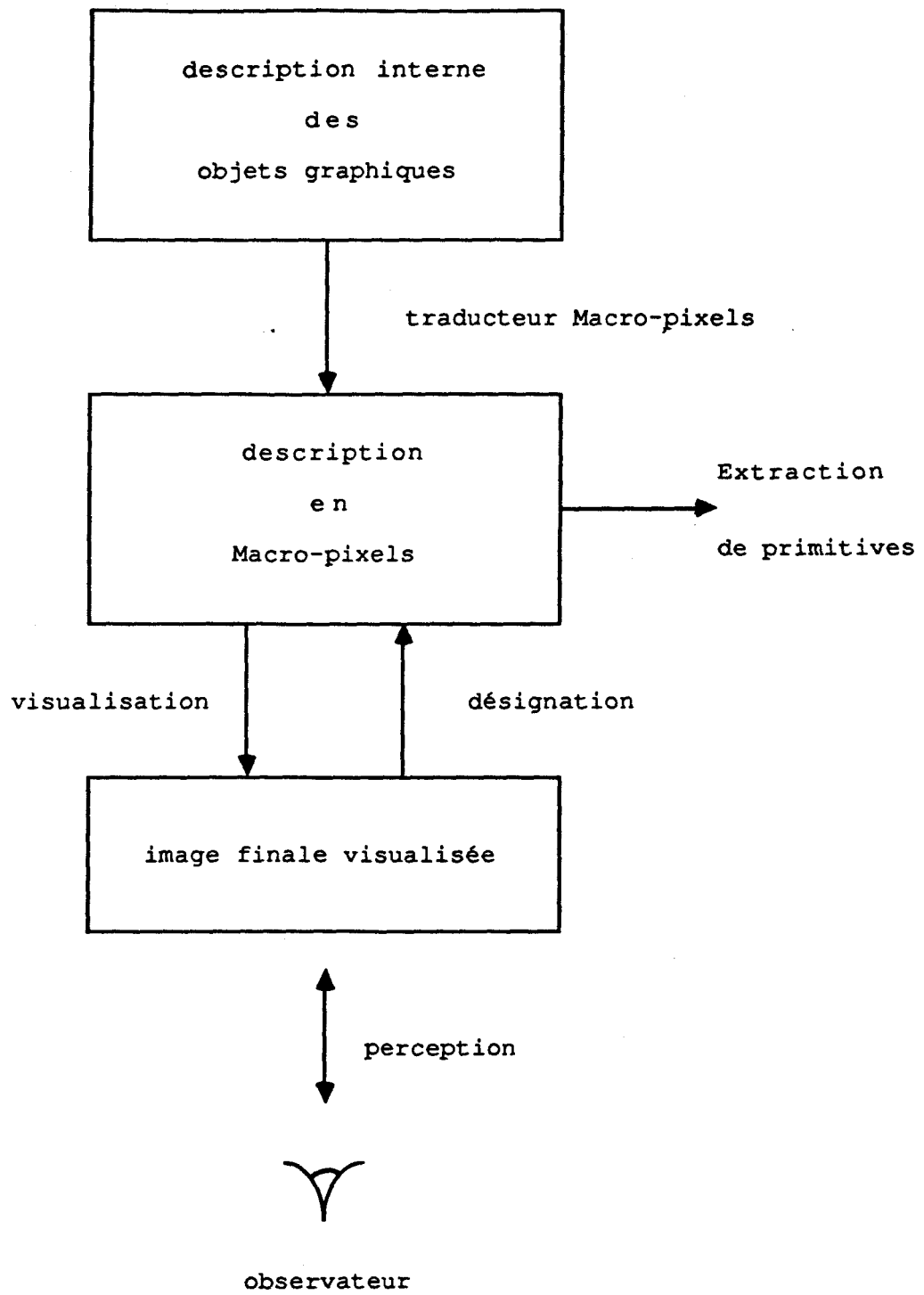
Il est intéressant, avant d'aborder la description de l'objet macro-pixel, de bien situer cet objet dans la chaîne aboutissant à la visualisation et la perception d'une image.

Nous devons conserver d'une part, le modèle pixel, qui au stade final de la visualisation reste très performant par ses possibilités à faible coût, et d'autre part, la nécessité d'une prise en compte d'un certain nombre de paramètres ou d'attributs propres à répondre efficacement aux besoins exprimés jusque là, mais non contenus de manière explicite dans la représentation pixel.

On a montré dans un premier temps que la combinaison de taches est plus adaptée pour la génération d'une image que la juxtaposition de pixels. Par ailleurs, une telle combinaison doit nous apporter une connaissance plus ou moins précise sur l'aspect contour de l'objet. Une telle information permettra entre autres :

- une extraction de primitives et une désignation plus simple des objets à peu de frais.
- des manipulations géométriques.
- des coloriages ou remplissages.
- des texturations (impossibles sur des pixels isolés).
- des transformations propres à la reconnaissance de formes.

De ce fait, la représentation des objets graphiques à l'aide de macro-pixels ne peut se faire qu'en interface entre la description interne et la visualisation finale des objets (Fig. III.3).



**Fig. II.3 - Schéma globale de visualisation d'une image à l'aide de macro-pixels.**

On voit dès lors que l'image ne représente qu'un support d'informations privilégié permettant une transmission performante entre un émetteur (dispositif de visualisation) et un récepteur (l'oeil humain).

A noter que les dispositifs de visualisation actuels affichent une image en terme de pixels ce qui impliquerait une transformation du macro-pixel au pixel. Cependant, il n'est pas exclu de voir éclore de nouveaux dispositifs permettant une visualisation d'images autrement que par des pixels.

Néanmoins, l'introduction d'une couche intermédiaire de représentation peut présenter des avantages et des inconvénients. Parmi les inconvénients on peut noter :

- une plus grande complexité du système de visualisation.
- la nécessité de définir des opérateurs nouveaux de conversion.

Cependant un tel schéma permet :

- une manipulation directe, tout en évitant le recours à la couche "objets", plus complexe.
- la possibilité de faire appel à des opérateurs cablés
- la possibilité éventuelle, si ces opérateurs fonctionnent en temps réel, de limiter la capacité de la mémoire de rafraichissement.

### **III.2.2 - Caractéristiques générales du macro-pixel**

Les inconvénients liés au modèle pixels d'une image nous ont amenés à concevoir un nouveau modèle de représentation d'images basé non plus sur le pixel, mais plutôt sur une description plus globale et macroscopique de l'image ou des objets la composant.

Ainsi, le modèle macro-pixel doit apporter une solution aux différents problèmes posés par la représentation pixel. Par conséquent, un tel modèle doit :

- être compact du point de vue codage d'image et de contour.
- faciliter l'extraction d'informations relatives aux différents objets composant l'image, et particulièrement de l'aspect contour.
- permettre une désignation rapide des objets directement de l'image.
- supporter naturellement des propriétés d'images trop "larges" pour un pixel, tels que anti-aliasage, texture, lissage...

Il vient alors que l'objet élémentaire macro-pixel qu'on veut définir doit contenir, par rapport à l'élément pixel, un certain nombre d'informations supplémentaires permettant une connaissance plus précise et approfondie de l'image.

Ces deux caractéristiques du macro-pixel, élémentarité et quantité d'informations apportée, nous amènent à le considérer, par analogie à l'élément pixel, sous trois aspects :

- **l'aspect intrinsèque**, définissant sa morphologie et son aspect visuel
- **l'aspect descriptif**, déterminant le type de composition qu'on peut réaliser avec le macro-pixel
- **l'aspect sémantique ponctuel**, définissant la quantité d'informations apportée par le macro-pixel indépendamment de l'image.

Ce choix est justifié dans le sens où un macro-pixel doit être une

entité élémentaire par son aspect intrinsèque, et doit permettre une description plus précise de l'image ou des objets la composant, ce qui impliquerait un certain apport sémantique et descriptif.

### **1. Aspect intrinsèque**

Cet aspect regroupe tout ce qui est forme géométrique et caractéristiques visuelles du macro-pixel. Dans notre cas, tout macro-pixel doit avoir des caractéristiques géométriques et visuelles suffisamment simples pour qu'on puisse effectivement le considérer comme élémentaire. Ce qui revient à dire que la génération de tout macro-pixel doit être une opération simple et facile à réaliser, ce qui implique l'existence d'un procédé ou d'un algorithme simple réalisant une telle opération. Nous étudierons plus en détail cet aspect aux paragraphes III.3 et III.4.

### **2. Aspect descriptif**

Dans le modèle pixel, tout objet graphique est obtenu par juxtaposition d'un certain nombre de pixels. Cette juxtaposition induit une certaine perte d'informations qui rend difficile des opérations telles que l'extraction de primitives ou la désignation d'objets directement de l'image, par exemple. De plus, le codage de cet objet est rendu difficile par le grand nombre de données (position, couleur) nécessaire pour le décrire complètement, sans qu'il y ait de relations réelles entre ces données.

Or, dans l'exemple du peintre, la combinaison de taches, tout en étant une opération simple et surtout concise, permettait d'éviter une trop grande perte d'informations et ainsi de décrire efficacement tout objet graphique.

De ce fait, dans le modèle macro-pixel la description de tout objet graphique doit être obtenue par combinaison de tâches. Ce qui permettra, entre autres, un codage efficace de l'objet et une extraction aisée de l'information contour en particulier, dont l'apport sémantique est non négligeable. D'autre part, la puissance de description des objets par combinaison de macro-pixels étant plus riche que la simple juxtaposition, l'apport sémantique est par conséquent plus important du fait qu'on peut spécifier les relations existantes entre ces macro-pixels.

Cependant, une telle combinaison nécessite de définir les opérateurs associés, qui seront étudiés au paragraphe III.5.

### **3. Aspect sémantique ponctuel**

On appellera information sémantique ponctuelle, l'information permettant de décrire le macro-pixel indépendamment de l'image.

Un macro-pixel, étant considéré comme une tâche, occupe un certain espace plus ou moins large dans l'image. De ce fait, on peut spécifier :

- sa taille, pouvant être variable (petite, moyenne, grande...).
- sa forme, pouvant changer d'un macro-pixel à l'autre (forme ovale, polygonale...).
- son contenu visuel (couleur ou texture).

Ce qui rend le macro-pixel, dans ses attributs, plus important que le pixel. En effet, ce dernier étant assimilé à un point dans l'image, possède peu de signification réelle. La taille et la forme sont fixes et uniques et l'intensité ou couleur est constante pour un pixel donné.

Par conséquent, l'information sémantique ponctuelle du macro-pixel doit être supérieure à celle du pixel.



### III.2.3 - Définition générale d'un pixel

A priori, nous serions tenté de définir un pixel comme étant un objet élémentaire, pouvant se confondre avec un point sur l'image. Cependant, cette définition n'étant pas complète on lui préférera l'interprétation suivante :

#### Définition

Un pixel est une zone de morphologie simple et unique (généralement rectangulaire pour des raisons techniques) pouvant se confondre à un point fixe sur l'image, ayant comme seule caractéristique visuelle une valeur de niveau de gris ou de couleur affectée ou déterminée dans le même emplacement de l'image réelle ou d'un objet graphique quelconque. Cette valeur n'a de sens qu'en cet emplacement. Dans ce cas, la description de tout objet graphique à l'aide de pixels, se fait par juxtaposition.

De cette définition, nous pouvons déduire la tableau III.1 regroupant les caractéristiques générales de l'objet pixel :

caractéristiques	
sémantiques ponctuelles	= 0
descriptifs d'objets	juxtaposition
géométriques	simples et uniques
visuelles	1 valeur de visibilité
position	fixe

Tableau II.1 - caractéristiques générales d'un pixel

### III.2.4 - Définition de l'objet macro-pixel

Nous avons vu, de l'énoncé des caractéristiques générales, les différents aspects auxquels un macro-pixel doit répondre. En fait, seuls les aspects intrinsèque et descriptif intéressent la définition du macro-pixel. Ces deux aspects impliquent ou définissent l'aspect sémantique du macro-pixel par :

- la spécification des caractéristiques géométriques et visuelles permettant ainsi de déduire l'information sémantique ponctuelle du macro-pixel.
- le type de combinaison réalisé sur les macro-pixels définissant ainsi l'apport sémantique du macro-pixel dans l'image.

En effet, pour être considéré comme élémentaire, un macro-pixel doit avoir d'abord des caractéristiques géométriques suffisamment simples pour être généré facilement (existence d'algorithme rapide de génération). De plus, un macro-pixel peut varier à l'intérieur même de cette tache, à condition que cette variation soit simple et facilement calculable. D'autre part, un macro-pixel ne possède pas, contrairement au pixel, une position fixe sur l'image, son emplacement dépend du positionnement de l'objet, il est possible alors de paramétrer cette position.

De ce fait, la définition d'un macro-pixel peut se formuler de la manière suivante :

#### **Définition :**

Un macro-pixel est une zone ou tache à géométrie simple et connue (pouvant regrouper un ensemble de points contigus dans l'image), ayant

comme caractéristique visuelle une variation simple et calculée ou calculable d'intensité, et pouvant se situer à un emplacement donné quelconque dans l'image réelle ou d'un objet graphique quelconque, la description de tout objet graphique est obtenue, dans ce cas, par combinaison de macro-pixels.

D'où le tableau suivant :

caractéristiques	
sémantiques ponctuelles	> 0
descriptifs d'objets	combinaison
géométriques	simple et connue
visuelles	variation simple calculable
position	paramétrable

**Tableau III.2 - Caractéristiques générales d'un macro-pixel**

Nous allons dans les deux paragraphes suivants développer l'aspect géomorphologique devant être simple et connu, et l'aspect visuel représentant pour le macro-pixel une variation simple et calculable d'intensité.

### III.3 - Aspect géomorphologique du macro-pixel

Nous allons essayer de déterminer, dans ce paragraphe, l'ensemble des figures géométriques connues, pouvant être considérées comme macro-pixels.

Si nous comparons les deux définitions du pixel et du macro-pixel, nous remarquerons que du point de vue caractéristiques géométriques, le pixel et le macro-pixel s'accordent sur l'aspect simple et connu de la zone ou la tache. La seule différence est que l'un est indivisible et que l'autre regroupe une zone plus ou moins large dans l'image ou d'un objet graphique.

Dans l'optique d'une représentation d'images par combinaisons de taches, il est évident qu'une tache quelconque, dans le sens de l'exemple du peintre, peut être plus ou moins compliquée et difficile à dessiner. Par conséquent, seul un nombre limité de figures géométriques peuvent servir de support aux macro-pixels.

Deux approches de ce problème peuvent être entreprises :

- une approche par décomposition des objets en éléments plus simples.
- une approche par l'étude des caractéristiques générales de la forme des objets.

En effet, du point de vue morphologie mathématique, il est connu que toute classe de figures, et particulièrement de polygones, peut recouvrir tout objet graphique complexe et réciproquement. Et le recouvrement à l'aide de pixels n'est qu'un cas particulier.

Par ailleurs, une connaissance précise de la forme globale des objets graphiques nous permettra de nous orienter dans le choix des figures géométriques.

### III.3.1 - Approche par décomposition d'objets

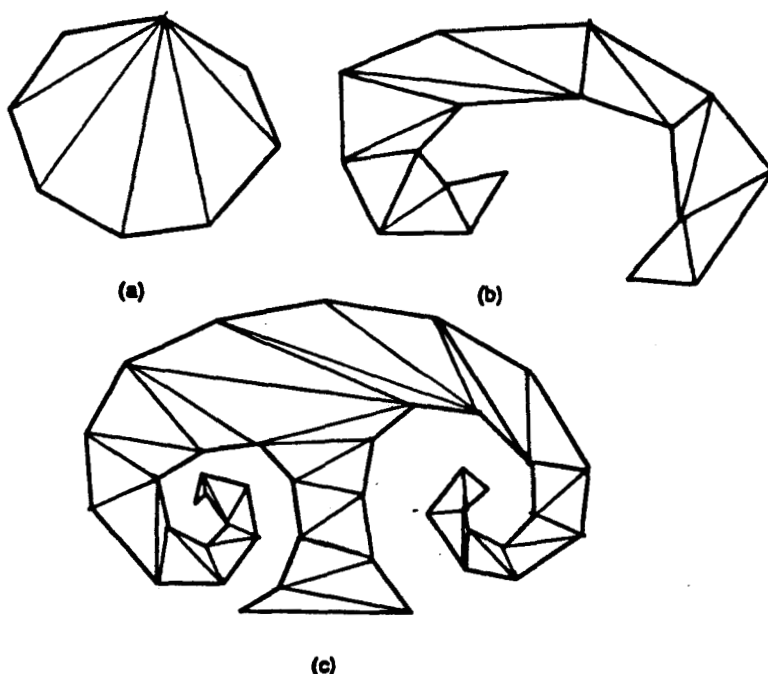
En informatique graphique un nombre important d'algorithmes ont été développés pour la décomposition d'objets - complexes - en éléments simples.

Mis à part l'intérêt théorique intrinsèque, la motivation pour de tels algorithmes est que les traitements ultérieurs appliqués à l'objet, tels que rotation, remplissage, calcul d'ombrage..., sont généralement plus faciles en utilisant les figures résultantes comparativement à un traitement global de l'objet lui-même.

Ainsi en reconnaissance des formes la décomposition d'un polygone en éléments simples permet de réduire la complexité de classification et d'analyse des objets [PAVL 78] [FENG 75]. En synthèse d'image, cela permet de résoudre certains problèmes de remplissage ou de calcul des faces cachées (par exemple [FOLE 82]).

Généralement l'objet est décrit ou approximé à l'aide d'un polygone, vu les facilités évidentes offertes par cette représentation pour les différents traitements. Dans ce cas, la décomposition se fait en triangles [WOO 85] [HERT 83], en trapèzes [FOUR 84] [CHAZ 84] et même en certains polygones particuliers tels que polygones convexes [SCHA 78] ou polygones en spirale [FENG 75]. Les triangles en particulier, sont bien adaptés aux problèmes d'interpolation de fonctions bidimensionnelles et les algorithmes de Gouraud/Phong en sont des exemples spécifiques.

Pour notre cas nous avons présenté [OULD 88] un algorithme de triangulation de polygones simples quelconques utilisant certaines propriétés des chaînes de sommets concaves et basé sur la recherche et la triangulation de polygones convexes, spirales ou en T (fig. III.4) déterminés à partir du polygone initial. Cet algorithme sera développé en annexe.



**Fig. III.4 : Exemple de polygones : (a) convexe, (b) spirale, (c) en T et leur triangularisation.**

La complexité temporelle de la plupart de ces algorithmes est de  $O(n \log n)$ , où  $n$  représente le nombre de sommets du polygone. Cette complexité devient égale à  $O(n)$  pour des polygones particuliers tels que polygones convexes, spirales [OULD 88] ou à points visibles [WOO 85].

Cependant le problème essentiel pour de tels algorithmes est de trouver l'ensemble minimum d'éléments décomposant l'objet considéré. De plus la plupart des auteurs ont élaboré des algorithmes dont l'approche est de type contour. La décomposition de l'objet se fait en utilisant des points spéci-

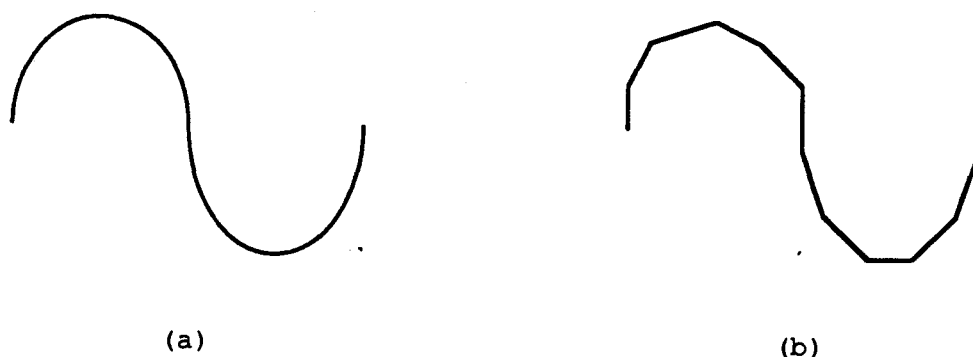
figues (sommets du polygone) ou calculés du contour. De ce fait les caractéristiques visuelles de l'objet n'entrent pas en compte. Par ailleurs, notons que la décomposition de figures ou d'objets concaves ou présentant des trous est difficilement réalisable.

Néanmoins, il est important de remarquer que tous les algorithmes permettent une décomposition d'objets sans possibilité de recouvrement entre les figures de base. En effet une telle option, tout en étant compact (possibilité de réduction du nombre de figures de base obtenues après décomposition) est plus complexe à mettre en oeuvre, et nécessite la prise en compte de l'aspect visuel de l'objet considéré.

### **III.3.2 - Approche par courbure de contour**

Afin de distinguer ou de différencier les différentes classes de macro-pixels de base, il est intéressant de regarder de plus près l'aspect global caractérisant le contour, en l'occurrence la courbure de celui-ci.

En effet, l'approximation d'un contour seulement par des segments de droites, de manière par exemple à obtenir un polygone pour faciliter la décomposition, devient très onéreuse si l'on veut conserver une certaine fidélité de l'objet dans l'image (Fig. III.5).



**Fig III.5 - une partie d'un contour d'objet (a) et son approximation par segments de droite (b).**

Le problème vient du fait que certaines parties du contour sont fortement courbés. De ce fait, il convient de distinguer pour le contour, ou pour des portions de celui-ci, les deux catégories suivantes :

- contours faiblement courbés
- contours fortement courbés

Dans ce cas, la première catégorie peut être alors approximée, à un degré de fidélité très fin, à des segments de droites. Par contre, la seconde catégorie s'apparente mieux à des arcs de cercles ou des portions d'ellipses.

A remarquer que de telles caractéristiques du contour ont été déjà considérées par [WIL 83] en élaborant une méthode de codage d'image par prédiction anisotropique non stationnaire. Dans ce cas, les contours de toute direction possible sont définis comme étant les non-stationnarités dans une image et peuvent être soit isotropiques, soit anisotropiques dépendant de la courbure, faible ou droite, du contour. Le second exemple a été réalisé par [ALAM 86] où les contours des objets ont été approximé à des arcs de cercle.



Par ailleurs, il est intéressant de noter que certains objets présentent lors de la perception visuelle des contours aux bords flous. Ceci vient des effets de perspective particulièrement pour les objets 3D. Il serait alors intéressant de prendre en compte cet aspect du contour lors de l'étude de l'aspect visuel des macro-pixels.

### III.3.3 - Les classes proposées

Il vient de l'approche par décomposition d'objets, que la plupart des algorithmes n'utilisent qu'une seule figure de base. De plus, les objets étant tous de type polygonale, la décomposition de certaines figures peut aboutir à un nombre relativement important de figures de base, telle la décomposition d'un cercle à l'aide de triangles.

D'autre part, ce qui est important à noter, la décomposition des objets se fait généralement sans possibilité de recouvrement entre ces figures de base, ce qui implique souvent une augmentation du nombre de ces figures (Fig. III.6).

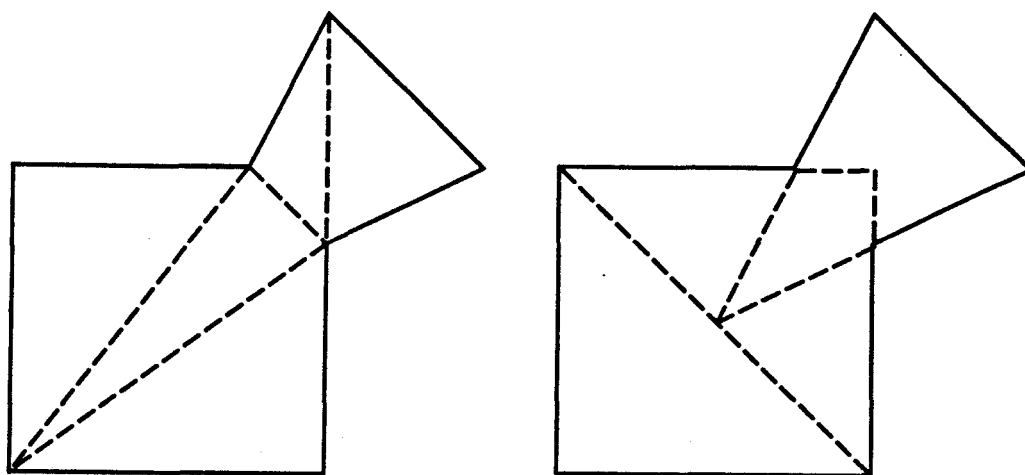


Fig. III.6 : Exemple de triangularisation d'un polygone

(a) sans recouvrement

(b) avec recouvrement

### **Remarque**

Dans la plupart des algorithmes de décomposition, le non recouvrement entre les figures de base permet une simplicité de mise en oeuvre. Cependant une telle option est dans certains cas peu compacte, aboutissant à plus de figures de base, et nécessite un traitement complémentaire lors de la visualisation, afin d'atténuer les différences d'intensité entre les différentes figures de base (lissage, modèle par facettes).

Par contre, la solution avec recouvrement est plus compacte et supporte mieux les opérations de lissage ou d'antialiasage, par exemple. Néanmoins, l'inconvénient majeur du recouvrement est d'imposer la définition d'une loi de composition entre les figures de base, ce qui rend l'algorithme de décomposition plus complexe du fait des différentes possibilités de recouvrement (transparence, composition...).

Il est évident d'après les remarques précédentes, qu'un seul macro-pixel de base ne suffit pas, même si certaines figures de base pour de tels algorithmes (par exemple triangle) peuvent être prises pour des macro-pixels. Par conséquent, il faudrait alors distinguer plusieurs classes de macro-pixels.

Cette dernière constatation, nous amène naturellement à distinguer deux classes de macro-pixels dépendant de leur courbure :

- macro-pixels à contour faiblement courbé
- macro-pixels à contour fortement courbé.

#### **III.3.3.1 - Le choix**

Parmi les différentes figures géométriques simples, connues et surtout facilement visualisables, nous avons choisi :

1 - pour les macro-pixels de la première catégorie

- le triangle
- le rectangle

2 - pour les macro-pixels de la seconde catégorie

- le cercle
- l'ellipse

### III.3.3.2 - Critères de sélection

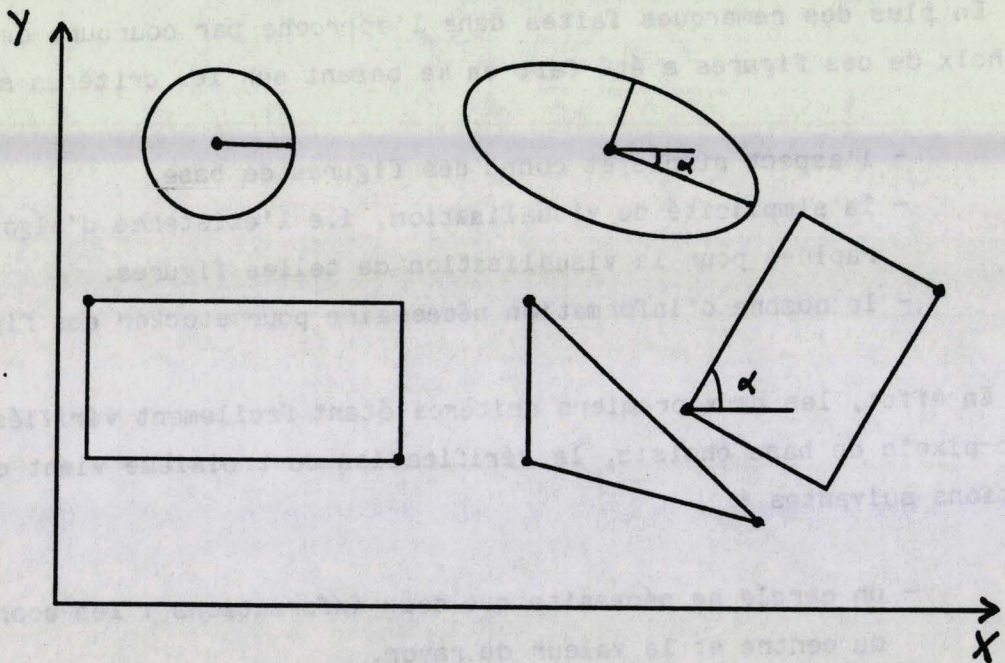
En plus des remarques faites dans l'approche par courbure de contour, le choix de ces figures a été fait en se basant sur les critères suivants :

- l'aspect simple et connu des figures de base.
- la simplicité de visualisation, i.e l'existence d'algorithmes rapides pour la visualisation de telles figures.
- le nombre d'information nécessaire pour stocker ces figures.

En effet, les deux premiers critères étant facilement vérifiés par les macro-pixels de base choisis, la vérification du troisième vient des constatations suivantes :

- un cercle ne nécessite que deux informations : les coordonnées du centre et la valeur du rayon.
- une ellipse nécessite quatre informations : les coordonnées du centre et les valeurs du grand et du petit axe, ainsi que la valeur de l'angle d'orientation (Fig. III.7).

- un triangle ne demande, pour être stocké, que trois informations : les coordonnées des trois sommets.
- un rectangle nécessite quant à lui trois informations : les coordonnées de deux sommets diamétralement opposés et la valeur de l'angle d'orientation, permettant de situer le rectangle par rapport à l'horizontale (Fig. III.7).



**Fig. III.7 - Un exemple de macro-pixels de base et leurs points caractéristiques**

## **Remarque**

La distinction faite entre une ellipse et un cercle (qui n'est qu'un cas particulier d'ellipse) d'une part, et un triangle et un rectangle (qui peut être obtenu par juxtaposition de deux triangles) d'autre part, tient à notre souci d'élimination des redondances d'informations nécessaires au stockage et à la visualisation des figures.

Les caractéristiques géométriques des macro-pixels étant définies, nous allons aborder maintenant le second volet de la définition du macro-pixel, par l'étude de l'aspect visuel de ce dernier.

### **III.4 - Aspect visuel du macro-pixel**

Dans la définition du macro-pixel l'aspect visuel a été défini comme une variation simple et calculable d'intensité lumineuse. Ce choix a été fait de manière à ce que le macro-pixel puisse d'abord constituer un objet élémentaire dans le sens d'une simplicité de ces attributs ; ensuite à prendre en compte les différentes possibilités que peut avoir l'aspect visuel d'un objet.

En effet, l'hypothèse la plus simple étant de considérer qu'un macro-pixel ne possède qu'un seul niveau de gris, cependant il serait alors difficile dans ce cas de représenter toutes les variantes d'aspect que peut prendre un objet (texture de l'objet).

Dans la représentation pixel, cet aspect de texture n'est pas pris en compte, dans le sens où à chaque pixel est associé une couleur indépendamment des autres points de l'objet. Cette couleur est généralement codée suivant les trois primaires Rouge, Vert, Bleu qui simulent efficacement la lumière émise par les objets, dont le spectre est continu sur tout le domaine des longueurs d'ondes visibles par l'oeil humain. La texture de l'ob-

jet est alors obtenue par diverses techniques (par exemple calcul d'ombrage) [OLEJ 84] souvent coûteuses en temps de calcul et en espace pour avoir l'aspect visuel désiré.

Cependant, en remarquant que localement l'intensité lumineuse (d'une des parties de l'objet) varie très lentement, il est alors possible de coder efficacement cette variation lente d'intensité.

Pour cela, après un exposé succinct des propriétés de la lumière, nous donnons pour notre cas, le type de codage choisi pour les macro-pixels.

### III.4.1 - Propriétés de la lumière

L'énergie lumineuse arrivant à l'observateur à partir d'un point donné d'un objet ou de la scène réelle est due à l'apport d'énergie provenant :

- 1 - **D'un éclairage diffus ( $E_d$ ) ou lumière ambiante** : chaque point reçoit de l'énergie lumineuse de manière uniforme suivant toutes les directions, ayant une intensité  $I_{d\lambda}$  déterminée par :

$$I_{d\lambda} = R_\lambda * I_d$$

où  $I_d$  désigne l'intensité de la lumière ambiante

$R_\lambda$  désigne la réflectance de la surface rencontrée

- 2 - **De sources ponctuelles** : dans ce cas, avant d'atteindre l'oeil de l'observateur, un rayon lumineux issu d'une source ponctuelle peut rencontrer :

a) une surface mate, il est alors réfléchi dans toutes les directions.

C'est la "réflexion diffuse". Cette réflexion suit la loi de

**Lambert** :

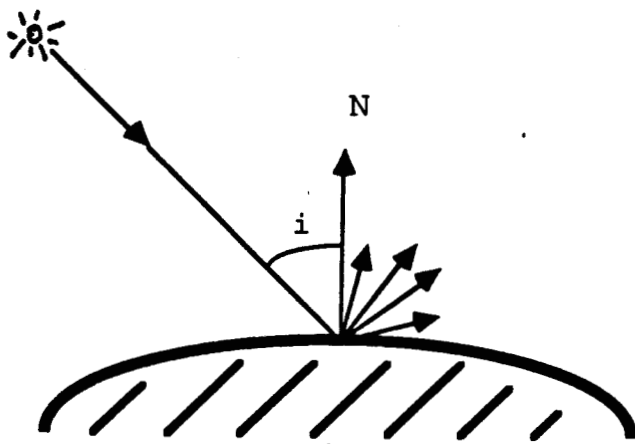
$$I_{\lambda r} = R_\lambda * \text{Max}(0, \cos i) * I_{\lambda p}$$

où

$I_{\lambda p}$  est l'intensité du rayon incident

$R_\lambda$  la réflectance

$i$  l'angle d'incidence.



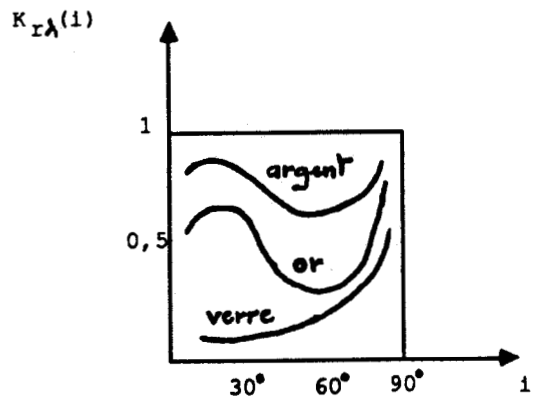
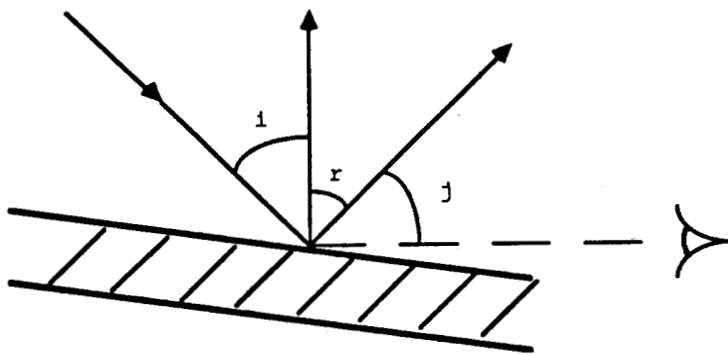
b) une surface polie parfaitement lisse. Il s'agit du phénomène de la réflexion spéculaire. Dans ce cas la réflexion suit la loi de **Descartes** et la lumière est réfléchié avec un angle  $r$  égal à l'angle d'incidence  $i$  :

$$I_{\lambda s} = k_{r\lambda}(i) * \text{Max}(0, f(j)) * I_{\lambda p}$$

avec

$f(j) = \cos^n(j)$  : dans le cas du modèle d'éclairément de **Phong**  
[PHONG 75]

$k_{r\lambda}(i)$  : étant le coefficient de réflexion de la surface, il dépend de l'angle d'incidence et de la matière de l'objet.



**Remarque :**

Les surfaces des solides réels n'étant ni parfaitement lisses ni parfaitement mates, l'intensité  $I_{\lambda s}$  provenant d'une source ponctuelle s'exprime comme une combinaison des deux cas précédents :

$$\begin{aligned} I_{\lambda s} &= I_{\lambda r} + I_{\lambda s} \\ &= (R_{\lambda} * \text{Max}(0, \cos i) + k_{r\lambda}(i) * \text{Max}(0, f(j))) I_{\lambda p} \end{aligned}$$

**3 - Par transmission** en traversant un objet non opaque. C'est la réfraction du rayon incident dispersé autour d'une direction privilégiée.

L'éclairement en un point donné de la surface de l'objet, se calcule comme la somme de toutes les intensités produites par un éclairage ambiant et par n sources ponctuelles éclairant directement ou non le point en question.

#### **III.4.2 - Le modèle proposé**

En appliquant brutalement le modèle décrit précédemment, il faudrait pour synthétiser une image considérer tous les rayons émis et ré-émis suivant les modèles de diffusion, réflexion spéculaire et transmission. Ainsi pour une image de 1000 pixels sur 1000 pixels on estime que l'on obtiendrait  $27 \cdot 10^8$  rayons différents. Ce qui est impossible à réaliser.

Cependant certains travaux ont permis de simuler ces effets et ainsi de diminuer sensiblement la complexité du modèle de la lumière, tel le lancé de rayon ou le calcul d'ombrage de **Gouraud** et **Phong**.



Dans cet ordre d'idée, une expérience intéressante réalisée dans la machine Pixel-Plane de [FUCH 81] et dans Pixel-Power [GOLD 86] où la couleur, en quantité de rouge, de vert et de bleu, est la valeur de trois fonctions  $F(x,y)$ , respectivement linéaires en  $x$  et  $y$  (Pixel-Planes) et quadratiques (Pixel-Power) de la forme :

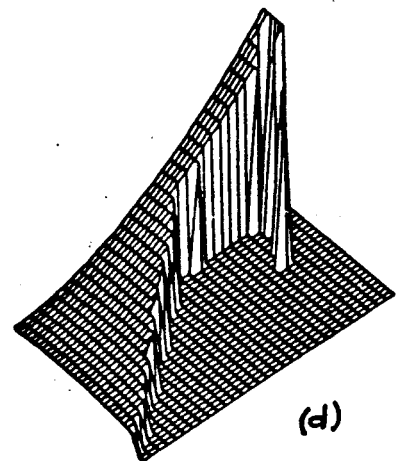
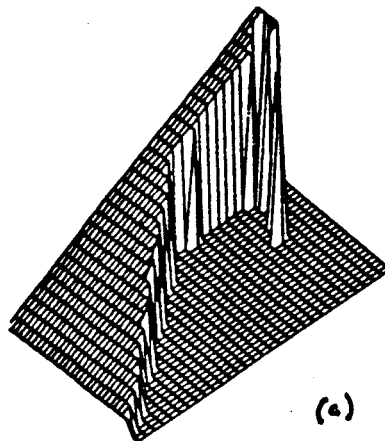
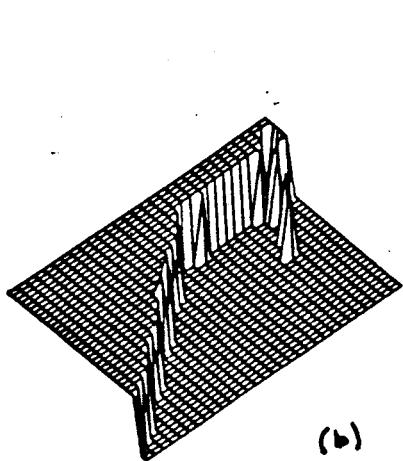
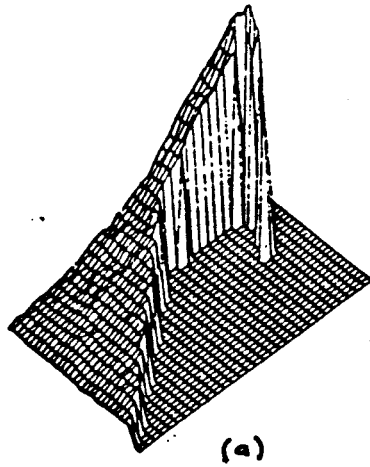
$$F(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$$

D'autre part, [KOCH 83] en réalisant une méthode de compression d'image numérique basée sur le modèle contour-texture, modélise la partie déterministe de la texture d'une région (partie restante après déduction des fortes variations de granularité) de l'image par un polynôme du second degré en  $x$  et en  $y$ . Un exemple d'approximation de la texture d'une région donnée par des polynômes d'ordre 0,1 et 2 est donné par la figure III.8.

On voit dès lors, que l'erreur d'approximation entre la région initiale et son approximation du second degré est très faible. Il y a aussi un effet de lissage de cette région.

Ces dernières expériences, nous ont amenées à modéliser l'aspect visuel d'un macro-pixel par un polynôme du second degré. En effet, un tel codage réalise un bon compromis entre les contraintes d'élémentarité d'un macro-pixel et la prise en compte des différentes variantes d'aspect d'un objet. De plus l'évolution lente et continue des fonctions polynômiales répond bien du fait que localement l'intensité varie peu dans l'image.

Cependant certaines régions de l'objet peuvent avoir une forte variation d'intensité (micro-texture). Dans ce cas, une solution sera alors de superposer ou de rajouter du bruit aléatoire, pseudo-gaussien par exemple, contrôlé par la puissance voulue (amplitude du bruit).



**Fig. III.8 - vues tridimensionnelles de l'approximation polynômiale d'une région**

**(a) originale, (b) ordre 0, (c) ordre 1, (d) ordre 2.**

### III.4.3 - Adéquation des classes de macro-pixels au modèle visuel

L'aspect visuel des macro-pixels est, tel que décrit, indépendant des limites d'expansion du macro-pixel. Ainsi, on dira qu'un macro-pixel possède une valeur visuelle définie aux environs de celui-ci, ce qui n'était pas le cas dans le modèle pixel, où la valeur visuelle n'avait de signification qu'à l'emplacement du pixel. Or, en généralisant, on peut définir la forme d'un macro-pixel par l'existence des points situés à l'intérieur d'un contour fermé et pour lesquels une valeur est calculable. Le contour est donc la limite au-delà de laquelle les valeurs du macro-pixel ne sont pas définies. Cette absence de définition peut s'assimiler à une valeur nulle. Il en résulte que la fonction de génération tend asymptotiquement vers 0 en s'approchant des bords. Par conséquent, les bords du macro-pixel peuvent être flous, ce qui les différencie du modèle par facettes [COTT 85] [NEW 79].

D'autre part, les différentes classes de macro-pixels possèdent un aspect géométrique simple et surtout convexe, ce qui permet alors un remplissage facile de ces figures de base à partir d'un point intérieur aux surfaces des macro-pixels, le centre de gravité par exemple.

Par ailleurs, ce modèle visuel doit être pris en compte par les opérateurs de combinaison des macro-pixels, dans leurs aspects géométriques (par recouvrement ou par juxtaposition) et visuels. A noter dans le cas d'une juxtaposition que les problèmes de différence d'intensité au niveau des contours sont atténués sinon éliminés si les deux macro-pixels possèdent la même fonction visuelle.

### III.5 - Les opérateurs de combinaison

Dans les deux paragraphes précédents, nous avons défini les caractéristiques géométriques et visuelles d'un macro-pixel élémentaire déterminant son aspect intrinsèque, et par voie de conséquence son aspect sémantique ponctuel. Nous avons aussi distingué quatre classes de macro-pixels :

- la classe des macro-pixels triangles.
- la classe des macro-pixels rectangles.
- la classe des macro-pixels cercles.
- la classe des macro-pixels ellipses.

Définissant ainsi quatre classes d'équivalences à partir d'éléments géométriques : triangle, rectangle, cercle, ellipse. Sans oublier la dimension visuelle du macro-pixel. A l'aide de ces quatre classes de macro-pixels, nous devons alors obtenir tout objet graphique.

On a vu dans les caractéristiques générales d'un macro-pixel (III.2.2, aspect descriptif) que la description d'un objet quelconque doit être une opération simple, concise et surtout obtenue par combinaison de macro-pixels de base avec possibilité de recouvrement entre ces macro-pixels. Une telle combinaison requiert de déterminer les lois de composition entre les macro-pixels. D'où la nécessité de définir les opérateurs associés.

Or, un objet graphique quelconque est complètement défini par son contour et son aspect visuel. Par conséquent deux types d'opérateurs doivent être pris en compte :

- les opérateurs associés à l'aspect contour de l'objet
- les opérateurs associés à l'aspect texture de l'objet

### III.5.1 - Les opérateurs de contours

Dans le modèle pixel, l'opérateur implicite de description d'objets est la juxtaposition de pixels. Pour les macro-pixels, le fait d'autoriser le recouvrement implique la définition d'autres opérateurs. Par conséquent, la description de tout objet doit tenir compte des contraintes suivantes :

- simplicité et concision.
- autoriser le recouvrement entre les macro-pixels.
- prise en compte de l'aspect visuel résultant.

Afin de pouvoir parler sans ambiguïté des macro-pixels, nous allons définir deux espaces affines euclidiens bidimensionnels :

- un espace abstrait, que nous appellerons espace propre macro-pixel, où chaque macro-pixel sera décrit entièrement.
- un espace objet où seront décrit les objets obtenus par combinaison de macro-pixels.

Cependant, l'emplacement ou le positionnement libre du macro-pixel dans l'image, nécessite de définir un point de référence à partir duquel sera positionné tout macro-pixel. Ce point de référence doit être choisi de

manière à simplifier la mise en oeuvre de la fonction de génération de valeur (espace propre) et de la composition (espace image).

Par conséquent, pour décrire un objet dans son espace, on doit disposer les macro-pixels à partir de leurs points de référence, de manière à ce qu'ils puissent s'intégrer dans l'objet et décrire complètement sa forme. Il serait alors préférable de choisir ce point de référence à l'intérieur de la surface du macro-pixel.

Etant donné qu'un macro-pixel est une tache occupant un certain espace géométrique dans l'image, cela nous amène à utiliser des opérateurs de type ensembliste tels que :

- union (U)
- différence ( $\setminus$ )
- intersection ( $\cap$ )

Regardons maintenant comment seront effectuées ces opérations sur les macro-pixels. A ce stade, on supposera que chaque macro-pixel est complètement défini dans son espace propre par :

- son identificateur
- son type : triangle, rectangle, cercle ou ellipse
- la valeur de ses attributs de position (coordonnées des sommets, centre, rayon, angle d'orientation...)
- son point de référence pris à l'intérieur du contour pour faciliter le remplissage (génération des valeurs visuelles) et la composition.

Pour cela, on procèdera de la manière suivante : on donnera d'abord une définition générale de l'opérateur, nous définirons ensuite les paramètres nécessaires à une telle opération et enfin nous donnerons un exemple d'application.

### III.5.1.1 - L'opérateur d'union

#### a - Définition

L'union de deux sous-ensembles X et Y de l'espace objet est un sous-ensemble Z du même espace regroupant tous les éléments de X et de Y.

**Remarque :** Le terme sous-ensemble peut désigner aussi bien un macro-pixel de base qu'un objet quelconque obtenu par combinaison de macro-pixels.

#### b - Paramètres nécessaires

Pour les macro-pixels, l'opérateur d'union regroupe la zone géométrique occupée par deux macro-pixels (pour une opération de base). Dans ce cas, le passage de l'espace propre à l'espace objet nécessite de spécifier l'emplacement de chaque macro-pixel dans l'espace d'arrivée en donnant des valeurs (coordonnées) à chaque point de référence. Le résultat de cet opérateur d'union sera un objet dont le point de référence sera choisi de manière à simplifier la manipulation de cet objet.

Ainsi, les paramètres d'entrée de l'opérateur d'union (U) de deux sous-ensemble A et B seront alors :

Id<sub>A</sub> : identificateur du sous-ensemble A (pointeur vers la structure de données de A)

Id<sub>B</sub> : identificateur du sous-ensemble B (pointeur vers la structure de données de B)

Pref<sub>A</sub> : valeur du point de référence de A dans l'espace objet

Pref<sub>B</sub> : valeur du point de référence de B dans l'espace objet

Les paramètres de sortie seront alors :

$Id_{AUB}$  : identificateur de l'objet résultant

$Pref_{AUB}$  : point de référence de l'objet ( $A \cup B$ )

avec

$Id_{AUB} = Union (Id_A, Id_B, pref_A, pref_B)$

### c - Exemple

A désigne un macro-pixel cercle

B désigne un macro-pixel rectangle

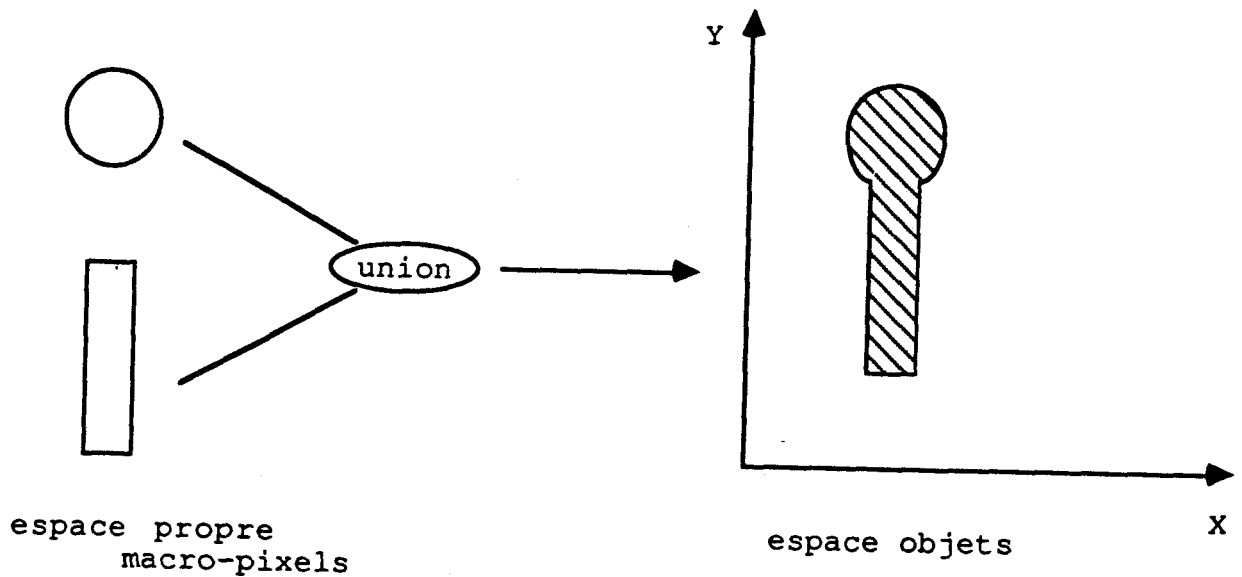


Fig III.9 - un exemple d'union de deux macro-pixels



### III.5.1.2 - L'opérateur de différence

En utilisant seulement l'opérateur d'union, il est possible parfois de dépasser le contour de l'objet qu'on veut décrire, de plus certains objets ne se prêtent pas facilement à l'union de deux ou même plusieurs sous-ensembles, telle la description d'un croissant de lune. Par conséquent, il est nécessaire de définir un opérateur permettant de supprimer une partie d'un objet.

#### a - Définition

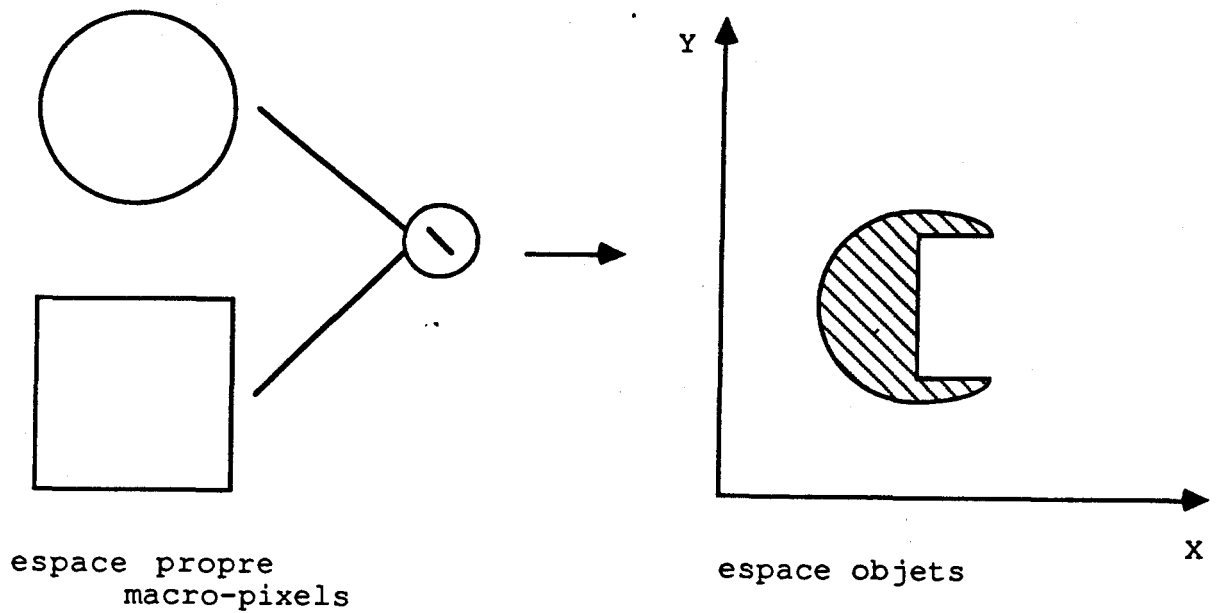
La différence de deux sous-ensembles X et Y dans l'espace objet est un sous-ensemble du même espace regroupant tous les éléments de X qui n'appartiennent pas à Y.

#### b - Paramètres nécessaires

De la même manière que pour l'opérateur d'union, on doit spécifier les valeurs des points de référence des sous-ensembles considérés. Le résultat de cette opération est un objet ayant comme point de référence un point choisi à l'intérieur de la surface résultante, afin de simplifier sa manipulation ainsi que la génération des valeurs d'aspect. A noter qu'une telle opération n'est pas commutative.

En utilisant les notations précédentes, on aura alors :

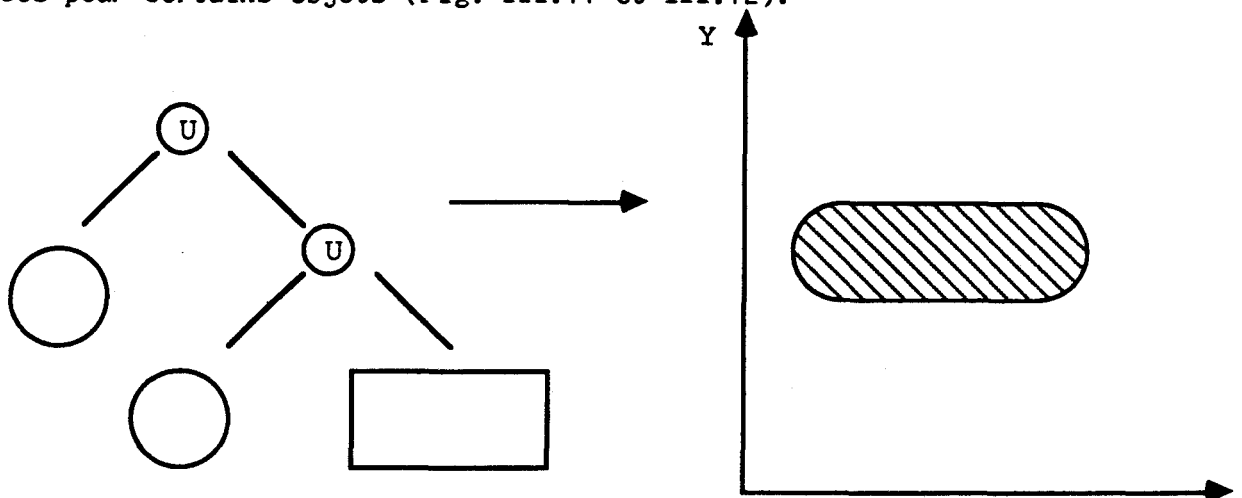
$$Id_{AB} = \text{différence} (Id_A, Id_B, \text{pref}_A, \text{pref}_B)$$



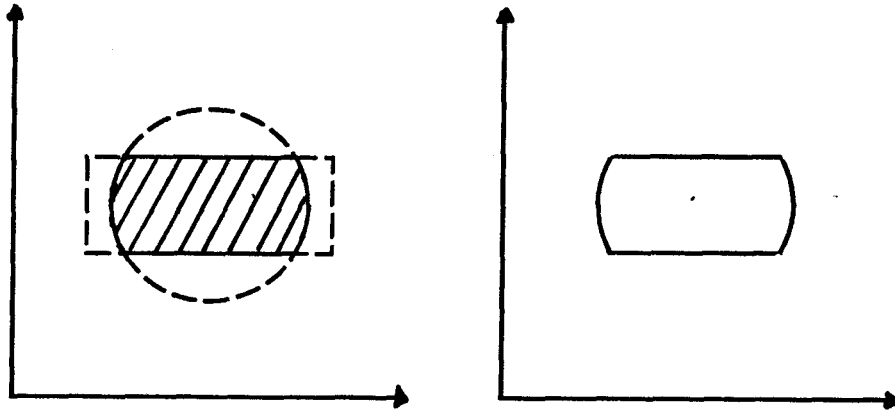
**Fig. III.10 - Un exemple de l'opération de différence**

### III.5.1.3 - L'opérateur d'intersection

A l'aide des deux opérateurs précédents on peut déjà décrire un nombre important d'objets graphiques. Cependant, certaines descriptions peuvent être longues, et par suite peu concises. Pour cette raison nous incluons l'opérateur d'intersection entre deux sous-ensembles. En effet, un tel opérateur permettra de réduire quelque peu le nombre d'opérations effectuées pour certains objets (Fig. III.11 et III.12).



**Fig. III.11 - Description d'un objet à l'aide de l'opérateur d'union**



**Fig. III.12 - Utilisation de l'intersection**

**a - Définition**

L'intersection de deux sous-ensembles X et Y de l'espace objet est un sous-ensemble du même espace regroupant tous les éléments de X qui appartiennent aussi à Y.

**b - Paramètres nécessaires**

On retrouve ici les mêmes nécessités que précédemment. Le résultat de l'opération d'intersection étant aussi un objet dont le point de référence sera choisi à l'intérieur de la surface d'intersection.

On a alors :

$$Id_{A \cap B} = \text{Intersection} (Id_A, Id_B, \text{pref}_A, \text{pref}_B)$$

c - Exemple

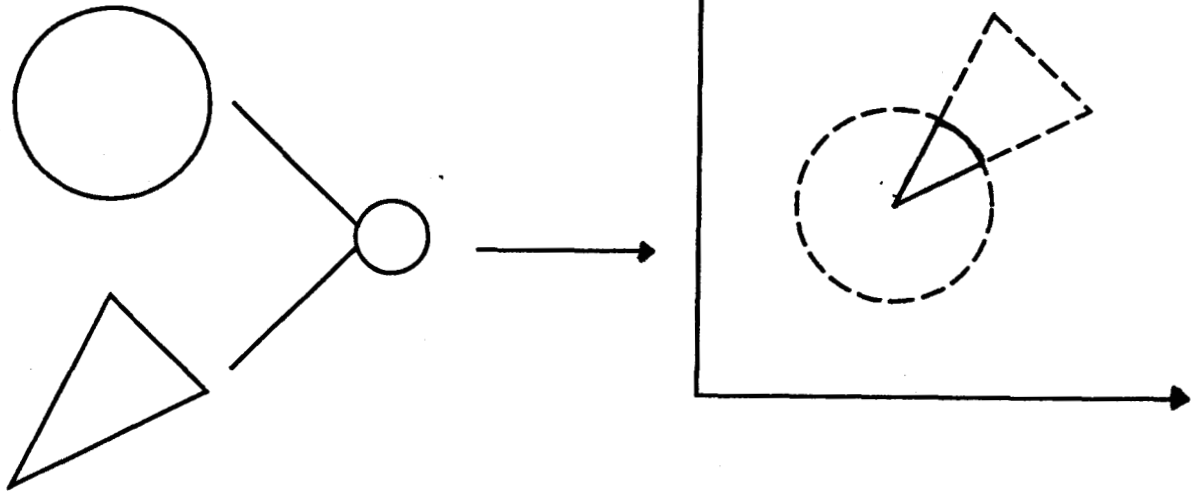


Fig. III.13 - Illustration de l'intersection de deux macro-pixels.

La structure de données naturelle de description d'un objet à l'aide des opérateurs d'union, de différence et d'intersection tels qu'ils ont été définis, est un arbre binaire où chaque feuille représente un macro-pixel et où chaque noeud est une opération concernant une partie de l'objet (Fig.III.14).

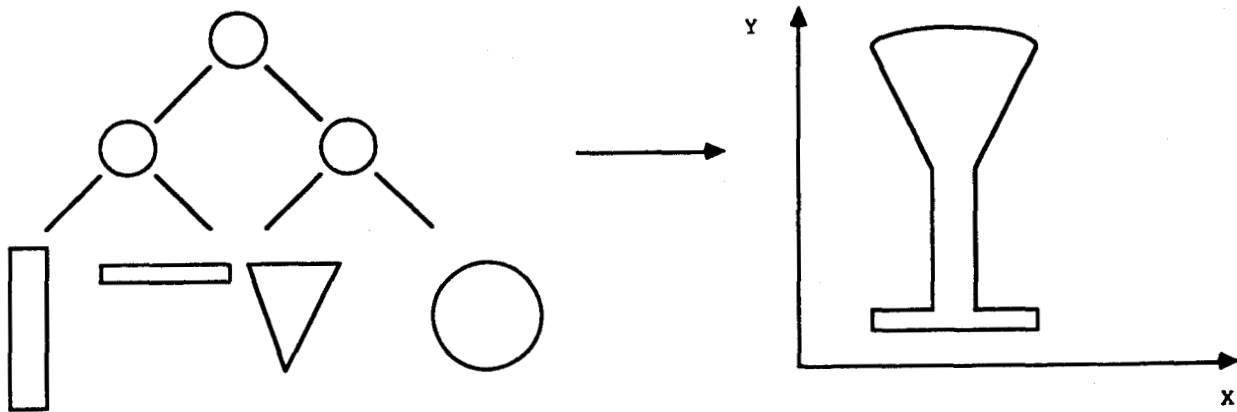


Fig. III.14 - Description d'un objet à l'aide des opérateurs de contour

Cependant, pour avoir une signification réelle, on doit alors tenir compte de l'aspect visuel de l'objet. D'où la nécessité de définir les opérateurs associés à cet aspect.

### III.5.2 - Les opérateurs de texture

Les opérations qu'on vient de décrire permettent un recouvrement entre des macro-pixels ou entre des parties de l'objet qu'on veut décrire. Il est par conséquent important de spécifier la loi de recouvrement dans son aspect visuel, définie dans la zone de recouvrement.

Pour cela, nous allons d'abord voir les caractéristiques des surfaces des objets, puis nous verrons comment appliquer de telles propriétés au modèle macro-pixel.

#### III.5.2.1 - Propriétés des surfaces

Un rayon lumineux qui rencontre un objet peut suivant la nature de celui-ci être réfléchi partiellement ou totalement, ou traverser l'objet en étant ou non dévié.

Pour tenir compte des différentes réactions des surfaces des objets, on utilise dans le calcul d'ombrage, trois coefficients caractérisant les surfaces :

- La réflectance, que l'on note  $R_\lambda$ , qui fournit la proportion de lumière que peut réfléchir une surface soumise à un éclairage. Cette réflectance peut varier selon la position où l'on se place sur une surface donnée.
- La transparence  $T$ , qui quantifie le degré d'opacité d'un solide et permet de prendre en compte dans les calculs, la lumière traversant les objets et provenant de l'arrière des objets.

- La réfraction  $r$ , qui traduit comme précédemment, une quantité de lumière traversant les objets, mais obéit à des lois différentes.

### III.5.2.2 - Application au macro-pixel

Il vient de ce qui précède qu'un objet suivant son degré d'opacité, peut lorsqu'il y a recouvrement, laisser passer la totalité ou une partie de la lumière réfléchie, ou être complètement opaque et ne rien laisser filtrer.

La zone de recouvrement occupe un certain emplacement dans l'image, ou ce qui est équivalent dans l'espace objet. Par conséquent, dire qu'un objet, recouvrant une partie d'un autre objet, est opaque veut dire qu'aucune lumière de l'objet recouvert ne peut passer, par conséquent la caractéristique visuel de la zone de recouvrement sera celle de l'objet en avant plan. Par contre, dire qu'un objet est complètement transparent, implique que la caractéristique visuel de la zone de recouvrement sera celle de l'objet en arrière plan. Entre les deux cas extrêmes, il y aura une diffusion du premier dans le second, ou du second dans le premier.

Pour notre cas, chaque macro-pixel étant défini dans son aspect visuel, par un polynôme du second degré en  $x$  et  $y$ . Il en résulte que les caractéristiques visuelles de la zone de recouvrement seront :

- celles de l'objet en avant plan, si cet objet est considéré comme opaque (pourcentage de diffusion nul).
- celles de l'objet en arrière plan, si l'objet en avant plan est complètement transparent (pourcentage de diffusion égale à 1).
- somme de polynômes suivant le pourcentage défini pour l'objet en avant plan, si cet objet est considéré comme partiellement

transparent ou opaque (pourcentage de diffusion compris entre 0 et 1).

On remarque alors, que les caractéristiques visuelles de la zone de recouvrement dépendent étroitement de l'objet en avant plan. D'où la nécessité de restituer cette dimension (profondeur). Pour notre cas, étant donné que le macro-pixel est défini dans un espace plan, on réalisera alors du 2D 1/2 en attribuant une valeur de profondeur à ce dernier, devant faire alors partie des attributs macro-pixels.

La prise en compte de telles caractéristiques, lors de la description d'un objet, se fera pour les opérations d'union et d'intersection (l'opérateur de différence n'intervient pas ici, du fait qu'on n'a pas à traiter le recouvrement dans ce cas).

Par conséquent, ces deux opérateurs (U, ) auront un double rôle, permettant de décrire le contour de l'objet, ainsi que sa texture.

Le paragraphe suivant permet de spécifier les différents paramètres nécessaires au modèle macro-pixel.

### **III. 6 - Les attributs des macro-pixels**

La description d'un objet quelconque dans le modèle macro-pixel est obtenue à l'aide d'un arbre binaire, permettant de distinguer deux types d'objets :

- les macro-pixels de base
- les sous-objets obtenus par combinaison de macro-pixels.

Ce qui implique de définir deux structures de données pour chaque type d'objet.

### III.6.1 - Les attributs du macro-pixel

A la lumière de ce qui a été dit précédemment, un macro-pixel sera défini par :

- son numéro d'identification permettant de distinguer les macro-pixels entre eux.
- son type, qui peut être : triangle, rectangle, cercle ou ellipse
- ses attributs de position : coordonnées des sommets caractéristiques, centre, rayon...
- les coefficients du polynôme représentant l'aspect visuel
- son orientation, valeur de l'angle d'orientation dans l'espace objet
- son point de référence, permettant de situer et de manipuler le macro-pixel dans l'espace image
- son pourcentage de diffusion déterminant le degré de transparence
- sa profondeur.

Ainsi, les 4 premiers attributs du macro-pixel déterminent l'aspect intrinsèque et sémantique ponctuel du macro-pixel. Les autres attributs définissent son aspect descriptif et son apport sémantique dans l'image.

### III.6.2 - Les attributs des sous-objets

Un sous-objet sera défini par :

- le numéro d'identification du sous-objet considéré
- le type d'opération qu'on doit réaliser
- les identificateurs des sous-objets à combiner



- la valeur des points de référence dans l'espace objet
- le point de référence du sous-objet
- la profondeur résultant de l'opération de combinaison et valable sur toute la surface obtenue
- le degré de transparence résultant.

Ces attributs permettent de définir la forme de l'objet résultant et, suivant le type d'opération, son aspect visuel.

A noter que les deux derniers attributs ont été inclus afin de pouvoir réaliser d'autres opérations entre des sous-objets. Par ailleurs, si l'opération réalisée est la différence, la valeur de ces 2 attributs sera celle du premier sous-objet ou macro-pixel.

### III.7 - Conclusion

Nous avons essayé dans ce chapitre, de définir un nouveau modèle de représentation apte à répondre efficacement aux besoins actuels de l'imagerie informatique. Nous avons montré qu'il fallait concevoir un modèle d'image basé non plus sur une description pixel, mais plutôt sur une description globale de l'image ou des objets la composant. Le modèle macro-pixels doit alors répondre aux contraintes de :

- compression d'image
- prise en compte de l'aspect contour d'un objet, dont l'apport sémantique, quant à la compréhension de l'image, est non négligeable
- possibilité de désignation rapide d'objets directement de l'image visualisée
- relations avec l'application.

Nous avons alors montré que tout objet devait être représenté par un arbre binaire de combinaison de macro-pixels, à l'aide des opérateurs d'union, de différence et d'intersection.

Une telle représentation permet d'avoir alors une forte relation avec l'application. D'autre part, les informations concernant l'aspect contour peuvent être extraites facilement à partir des attributs des macro-pixels, particulièrement le type et les attributs de position. Par ailleurs, la désignation des objets directement de l'image peut être facilement réalisée étant donné que chaque macro-pixel occupe une certaine zone dans l'espace image, et que les relations entre l'objet et les macro-pixels sont complètement définies par l'arbre binaire de description. Enfin, une description d'objets par zones ou taches permet une grande compression de l'image.

Cependant, un tel modèle pose des problèmes :

- de décomposition d'objets en macro-pixels, dûs au fait que dans ce modèle on autorise le recouvrement entre objets, qui impose des lois sur des valeurs difficilement résolues.
- de connaissance à priori du type de texture obtenu à partir d'un polynôme donné. A noter dans ce cas, que le passage des valeurs des pixels au polynôme correspondant peut être facilement réalisé [KOCH 83].

Néanmoins, nous pensons qu'un tel modèle permettra de pousser plus loin les limites atteintes par le modèle pixels, en particulier dans les domaines de :

- compression d'image
- détermination de l'aspect contour d'un objet

- relation avec l'application
- reconnaissance d'objets directement de l'image

Ayant défini le modèle macro-pixel, nous allons proposer maintenant un type d'architecture adapté à un tel modèle.

## **CHAPITRE IV**

### **PROPOSITION D'ARCHITECTURE POUR UNE MACHINE MACRO-PIXEL**

Nous allons essayer, dans ce chapitre, de proposer un type d'architecture de machine qui soit compatible avec la modélisation macro-pixels d'une image. Pour cela nous allons dans un premier temps, étudier quelques machines existantes ou pensées pour la génération d'images réalistes. Par la suite, nous essayerons de déterminer le degré d'adaptabilité de ces machines au modèle proposé.

#### IV.1 - Introduction

Les problèmes de la génération d'images réalistes peuvent être de deux sortes, liés :

- 1 - à la description formelle des objets composant la scène.
- 2 - à la génération temps réel des images ainsi décrites.

Si, pour la description formelle des objets, les progrès ont été remarquables ces dernières années (modélisation CSG, BREP,...) permettant de décrire de manière assez fine les objets de la scène en 3D ; il n'en va pas de même pour la génération temps réel de ces images.

En effet, les machines existantes fournissent des images de haute qualité en des temps variant entre quelques minutes et ... quelques heures si l'on utilise, par exemple, la technique de lancer de rayon [BOUV 84]. Cet inconvénient majeur a poussé les concepteurs à rechercher les moyens de réduire les temps de réponse, prohibitifs eu égard à la complexité des traitements qu'on doit effectuer ainsi qu'au grand nombre de données qu'on doit traiter.

Trois voies de recherche se distinguent :

- l'amélioration des algorithmes de génération.

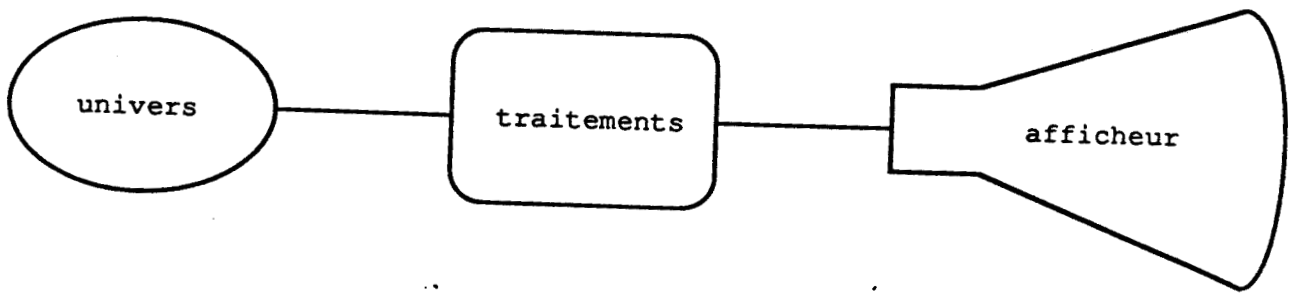
- tracé de cercles, de segments
  - remplissage
  - calcul d'ombrage...
- 
- la spécialisation du système graphique à un domaine limité d'application : simulateur de vol, CFAO, mais aussi jeux vidéo.
  - Etude des machines exploitant le parallélisme existant due à la redondance des données et/ou des traitements.

C'est cette dernière voie que nous nous proposons d'explorer, bien que ces trois voies de recherche peuvent être liées.

#### **IV.2 - Exploitation du parallélisme dans les systèmes graphiques**

On peut trouver aujourd'hui sur le marché des machines de production d'images extrêmement différentes, tant du point de vue de leurs performances, que des fonctionnalités qu'elles présentent. Néanmoins, toute machine destinée à produire des images informatiques est constituée de trois parties principales :

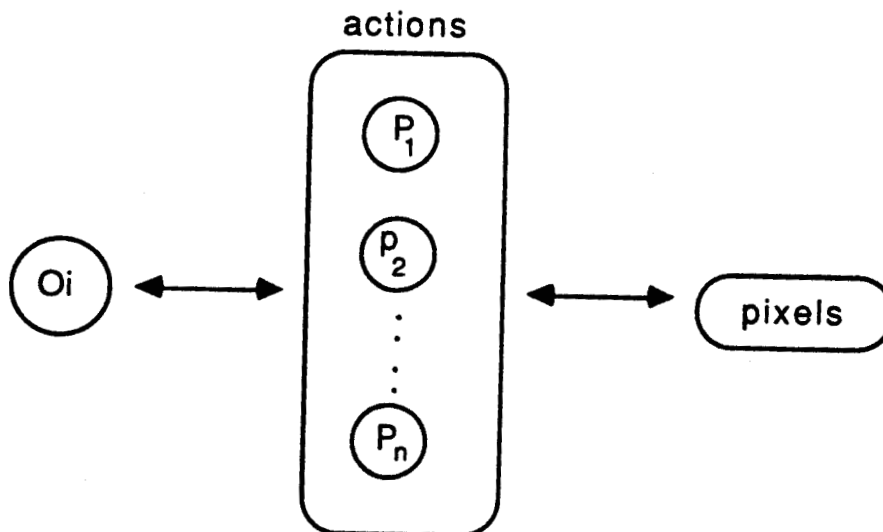
- Une partie de structure de données graphiques permettant de décrire la scène, ou univers, obtenue par le logiciel de description.
- Une partie de traitement produisant les informations nécessaires à la visualisation du dessin ou de l'image.
- Une partie propre de visualisation qui gère le support du dessin.



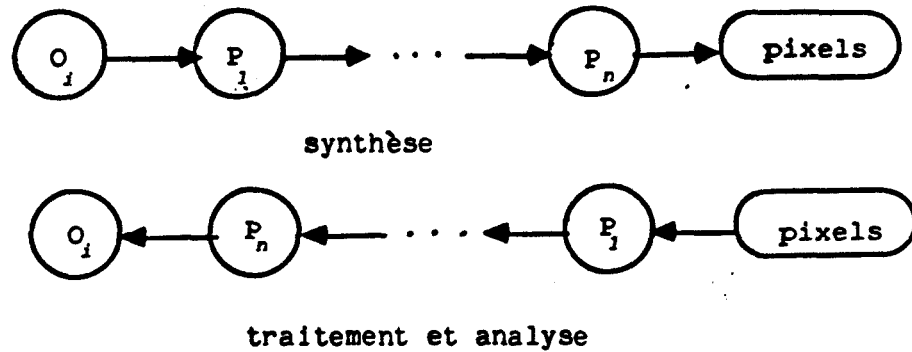
L'exploitation des possibilités de parallélisme dans l'une au moins des trois parties permet de distinguer trois types de parallélisme.

#### IV.2. 1 - Par découpage fonctionnel

Le découpage fonctionnel consiste à traiter les actions liées à la partie traitement en parallèle. On associe alors un processeur à chaque action à réaliser, les objets étant traités séquentiellement. Le parallélisme est alors égal au nombre d'actions qu'on peut réaliser simultanément.

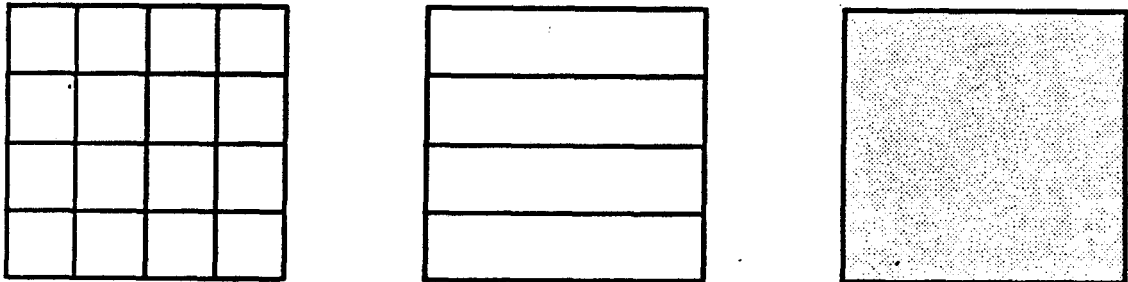


Un cas particulier important est celui des machines pipe-line.



#### IV.2.2 - Par partitionnement de l'espace image

Dans ce cas l'espace de visualisation est divisé en plusieurs zones, chacune d'elles est alors traitée par un processeur qui sera responsable de toutes les actions sur les objets composant cette zone. Un tel découpage peut être constitué de zones carrés, de zones comportant plusieurs lignes de balayage consécutives, de zones comprenant des pixels dispersés.



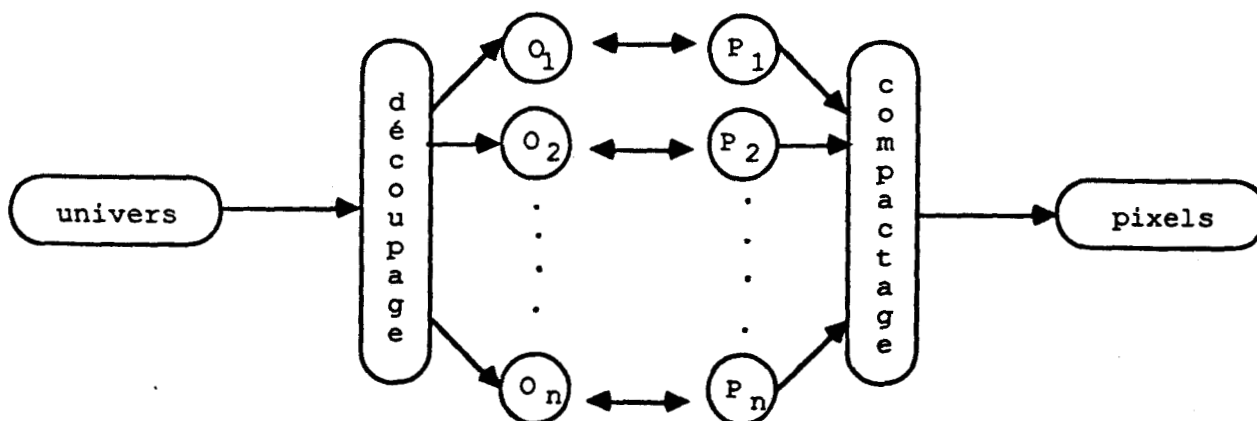
Un cas limite très intéressant est celui où l'on dispose d'un processeur par pixel. Néanmoins, si l'idée de réduire le traitement de toute l'image à plusieurs traitements de sous-domaines pouvant être réalisé en parallèle paraît séduisante des problèmes de découpage des objets et des actions à réaliser par chaque processeur apparaissent. A noter qu'un tel



partitionnement est plus souvent utilisé en analyse qu'en synthèse d'images.

#### IV.2.3 - Par partitionnement de l'espace objet

Dans ce type de partitionnement, l'univers est décomposé en objets élémentaires. Chaque objet étant alors assigné à un processeur. Si l'on ne tient pas compte des relations inter-objets, le parallélisme potentiel est alors égal au nombre d'objets.



Ainsi les objets sont traités en parallèle, par contre les pixels sont générés en séquentiel. Ceci pose un problème de visibilité au niveau du pixel étant donné que tous les processeurs traitent simultanément le même pixel.

Ayant vu les différents types de partitionnement quant au traitement parallèle dans les systèmes graphiques, nous allons voir maintenant quelques exemples de machines de ces trois catégories.

### IV.3. Exemples de machines graphiques à traitement parallèle

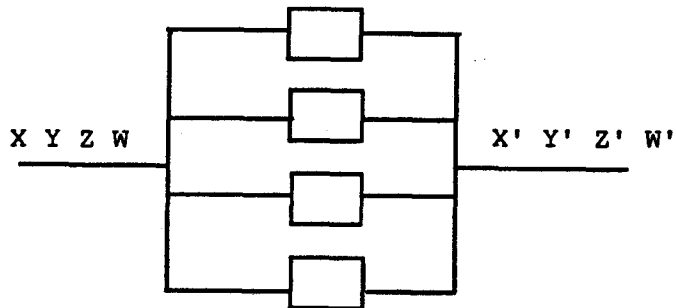
Nous allons exposer dans ce paragraphe quelques machines graphiques à traitement parallèle existantes. Dans ce cas, nous appellerons machine-action, machine-pixels, machine-objets [MER 84] suivant le cas où prédominera un partitionnement fonctionnel, de l'espace image ou de l'espace objets.

#### IV.3.1 - Machine-action

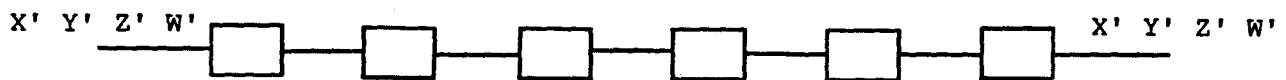
##### IV.3.1.1 - Le "geometry Engine" [CLARK 82]

Dans cette machine, il s'agit de réaliser un parallélisme pipe-line en intégrant, sous forme de VLSI, la plupart des traitements géométriques. Clark distingue trois types d'opérations :

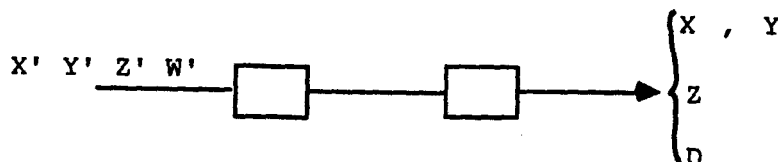
- les transformations géométriques en deux ou trois dimensions réalisées à l'aide d'un produit matriciel  $4 \times 4$  en coordonnées homogènes. Cette opération est obtenue par 4 composants réalisant des opérations simples sur des nombres flottants.



- le fenêtrage (clipping) en deux ou trois dimensions basé sur l'algorithme de découpage réentrant de Sutherland-Hodgman [SUH 74]. Une telle opération nécessite 6 composants en cascade



- la mise en échelle/transformation en perspective étant la dernière opération considérée. Réalisée à l'aide de deux composants fournissant ainsi les coordonnées X et Y que l'on obtiendra sur l'écran, mais aussi la valeur Z utilisable par la suite pour le calcul des lignes cachées, et la valeur D utilisable pour la génération de vues stéréoscopiques.



Le découpage fonctionnel est ici évident. Le système complet comprend ainsi 12 composants réalisant les 3 opérations indiquées intégré à une station de travail graphique (IRIS de Silicon Graphics Inc).

#### IV.3.2 - Machine-pixels

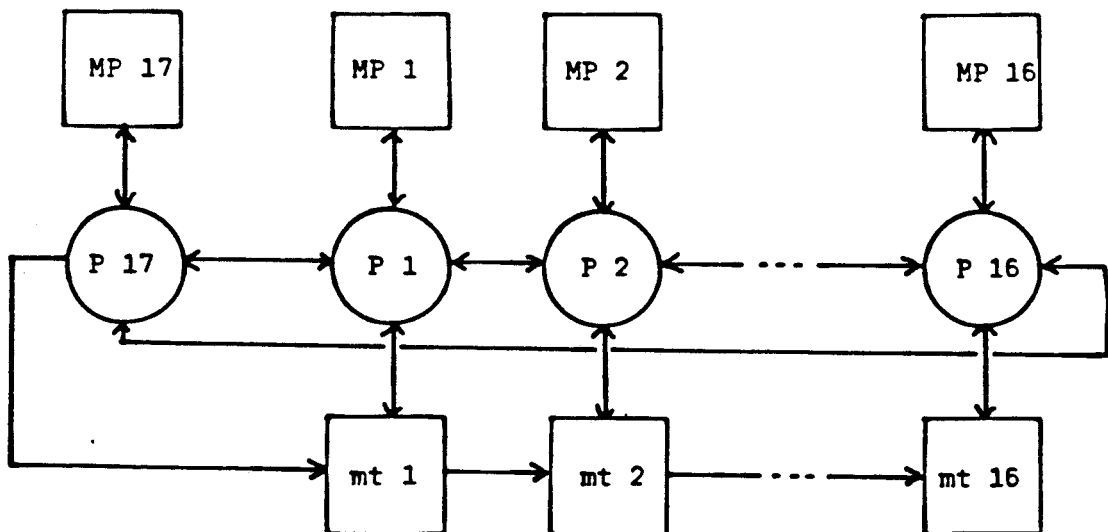
On peut inclure dans cette classe les réseaux cellulaires et les terminaux multiprocesseurs basés sur un découpage géographique. Pour notre cas, nous avons choisi deux exemples de machines : la machine MAP réalisée au Laboratoire d'Informatique de Lille, et la machine Pixel-planes.

#### IV.3.2.1 - MAP(Multiprocesseur Associatif Parallèle) [CORDO 81] [REDJI 84]

Dans cette machine, la mémoire de trame est subdivisée en seize régions formant ainsi une partition de l'espace image (écran). À chaque région est affecté un processeur chargé alors des traitements relatifs à un seizième de l'image à afficher.

Par ailleurs, un dix-septième processeur est assigné à la gestion du calcul d'adresse et du contrôle des traitements.

L'architecture de MAP est illustrée par la figure suivante :



$MP_i$  = Mémoire de Programme du Processeur  $i$

$P_i$  = Processeur  $i$

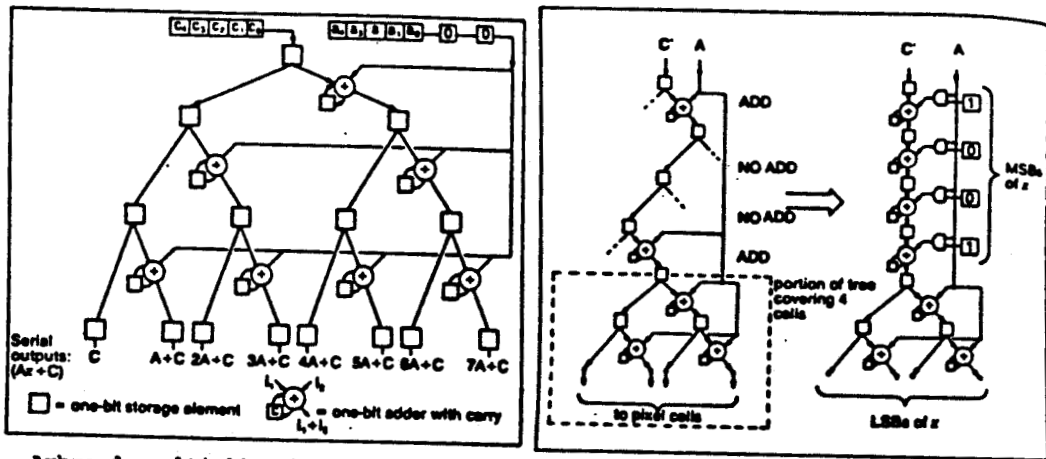
$mt_i$  = Partie de la Mémoire de Trame calculée par  $P_i$

Un certain nombre de logiciels élémentaires classiques au traitement (manipulation de la mémoire de trame) et à la synthèse d'images (tracé de segments de droites, de cercles) [PELERIN 82] ont été implémentés.

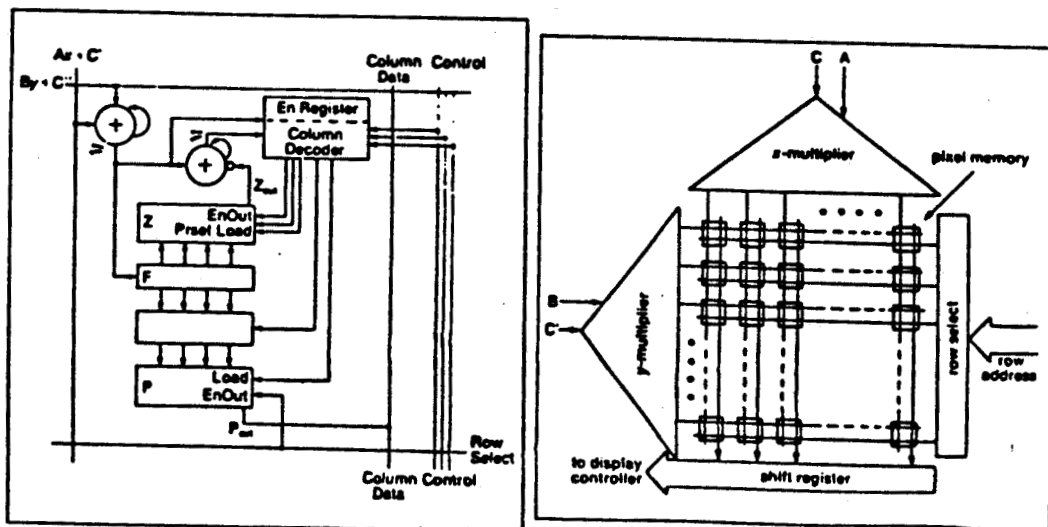
### IV.3.2.2 - Pixel-Planes [FUCH 81]

Pixel-Planes est l'une des rares machines cellulaires de synthèse d'image. Présentée en 1981, par Fuchs et Poulton, Pixel-Planes comporte deux parties : un pré-processeur qui convertit les données polygonales dans une forme adéquate, et un réseau de cellules-mémoires à raison d'une par pixel. Le principe étant d'évaluer simultanément les valeurs d'une fonction linéaire  $F(x,y) = ax + by + c$  pour tout pixel  $(x,y)$  de la mémoire d'image. Une telle architecture permet, en effet, d'effectuer rapidement les procédures graphiques les plus coûteuses [FUCH 85].

- remplissage de polygones
- algorithme du Z. Buffer
- lissage de type Gouraud



Arbre de multiplication : principe et répartition dans chaque cellule



La cellule élémentaire

La machine complète

Le calcul de  $F(x,y)$  est réparti à l'aide d'arbres de multiplication dans l'ensemble des cellules qui sont toutes identiques et contiennent une VAL d'un bit, un bit d'inhibition qui sert de masque et une certaine quantité de mémoires.

Nous voyons alors dans cet exemple que le découpage de l'espace image est poussé à la limite (1 cellule par pixel), néanmoins le problème qui peut se poser est le taux d'utilisation du pré-processeur par rapport au réseau cellulaire. En effet on risque pour certaines opérations graphiques tel le lancer de rayons de solliciter fortement le pré-processeur. Toutefois, les résultats obtenus sont appréciables et les auteurs pensent que cette machine serait capable de traiter 15 à 30 000 polygones par seconde.

A noter qu'une extension de Pixel-planes, appelée Pixel-Powers [GOLD 86], peut évaluer des fonctions quadratiques de la forme :

$$F(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$$

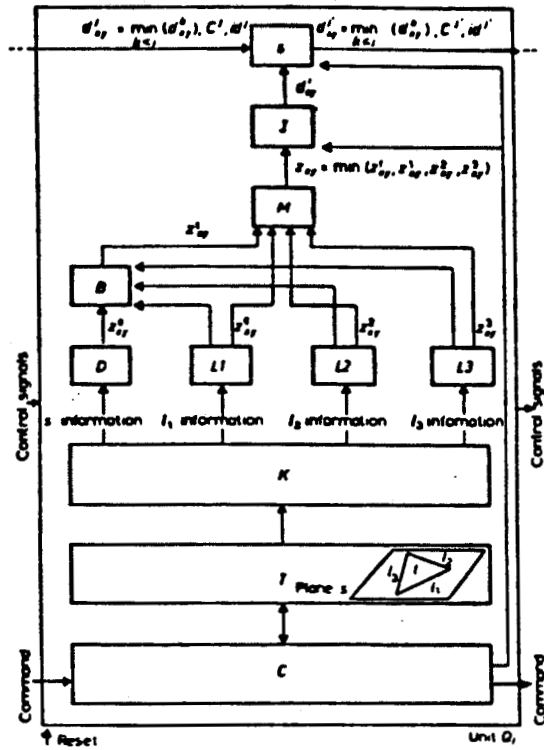
simultanément pour chaque pixel  $(x,y)$ , ce qui permet d'augmenter la précision du calcul.

#### **IV.3.3 - Machine-Objet**

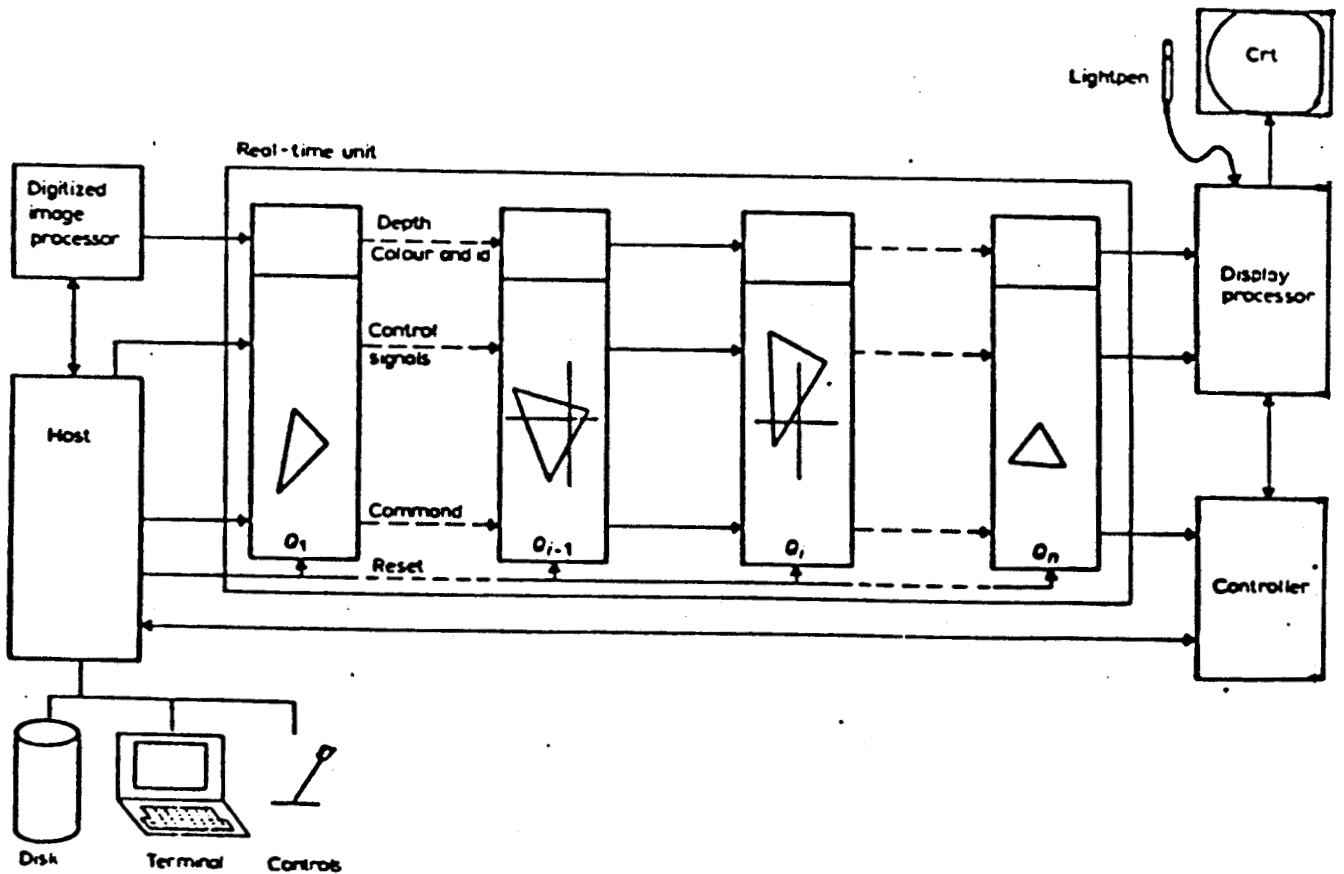
Il s'agit ici de décomposer la scène en sous-espaces, chaque sous-espace étant traité par un processeur. Ce type de parallélisme a donné lieu à plusieurs réalisations dont nous avons choisi deux exemples : ARTEMIS [GRU-CAT 81] et un prototype de machine fonctionnant sans mémoire de trame [DUR 83].

##### **IV.3.3.1 - ARTEMIS [GRU-CAT 81]**

Dans ce cas, un objet est considéré comme étant un ensemble de triangles. La machine comprend donc un grand nombre de "processeurs triangle" sous forme de pipe-line. Chaque processeur triangle a l'architecture suivante :



Les processeurs-triangle sont chargés de la gestion (commandes de création, suppression,... de triangle) et des transformations d'attributs (translation, rotation, mise à l'échelle, perspective), ainsi que la conversion du triangle au rythme du balayage de l'écran. Ces processeurs étant agencés sous forme de pipe-line, l'architecture générale est la suivante :



Toutes les communications (commandes, contrôle, données à afficher...) empruntent ainsi le pipe-line. Les modifications possibles en temps réel se limitent sur les attributs géométriques, de transparence et sur la distance point de vue-écran. Enfin, l'identification est immédiate étant donné que le pipe-line de sélection convoie la profondeur, la couleur et l'identificateur de l'objet prioritaire.

Cependant, certains problèmes subsistent, tels que tolérance aux pannes, traitement de l'antialiasage... Néanmoins, la simulation de cette machine montre qu'il est possible de l'intégrer sous forme de VLSI.

#### IV.3.3.2 - La machine sans mémoire de trame

Le système graphique présenté dans [DUR 83] tend à réaliser trois buts

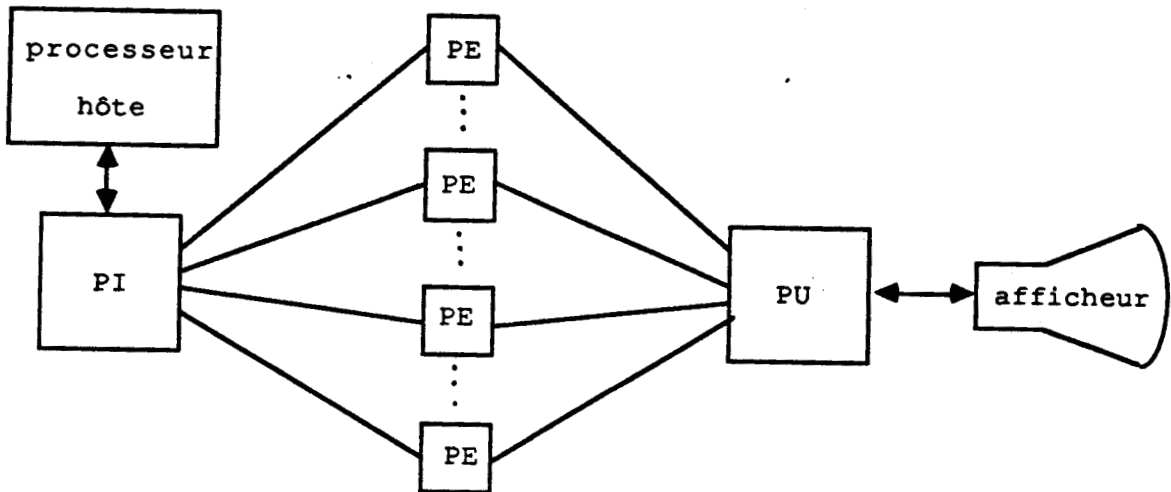
- suppression de la mémoire de trame.
- manipulation d'objets en temps réel.
- modularité et extensibilité de l'architecture.

Le système comprend ainsi trois parties :

- 1 - l'ensemble des processeurs objets, appelés processeurs-éléments (PE).
- 2 - le processeur d'interface PI qui assure le dialogue  
ordinateur hôte <-> {PE}
- 3 - le processeur d'unification (PU) qui permet de récupérer les données fabriquées par les PE et de sélectionner celles qui doivent être affichées.



L'architecture du système graphique proposé est donnée par la figure suivante :



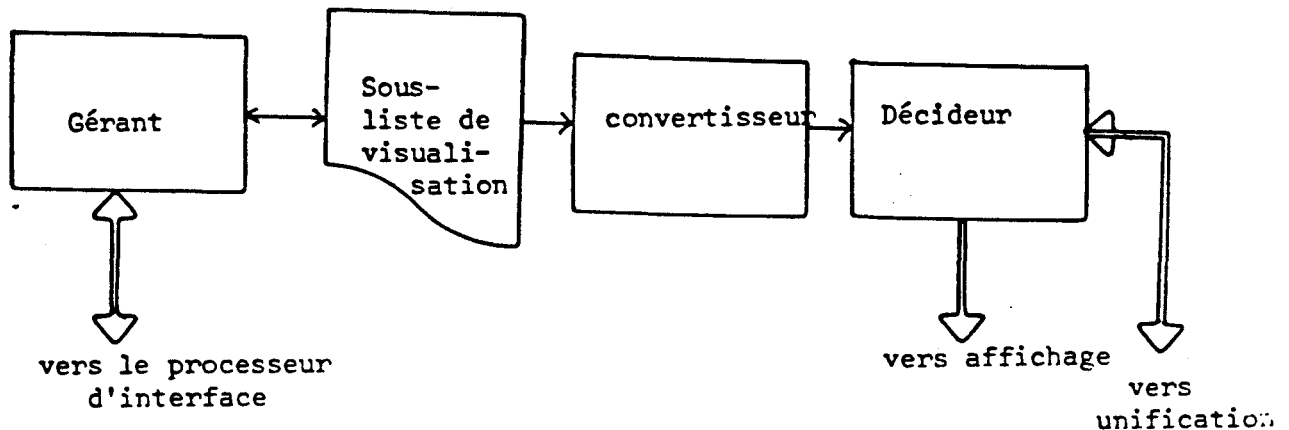
Les objets graphiques de base, ou éléments, manipulés dans ce système sont typés et correspondent à des figures géométriques simples décrites en trois dimensions et interprétables par un PE. Les types considérés sont : le segment de droite, facette polygonale, sphère.

Les objets graphiques de la scène sont structurés à l'aide d'une arborescence. De ce fait, l'implémentation matérielle d'une telle architecture nécessite :

- de figer l'arborescence.
- d'obliger tous les éléments à se trouver au même niveau (le plus bas).

La fonction principale d'un PE est de convertir (en temps réel) au rythme du balayage télévision, ligne par ligne et point par point, la morphologie et l'aspect d'un élément.

Chaque PE contient la liste de visualisation du système graphique, ainsi qu'une sous-liste décrivant l'élément accessible par le PI. L'architecture générale d'un PE est la suivante :



La partie "gérant" assure la gestion de la sous-liste. La partie "décideur" assure quant à elle la fonction d'identification d'objet à l'aide de ce que fournit le PU au PE.

L'avantage offert par une telle architecture est que d'une part l'identification d'un objet repéré par les coordonnées (x,y) d'un point de l'image est plus facile, d'autre part qu'une animation en temps réel est possible à condition que le processeur hôte soit suffisamment rapide. Cependant l'inconvénient majeur est le faible taux de participation de chaque PE à l'élaboration de l'image.

#### IV.4 - Adaptation au modèle macro-pixels

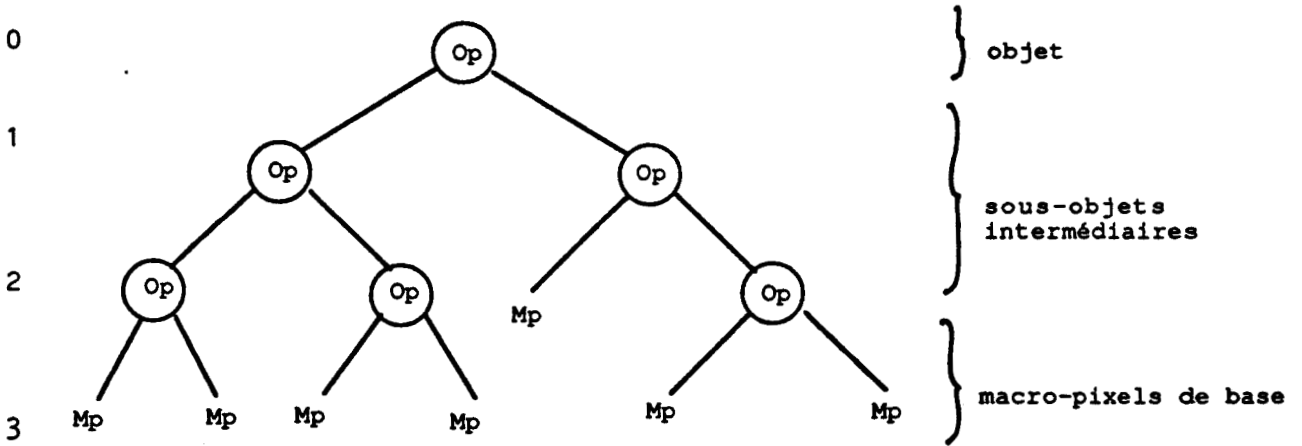
Dans ce paragraphe nous allons essayer de déterminer le type d'architecture, adapté au modèle macro-pixels, qui permettra de générer tout objet graphique décrit sous forme de macro-pixels.

Pour cela nous allons voir le type de partitionnement le plus adapté, puis on fixera les contraintes de réalisation, avant de proposer une solution possible.

#### IV.4.1 - Type de partitionnement

Le but principal du modèle macro-pixel est de permettre une description plus globale d'une image, et plus particulièrement d'un objet graphique quelconque, de manière à préserver tout le sens sémantique de ce dernier. Nous avons vu alors que tout objet graphique peut être obtenu par combinaison de macro-pixels élémentaires, à l'aide d'un certain nombre d'opérateurs relationnels. La description finale d'un objet étant alors un arbre binaire où chaque noeud détermine une opération à effectuer sur les deux branches associées dans l'arbre, et où chaque feuille représente un macro-pixel élémentaire.

niveau



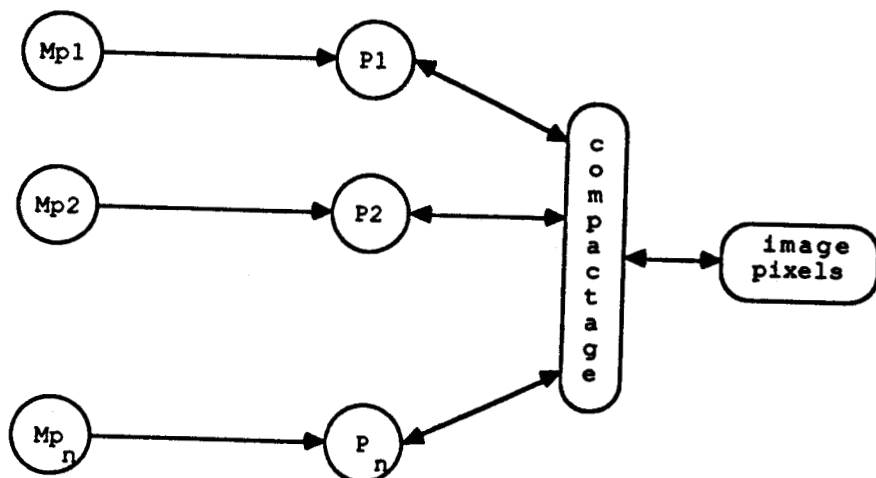
Or le positionnement libre (paramétré) du macro-pixel dans l'image rend le partitionnement de l'espace image inadapté à la représentation macro-pixel.

D'autre part, un découpage fonctionnel ne nous paraît pas évident à cause des fortes relations liant les macro-pixels et les sous-objets. En effet, un sous-objet de niveau  $i$  n'est défini que si les sous-objets associés du niveau  $(i + 1)$  le sont, et ne représente qu'une partie de l'objet. L'aspect séquentiel et répétitif à un certain degré des opérations de traduction de l'arbre rend ainsi difficile la recherche des actions à réaliser sur les objets ou sous-objets.

Par contre, il est possible d'exploiter le parallélisme existant en générant tous les macro-pixels élémentaires en parallèle. Les opérations de combinaison seront alors effectuées lors du compactage final. Ce schéma rend le partitionnement de l'espace objet plus adéquat à la réalisation d'une "machine macro-pixel".

#### IV.4.2 - Les contraintes de réalisation

Le schéma de réalisation est le suivant :



Pour une réalisation matérielle de ce schéma, un certain nombre de problèmes peuvent se poser :

- l'arbre binaire de description de tout objet peut être de niveau variable. Il est évident qu'une réalisation matérielle impose de fixer le nombre de niveaux autorisés.
- l'ordre d'exécution des opérations de combinaison étant strict, ceci implique d'identifier les macro-pixels lors du compactage. Une solution serait alors de distribuer les macro-pixels aux différents processeurs suivant un ordre de parcours de l'arbre et de numéroter les entrées au niveau du compactage.

Par ailleurs une telle machine doit permettre de :

- générer tout objet graphique décrit à l'aide d'un macro-pixel.
- identifier tout objet directement de l'image à partir de la donnée des coordonnées (x,y) d'un point de l'image.
- modifier, en temps réel si possible, les attributs des macro-pixels afin de réaliser certaines opérations géométriques (translation, rotation...).

#### **IV.4.3 - La solution proposée**

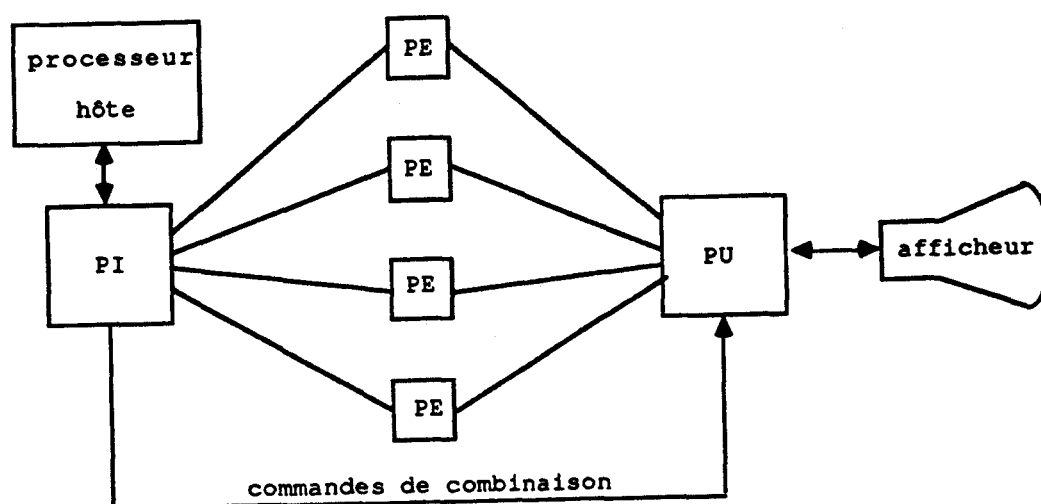
A priori la structure d'une machine permettant de traduire l'arbre binaire de représentation d'objets dans le modèle macro-pixels est une machine Data-Driven ou à Réduction de graphe. Néanmoins, il est pour le moins difficile dans ce cas de générer et de manipuler en bloc des données images (ensemble de pixels) aussi facilement. En effet, il ne faut pas perdre de vue qu'un macro-pixel regroupe un certain nombre de points dans l'image et qu'une opération telle que l'union de deux macro-pixels, pour

ne prendre que la plus simple, nécessite le traitement de tous ces points, qu'on doit par ailleurs générer.

De plus, on a vu que le modèle macro-pixel s'adapte bien à un partitionnement de l'espace objet. De ce fait, c'est cette classe de système graphique qui va nous intéresser. En particulier le système proposé par Durif (Machine Sans Mémoire de Trame)(IV.3.3.2). En effet, une telle machine :

- est extensible, suivant le nombre de PE (processeurs-élément).
- permet l'exploitation efficace du parallélisme potentiel de l'arbre binaire de représentation. Dans ce cas, tous les macro-pixels seront générés en parallèle, par contre les pixels seront générés en séquentiel.
- permet la redéfinition des rôles joués par les différents composants.

Néanmoins certaines modifications doivent être apportées afin de tenir compte des caractéristiques propres au modèle. Le schéma proposé est le suivant :



A ce stade, nous considérons que l'objet graphique est complètement défini par son "arbre macro-pixels", et que tous les paramètres des macro-pixels et des sous-objets intermédiaires sont connus. De plus le nombre de niveaux de l'arbre est fixé.

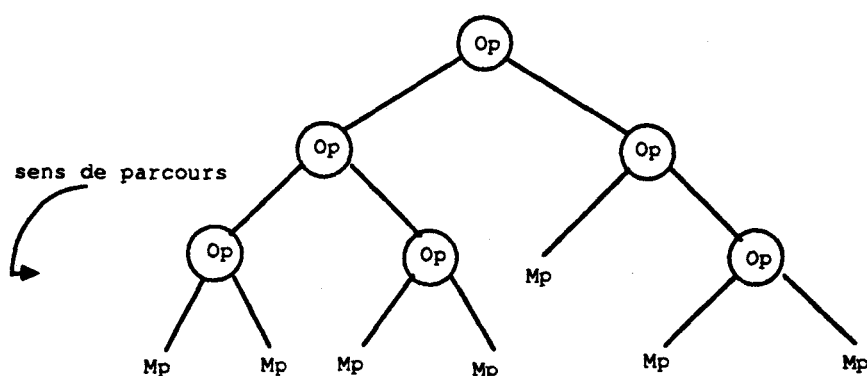
Regardons maintenant le rôle de chaque composant de l'architecture proposée.

#### IV.4.3.1 - Le processeur d'interface (PI)

Le rôle du processeur d'interface est de :

- gérer la communication avec le processeur hôte.
- assigner les macro-pixels aux différents PE.
- envoyer les commandes de combinaison définissant la structuration des opérations de base (union, différence, intersection) au processeur d'unification.

A noter que les macro-pixels doivent être assignés aux différents PE suivant un certain ordre de parcours de l'arbre de représentation.



Celui-ci étant fixe de gauche à droite en raison de la non-associativité de certaines opérations, en particulier la différence ( ) et l'intersection ( )

#### IV.4.3.2 - Le processeur-élément (PE)

Le rôle du PE est double :

- d'abord convertir en temps réel, i.e au rythme du balayage de l'écran, les attributs du macro-pixel en données image (pixels).
- gérer le dialogue amont (avec le PI) et le dialogue aval (avec l'affichage).

Les macro-pixels peuvent être de plusieurs types (cercle, ellipse, rectangle, triangle). Dans ce cas chaque PE doit être capable de convertir tout macro-pixel quel que soit son type. De ce fait, nous excluons de typer les PE.

Par ailleurs, le fait de fixer le nombre de niveaux dans l'arbre de représentation permet de fixer le nombre de PE associés à l'objet et donc une utilisation optimale des PE dans l'élaboration de l'image.

A noter qu'à un instant donné l'ensemble ou un sous-ensemble des PE contient toutes les informations nécessaires à l'élaboration de l'image ou d'un objet graphique.

Le second rôle des PE est de permettre le dialogue avec le PI. Ainsi, peuvent être prises en compte les demandes de modification des attributs modifiables, (tels que attributs de position) et de transformation (rotation, translation, zoom).

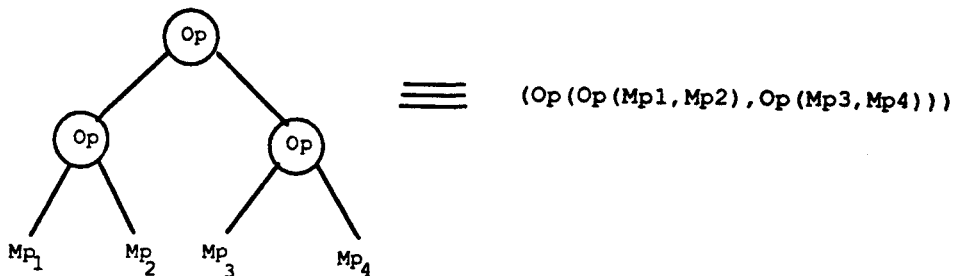
On remarquera, en dernier lieu, que des processeurs graphiques, tel que le TMS 34010 de Texas Instrument [KILLE 90], peuvent être bien adaptés, bien que d'autres choix soient possibles.



#### IV.4.3.3 - Le processeur d'unification (PU)

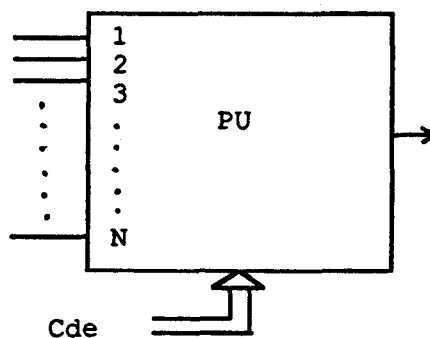
Le rôle du processeur d'unification est de réaliser la synthèse des sorties des PE pour un pixel donné, suivant la commande de combinaison, définissant l'ordre des opérations de base à effectuer envoyée par le PI. Il doit déterminer les caractéristiques d'aspect pour le pixel.

La commande détermine l'ordre d'exécution des opérations de base, ainsi que les paramètres de ces opérations (les différents macro-pixels). Cette commande traduit directement l'ordre binaire de représentation



correspondance de l'arbre binaire et de la commande de combinaison.

Pour cela, nous proposons de numéroter les entrées du PU de manière à simplifier l'identification des macro-pixels.



La commande de combinaison contient deux types de données :

- les opérateurs, en fait leur type (Union, différence, intersection).
- les paramètres identifiant le macro-pixel et nécessaires à l'opération.

Parmi les paramètres d'identification du macro-pixel au niveau de la commande on retrouvera :

- la profondeur  $Z$
- le pourcentage de transparence  $\alpha$

Ainsi les opérateurs seront définis, pour deux macro-pixels A et B, par :

- Union (A,  $Z_A$ ,  $\alpha_A$ , B,  $Z_B$ ,  $\alpha_B$ )
- différence (A,  $Z_A$ ,  $\alpha_A$ , B,  $Z_B$ ,  $\alpha_B$ )
- intersection (A,  $Z_A$ ,  $\alpha_A$ , B,  $Z_B$ ,  $\alpha_B$ ).

Par ailleurs, tous les PE traitant le même pixel simultanément au rythme du balayage de l'écran, nous associerons la valeur 0 au pixel calculé n'ayant pas de sens pour le macro-pixel associé à un PE.

L'algorithme de chaque opérateur est le suivant :

- 1 - **Opérateur d'union** : permet de sélectionner toutes les valeurs issues de A et de B.

```
procédure Union (A,  $\alpha_A$ ,  $Z_A$ , B,  $\alpha_B$ ,  $Z_B$ )  
  Si A = 0 alors sortir (B)  
  Sinon si B = 0 alors sortir (A)  
    sinon si  $Z_A < Z_B$  /* A est avant B */  
      alors sortir (A +  $\alpha_A$  B)  
      sinon sortir (B +  $\alpha_B$  A)  
    fsi  
  fsi  
finproc
```

- 2 - **Opérateur différence** : permet de sélectionner les valeurs de A qui n'appartiennent pas à B.

```
procédure différence (A,  $\alpha_A$ ,  $Z_A$ , B,  $\alpha_B$ ,  $Z_B$ )  
  si B = 0  
    alors sortir (A)  
    sinon sortir (0)  
  fsi  
finproc
```

Remarque on n'utilise pas dans ce cas les valeurs  $\alpha_i$  et  $Z_i$ , pour  $i = A, B$ .

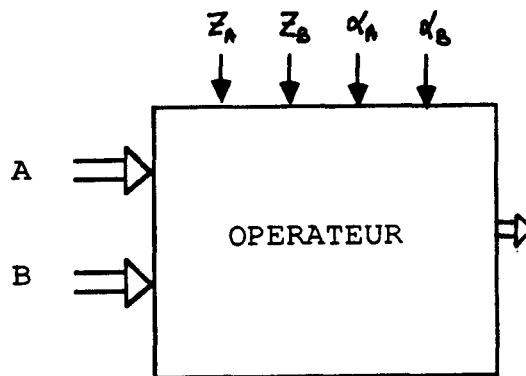
- 3 - **Opérateur d'intersection** : permet de sélectionner les valeurs issues de A et de B. L'algorithme est le suivant :

```

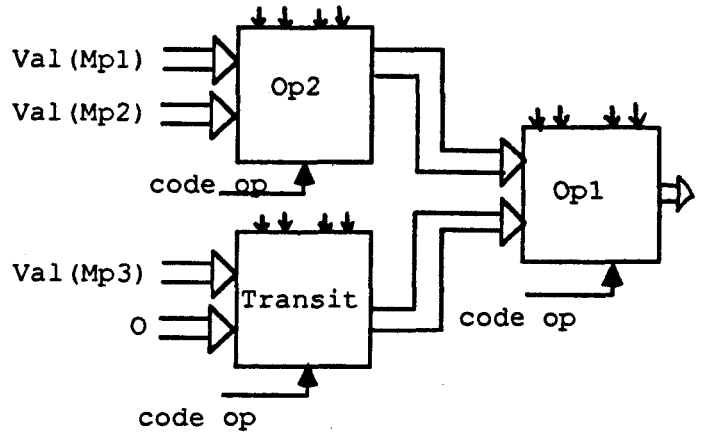
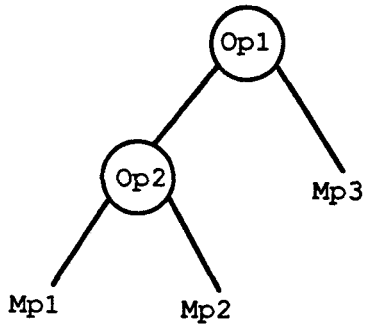
procédure Intersection (A,  $\alpha_A$ ,  $Z_A$ , B,  $\alpha_B$ ,  $Z_B$ )
  si A = 0 ou B = 0
    alors sortir (0)
  sinon si  $Z_A < Z_B$  /* A devant B */
    alors sortir (A +  $\alpha_A B$ )
    sinon sortir (B +  $\alpha_B A$ )
  fsi
fsi
finproc

```

On peut schématiser ces opérateurs par une boîte noire comme suit :

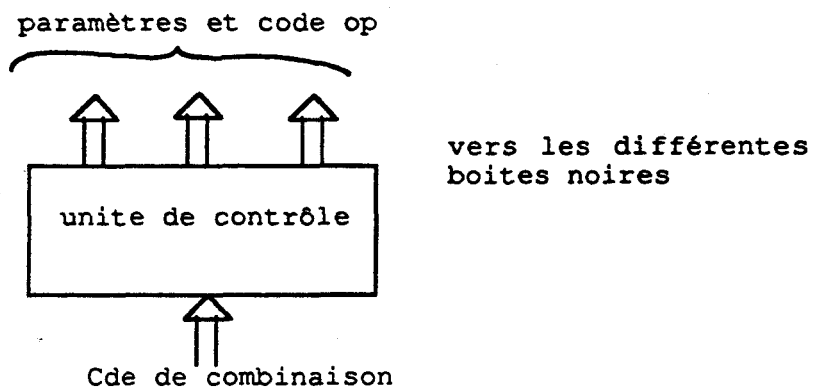


On peut alors associer cette boîte à un noeud de l'arbre. Dans ce cas il est souhaitable de pouvoir programmer cette boîte selon l'opération désirée. Par conséquent, un tel circuit doit être capable de réaliser toutes les opérations de base. En outre, un tel circuit doit aussi remplir la fonction de Transit permettant de faire passer une des deux valeurs obtenues des PE à l'étage supérieur avec un certain retard. Une telle fonction est nécessaire lorsqu'on n'utilise pas tous les opérateurs dans une description d'objet, comme c'est le cas dans l'exemple suivant :



Afin de compléter la structure du PU, il est nécessaire de lui associer une unité de contrôle qui doit avoir pour rôle :

- de réceptionner la commande de combinaison issue du PI.
- de programmer, grâce à un code opération, les différentes boîtes noires (ceci nécessite 2 bits vu qu'on a 4 fonctions à réaliser : les 3 opérateurs plus la fonction de transit).
- de distribuer les paramètres des différentes opérations (Z,  $\alpha$ ) aux boîtes correspondantes.



Néanmoins une étude de faisabilité doit être réalisée particulièrement pour le nombre de portes nécessaires.

#### IV. 5 - Conclusion

Nous avons présenté une architecture de "machines macro-pixel" permettant de générer tout objet décrit dans le modèle proposé. Une telle architecture est fortement inspirée de celle proposée par Durif [DUR 83] bien que les objectifs soient différents.

L'avantage offert par un tel schéma est que toute modification des attributs du macro-pixel peut être prise en compte en temps réel, permettant de réaliser :

- des translations en modifiant les attributs de position.
- des rotations en modifiant les attributs d'orientation.
- une animation en temps réel si le processeur hôte est assez rapide pour calculer toute valeur à modifier, avec la possibilité de ne modifier qu'une partie de l'objet (une branche de l'arbre) et de réaliser par cela l'animation d'une partie de l'objet.

Néanmoins certaines questions restent en suspens, telles :

- la détermination du nombre maximum de niveaux d'arborescence autorisé afin de permettre une génération temps réel. En effet, l'augmentation du nombre de niveaux implique une modification de la structure du PU. Une telle augmentation ne doit pas rendre impossible la faisabilité du PU.
- la synchronisation des opérateurs du PU.

## **CONCLUSION**

## CONCLUSION

Les conséquences de l'absence de relations sémantiques entre les pixels de l'image et les objets véhiculés par celle-ci, nous ont poussé à réfléchir sur un nouveau modèle de description, placé en interface entre la description interne des objets et la représentation finale de l'image, apte à répondre efficacement aux besoins exprimés en imagerie informatique.

Ainsi nous avons montré que tout objet (ou région) peut être représenté(e) par un arbre binaire, permettant de combiner de nouveaux éléments de base que nous avons appelés macro-pixels (dans le sens où un macro-pixel doit être simple à afficher d'une part, et contenir plus d'informations et de propriétés que le pixel d'autre part). Chaque macro-pixel est alors défini par un certain nombre d'attributs déterminant des aspects intrinsèque, descriptif et sémantique supérieurs au pixel.

Nous avons alors montré qu'une telle modélisation permet de:

- réaliser une certaine compression d'image, étant donné que chaque macro-pixel occupe une zone ou tache plus ou moins large dans l'image, regroupant ainsi en une seule information toutes les valeurs des pixels composant cette tache.
- faciliter la relation entre l'objet et sa représentation visuelle grâce à l'identification de l'objet lui-même qui est poussée, par le truchement de l'arbre binaire associé, jusqu'au macro-pixel et donc au pixel.
- prendre en compte l'aspect contour, dont l'apport sémantique est non négligeable (particulièrement en reconnaissance de forme).
- modifier les attributs des macro-pixels, offrant ainsi la possibilité, sous certaines conditions matérielles, d'une animation des objets en temps réel.

Nous avons par ailleurs proposé une implémentation matérielle possible d'une machine macro-pixel permettant de générer tout objet décrit dans ce modèle.



Cependant un certain nombre de problèmes subsistent, tel la décomposition d'objets en macro-pixels avec recouvrement, ouvrant la voie à plusieurs axes de recherche tels :

- l'étude d'un dispositif d'acquisition d'images directement en macro-pixels.
- la faisabilité du processeur d'unification (PU) de la machine macro-pixel proposée.
- la possibilité de visualiser directement les macro-pixels sans passer par une traduction pixel. Nous pensons dans ce cas aux nouvelles technologies de visualisation d'images telles que écrans à plasma et surtout à cristaux liquides.

## **BIBLIOGRAPHIE**

## BIBLIOGRAPHIE

- [ALAM 86] M. ALAMI, L. PERALTA et al  
*Coding picture edge by circle approach*  
Deuxième colloque Image, Nice, avril 1986, p 179-184
- [ALT 88] P. ALT, V. CORDONNIER  
*An evaluation of some three-color tiling patterns*  
A paraître
- [BEAU 84] K.G. BEAUCHAMP  
*Applications of Walsh and related fonctions*  
Academic Press Inc, London 1984
- [BER 77] J. BERTIN  
*La graphique et le traitement graphique de l'information*  
Flammarion, 1977
- [BEZ 86] P. BEZIER  
*Courbes et surfaces*  
Collection Mathématiques et CAO, Vol 4, Hermes, 1986
- [BOUV 84] C. BOUVILLE et al  
*Synthèse d'image par lancer de rayon: algorithme et architecture*  
CESTA, Mai 1984, p 683-696
- [BRES 65] J.E. BRESENHAM  
*Algorithm for computer control of digital plotter*  
IBM Syst. Journ. , n° 4, 1965, p 25-30

- [BURT 83] P.J. BURT, E.H. ADELSON  
*The Laplacien Pyramid as a compact image code*  
IEEE Trans on Comm, Vol COM-31, n° 4, April 1983, p532-540
- [CEI 82] CEA-EDF-INRIA  
*La réalisation des logiciels graphiques interactifs*  
Collection de la direction des études et recherches  
d'Electricité de France, Edition Eyrolles, 1982
- [CHAZ 84] B. CHAZELLE, J. INCERPI  
*Triangulation and shape complexity*  
ACM Trans on Graphics, Vol 3, n° 2, April 1984, p 135-152
- [CLARK 82] J.H. CLARK  
*The Geometry Engine: a VLSI Geometry System for Graphics*  
ACM Trans on Graphics, Vol 3, n°2, April 1984, p 135-152
- [CORDO 81] V. CORDONNIER, MOUSSU  
*The MAP project: an associative processor for speech  
processing*  
Pub interne, USTL, 1981
- [COTT 85] M.S. COTTINGHAM  
*A Compressed Data Structure for surface representation*  
Computer Graphics Forum, n°4, 1985, p 217-228
- [CROW 77] F.C. CROW  
*The Aliasing problem in computer generated shaded images*  
Comm ACM, Vol 20, n° 11, Nov. 1977, p 799-805
- [DUR 83] P. DURIF  
*Etude d'une machine parallèle de synthèse d'images à  
découpage par objet*  
Thèse de 3<sup>ieme</sup> Cycle, LIFL, USTLille, 1983

- [FENG 75] E. FENG, T. PAVLIDIS  
*Decomposition of polygons into simpler components: Feature generation for syntactic pattern recognition*  
IEEE Trans on Comp, Vol C24, n° 6, June 1975, p636-650
- [FER 81] N.F. FERREIRA  
*Conception et réalisation d'un système interactif pour la synthèse d'images réalistes: Hélios*  
Thèse de Docteur-Ingenieur, Grenoble, 1981
- [FOLE 82] J. FOLEY, A. VANDAM  
*Fundamentals of interactive computer graphics*  
Addison-Wesley, Reading, Mass 1982
- [FOUR 84] A. FOURNIER, D.Y. MONTUMO  
*Triangulating simple polygons and equivalent problems*  
ACM Trans on Graphics, Vol 3, n°2, April 1984, p 153-174
- [FRE 61] H. FREEMAN  
*On the encoding of arbitrary geometric configurations*  
IRE Trans on computers, Vol 10, n° 2, June 1961
- [FU 81] K.S. FU, J.K. MUI  
*A survey on image segmentation*  
Pattern recognition, Vol 3, 1981, p 3-16
- [FUCH 81] H. FUCHS, J. POULTON  
*Pixel-Planes: a VLSI Oriented Design for Raster Graphics Engine*  
VLSI Design, n° 3, 1981, p 20-28
- [FUCH 85] H. FUCH, J. GOLDFEATHER and al  
*Fast spheres, textures, transparencies and image enhancements in Pixel-Planes*  
Computer Graphics, Vol 19, n° 3, 1985, p111-120

- [GOLD 86] J. GOLDFEATHER, J.P.M. HULTQUIST, H. FUCHS  
*Fast constructive solid geometry display in the Pixel-Power Graphics system*  
Computer Graphics, Vol 20, n° 4, August 1986, p 107-116
- [GROS 87] M. GROS  
*La modélisation 3D pour l'utilisateur: Presentation générale, application au système CAM-X*  
Revue de CFAO et d'infographie, Vol 1, n°4, 1987,p17-39
- [GRU-CAT 81] R. GRUIA-CATALIN, K. TAKAYUKI  
*VLSI perspective of real time hidden surface elimination*  
Computer Aided Design, Vol 13, n°2, Mars 1981
- [GUPT 81] S. GUPTA, R.F. SPROULL  
*Filtering edge for gray scale displays*  
Computer Graphics, Vol 15, n° 3, August 1981, p 1-5
- [HAR 72] H. F. HARMUTH  
*Transmission of information by orthogonal fonctions*  
2<sup>nd</sup> Ed, Springer-verlag, Berlin 1972
- [HERT 83] S. HERTEL, K. MELHORN  
*Fast triangulation of simple polygons*  
Proc of the 1983 International FCT-Confernce, Bergholm, Sweden, August 1983,p207-218
- [JAIN 81] A.K. JAIN  
*Image data compression: A Review*  
Proc of IEEE, Vol 69, n° 3, Mars 1981, p349-389
- [JERN 81] M.E. JERNIGAN  
*Does the eye contain optimal edge detection mechanisms ?*  
IEEE Trans on System,Man and Cybernetics, Vol SMC-11, n°6, June 1981, p441-444

- [KIL 81] A.C. KILGOUR  
*A hierarchical model of a graphic system*  
Computer graphics, April 1981
- [KILLE 86] C.R. KILLEBREW  
*The TMS 34010 Graphics System Processor*  
Byte, Dec 1986, p193-202
- [KOBA 74] H. KOBAYASHI, L.R. BAHL  
*Image data compression by predictive coding:*  
*Part I - Prediction algorithms*  
IBM J. Res. and Develop., n° 18, Mars 1974, p164-171
- [KOCH 83] M. KOCHER  
*Codage d'images à haute compression basé sur un modèle*  
*Contour - Texture*  
Thèse de Docteur es-sciences techniques, Lausanne, EPFL,  
1983
- [KUNT 84] M. KUNT, A. IKONOMOPOULOS  
*Compressin d'images: Méthodes de la deuxième génération*  
Premier colloque image, Biarritz, Mai 1984, p93-105
- [LAID 86] D.H. LAIDLAW, J.F. HUGHES  
*Constructive Solide Geometry for polyhedral objects*  
Computer Graphics, Vol 20, n° 4, August 1986, p161-168
- [LUC 77] M. LUCAS  
*Contribution à l'étude des techniques de communication*  
*graphique avec un ordinateur - Eléments de base des*  
*logiciels graphiques interactifs*  
Thèse d'Etat, Grenoble, Dec 1977

- [MARR 80] D. MARR, E. HILDRETH  
*Theory of edge detection*  
Proc of Roy Soc London, Serie B, Vol 207, n° 1167,  
Fev 1980, p 187-217
- [MAR 82] F. MARTINEZ  
*Vers une approche systématique de la synthèse d'image -  
Aspects logiciel et materiel*  
Thèse d'Etat, Grenoble, Nov 1982
- [MER 79] M. MERIAUX  
*Etude et réalisation d'un terminal graphique couleur  
tridimensionnel fonctionnant par taches*  
Thèse de Docteur-Ingénieur, USTLille, LIFL, 1979
- [MER 84] M. MERIAUX  
*Contribution à l'imagerie informatique: aspects  
algorithmiques et architecturaux*  
Thèse d'Etat, USTLille, LIFL, 1984
- [MICH 87] D. MICHELUCCI  
*Constructions de BRep, problèmes d'imprecision*  
Thèse , Ecole des Mines, Saint-Etienne, 1987
- [MORV 76] P. MORVAN, M. LUCAS  
*Introduction à l'infographie interactive*  
Larousse Université, 1976
- [NETRA 80] A.N. NETRAVALI, J.O. LIMB  
*Picture coding: A Review*  
Proc of IEEE, Vol 68, n° 3, Mars 1980, p366-406
- [NEW 79] W. NEWMAN, R.F. SPROULL  
*Principles of interactive computer graphics*  
Mac Graw-Hill, 2<sup>nd</sup> Ed, 1979



- [OLEJ 84] R. OLEJNIK  
*Synthèse interactive de texture*  
Thèse de 3<sup>ième</sup> Cycle, USTLille, LIFL, Nov 1984
- [OULD 87] D. OULD-BRAHIM  
*Quelques aspects sur les méthodes de compression numérique des images: 1- Méthodes de première génération*  
Rapport interne, n°51, USTLille, LIFL, Juin 1987
- [OULD 88] D. OULD-BRAHIM  
*Quelques aspects sur les méthodes de compression numérique des images: 2- Méthode de seconde génération*  
Rapport interne, n°55, USTLille, LIFL, Fevrier 1988
- [OULD 88] D. OULD-BRAHIM, M. MERIAUX  
*Triangularisation de polygones quelconques*  
Rapport interne, n°56, USTLille, LIFL, Fevrier 1988
- [PAVL 78] T. PAVLIDIS  
*A Review of algorithms for shape analysis*  
Computer Graphics and Image Proc, Vol 7, n° 2, 1978,  
p243-258
- [PELERIN 82] M. PELERIN  
*Conception et réalisation d'outils de synthèse d'images nécessaires à l'élaboration d'un logiciel graphique sur MAP*  
Mémoire de DEA, Lille I, 1982
- [PERO 88] B. PEROCHE  
*La synthèse d'images*  
Ed Hermes, Paris, 1988

- [PHONG 75] B.T. PHONG  
*Illumination for computer generated pictures*  
Comm ACM, Vol 18, n°6, June 1975, p311-317
- [PITT 80] M.L.V. PITTEWAY, D.J. WATKINSON  
*Bresenham's algorithm with gray scale*  
Comm ACM, Vol 23, n°11, Nov 80, p625-626
- [PLE 88] D. PLEMENOS  
*La modélisation des volumes*  
ATP DU CNRS, Lyon, 26-27 Avril 1988
- [PRAT 78] W.K. PRATT  
*Digital image processing*  
A Wiley-Interscience publication, New-York, 1978
- [REDJI 84] M. REDJIMI  
*Etude et réalisation d'un système parallèle pour le traitement graphique*  
Thèse de Docteur-Ingénieur, USTLille, LIFL, 1984
- [REEV 83] W.T. REEVES  
*Particle systems - A technique for modeling a class of fuzzy objects*  
Computer Graphics, Vol 17, n°3, 1983, p359-376
- [REEV 85] W.T. REEVES, R. BLAU  
*Approximate and probabilistic algorithms for shading and rendering structured particles systems*  
Computer Graphics, Vol 19, n°3, 1985, p 313-322
- [REQ 80] A.A.G. REQUICHA  
*Representations for rigid solids: Theory, methods and systems*  
Computer surveys, Vol 12, n°4, Dec 1980, p437-464

- [REQ 85] A.A.G. REQUICHA, H.B. VOELCKER  
*Boolean operations in solid modeling: Boundary Evaluation  
and merging algorithms*  
Proceedings IEEE, Vol 73, n°1, Janv 1985, p 30-44
- [ROTH 82] S. D. ROTH  
*Ray casting for modeling solid*  
Computer Graphics and image processing, Vol 18, 1982,  
p109-144
- [SAM 83] H. SAMET  
*A quadtree Medial Axis Transform*  
Comm ACM, Vol 26, n°9, Sep 1983, p680-693
- [SCHA 78] B. SCHACHTER  
*Decomposition of polygons into convexs sets*  
IEEE Trans on computers, Vol C27, n° 11, Nov 1978,  
p1078-1082
- [SEDE 86] T.W. SEDERBERG, D.C. ANDERSON  
*Free-form deformation of solid geometric models*  
Computer Graphics, Vol 20, n°4, August 1986, p151-160
- [SMIT 84] A.R. SMITH  
*Plants, Fractals and formal languages*  
Computer Graphics, Vol 18, n°3, 1984, p1-9
- [SUH 74] I.E. SUTHERLAND, G.W. HODGMAN  
*Reentrant polygon clipping*  
Com ACM, Vol 17, n°1, 1974, p 32
- [VERO 87] M. VERON  
*Modélisation de courbes et de surfaces*  
Revue de CFAO et d'Infographie, Vol 2, n°3, 1987, p39-66

- [WHIT 83] T. WHITED  
*Anti-aliased line drawing using brush extrusion*  
Computer Graphics, Vol 17, n°3, July 1983, p151-156
- [WIL 83] R. WILSON et al  
*Anisotropic nonstationary image estimation and its applications Part II - Predictive image coding*  
IEEE Trans on Comm, Vol COM-31, n°3, Mars 1983, p383-406
- [WINTZ 72] P.A. WINTZ  
*Transform picture coding*  
Proc of IEEE, Vol 60, n°7, July 1972, p809-820
- [WOO 85] T.C. WOO, S.Y. SHIN  
*A linear time algorithm for triangulating a point-visible polygon*  
ACM Trans on Graphics, Vol 4, n°1, January 1985, p60-70
- [WU 86] A.Y. WU et al  
*computation of geometric properties from the Medial Axis Transform in  $O(n \log n)$  time*  
Computer Vision Graphics and Image Processing, n°34, 1986  
p76-92

***Annexe***

# Annexe A

## Triangularisation de Polygones Quelconques

### 1 Introduction

Dans ce papier, nous présentons un algorithme de triangularisation de polygones simples quelconques utilisant certaines propriétés des chaînes de sommets concaves et basé sur la recherche et la triangularisation de polygones *convexes*, en *spirales* ou en *T* ( fig 1 ) déterminés à partir du polygone initial. En effet, en remarquant que la triangularisation de ces trois types de polygones est une opération rapide— dans le cas particulier des polygones convexes et en spirale, de complexité temporelle de  $O(r)$  où  $r$  est le nombre de sommets de chaque polygone — tout le problème revient à décomposer le polygone  $P$  en ces trois formes et de triangulariser par la suite chacune des figures obtenues.

Nous adopterons, pour la définition de la triangularisation, celle donnée par [WOO 84] , où le problème de triangularisation d'un polygone simple  $P$  à  $n$  sommets est défini comme étant la recherche d'un ensemble de  $(n-3)$  segments disjoints, joignant chacun deux sommets non adjacents de  $P$  tel que :

- tout segment obtenu se trouve complètement inclus dans le polygone
- l'ensemble des segments permettent, avec les segments joignant les sommets du polygone, de partitionner  $P$  en  $(n-2)$  triangles sans recouvrement possible entre deux triangles

- l'ensemble des segments permettent, avec les segments joignant les sommets du polygone, de partitionner P en  $(n-2)$  triangles sans recouvrement possible entre deux triangles

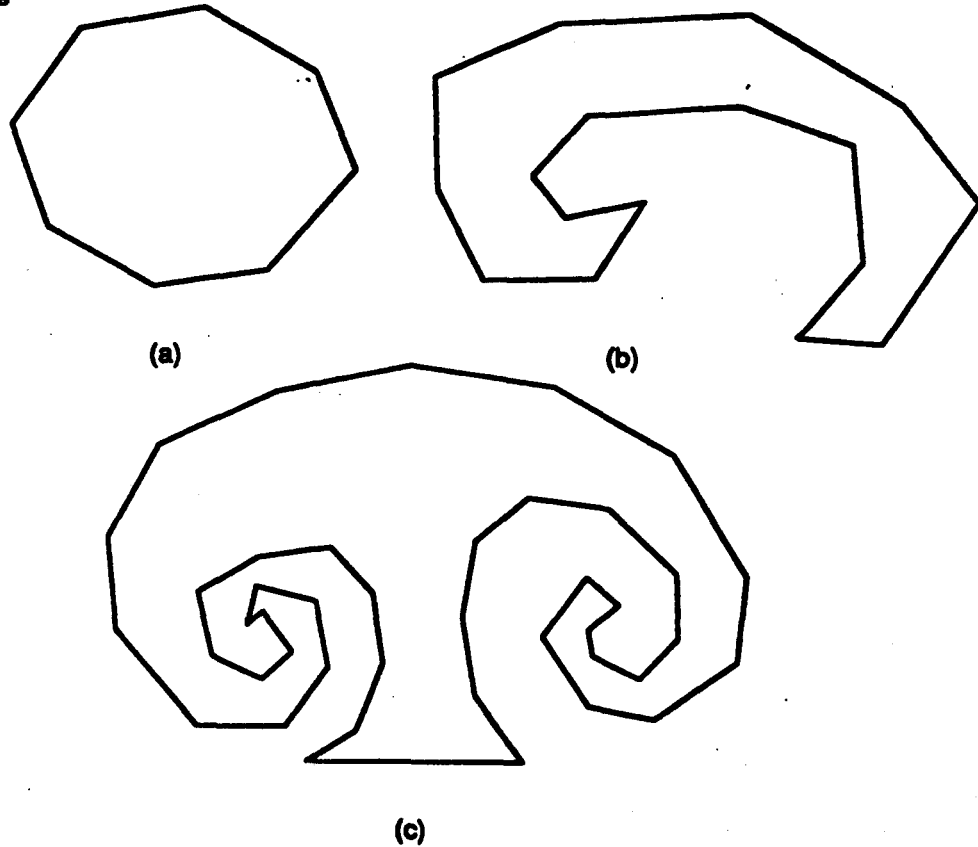


Fig 1 — Exemples de polygones (a) convexe, (b) spirale, (c) en T

Dans notre cas, un polygone P sera décrit par une succession de sommets données dans le sens des aiguilles d'une montre :

$$P = s_1 s_2 \dots s_i \dots s_{n-1} s_n$$

tel que :

- $\text{Succ}(s_i) = s_{i+1}$  et  $\text{Pred}(s_i) = s_{i-1}$
- $\text{Succ}(s_n) = s_1$  et  $\text{Pred}(s_1) = s_n$

De plus on ne considère que des polygones simples, c'est à dire sans intersection possible entre les segments du polygone.

Avant de développer les différents algorithmes de triangulation, nous allons donner quelques définitions et notations devant nous servir pour la preuve de convergence de nos algorithmes.

## 2 Définitions et Préliminaires

Soit un polygone simple  $P$  à  $n$  sommets :

$$P = s_1 s_2 \dots s_i \dots s_j \dots s_{n-1} s_n$$

- On note par  $A(s_i)$  la valeur angulaire de l'angle formé par les 3 sommets successifs :  
 $s_{i-1}, s_i, s_{i+1}$
- Un sommet  $s_i$  de  $P$  est dit convexe et noté par  $x$  si  $A(s_i) < \pi$
- Un sommet  $s_j$  de  $P$  est dit concave et noté par  $y$  si  $A(s_j) > \pi$

Definition : On définit la caractéristique angulaire d'un polygone  $P$  à  $n$  sommets par :

$$AC(P) = a_1 \dots a_n \in \{x, y\}^+$$

tel que :

- $a_i = x$  si  $A(s_i) < \pi$
- $a_i = y$  si  $A(s_i) > \pi$

Remarque : Par soucis de simplicité, nous considérons dans la suite qu'il n'existe pas dans le polygone  $P$  trois sommets qui soient alignés.

L'algorithmme de triangularisation d'un polygone simple quelconque, présenté dans ce papier, est basé sur l'idée suivante : si on sait comment triangulariser un type donné de polygone, alors la triangularisation d'un polygone simple quelconque se trouverait accélérée en isolant dans le polygone des sous-polygones de la forme donnée.

Par exemple, un quadrilatère est un polygone dont la décomposition en triangles est une opération simple et facile à réaliser, par conséquent, la recherche de quadrilatères dans le polygone simplifierait notablement sa triangularisation. De même, et à un degré supérieur, la recherche de polygones convexes dans un polygone quelconque accélérerait sa triangularisation.

L'avantage offert par les polygones convexes par rapport aux quadrilatères, dans ce cas, est qu'un quadrilatère est une figure géométrique à caractéristique en nombre de sommets fixe et connue ; contrairement aux polygones convexes, dont la seule contrainte est que tous les sommets, quel que soit leur nombre, soient convexes. Ce qui nous confère un degré de liberté plus grand.

A noter que cette idée a déjà été exploitée dans [FOUR 84], ainsi que dans [CHAZ 84], qui passent d'abord par une trapézoïdation du polygone avant de réaliser effectivement sa triangularisation.

Cependant, le passage par des polygones convexes par exemple pose le problème de minimisa-



tion du nombre des sous-polygones déterminés ou isolés, problème dû surtout à la présence de concavités dans le polygone.

Pour cette raison, en remarquant que tout polygone peut être caractérisé par sa succession de chaînes de sommets convexes et de sommets concaves, nous avons pensé à inclure dans notre recherche les polygones en spirale et les polygones en « T ». En effet, d'une part la recherche de polygones en spirale, en plus des polygones convexes, permet de résoudre, ou du moins de réduire le problème posé par les concavités — vu qu'un polygone en spirale possède une chaîne de sommets concaves ; d'autre part, en présence de plusieurs chaînes de sommets concaves, les polygones en T sont plus faciles à isoler qu'un polygone convexe ou en spirale et permettent de réduire de manière significative le nombre de sommets concaves du polygone initial.

Avant de développer l'algorithme final de décomposition d'un polygone simple quelconque, nous allons voir la triangularisation de ces trois types de polygones particuliers : Convexe, Spirale et en T.

### 3 Triangularisation d'un polygone convexe

La classe des polygones convexes est une classe dont la triangularisation est la plus simple, car ce type de polygones ne présente aucune particularité en ses sommets qui sont tous de même type  $x$ .

En adoptant les conventions de la théorie des langages, soit  $L_c$  la classe des caractéristiques angulaires des polygones convexes, alors :

$$L_c \subset \{x\}^*$$

Or, on sait en géométrie plane que tout polygone possède un nombre fini de sommets et au moins 3 sommets convexes.

Par conséquent, on peut affirmer que :

$$L_c = x^3\{x\}^m \quad \text{avec } m \in \mathbb{N} \setminus \{\infty\}$$

On peut décomposer de plusieurs manières un polygone convexe en triangles. La méthode proposée ici est assez simple et permet de joindre, à partir d'un sommet quelconque  $s_i$  de  $P$ , les  $(n-3)$  sommets non-adjacents à  $s_i$  (fig 2) et de former par cela  $(n-2)$  triangles  $T_j$ ,  $j = 1, \dots, n-2$  tel que :

$$T_j = (s_i s_{i+j} s_{i+j+1})$$

où les indices  $i, i+j, i+j+1$  sont calculés modulo  $(n)$

L' algorithme de triangularisation est le suivant :

En Entrée :  $P$  polygone convexe ,  $s_i$  un sommet de  $P$

En Sortie : ensemble de triangles

Procédure  $C$  ( $P$  : polygone,  $s_i$  : sommet ) ;

debut

$j := 1$  ;

Tant que  $AC(P) <> xxx$

Faire

sortir-Triangle ( $s_i, s_{i+j}, s_{i+j+1}$ ) ;

$P := P - \{s_{i+j}\}$  ;

$j := j+1$

Fait ;

sortir-Triangle ( $s_i, s_{i+j}, s_{i+j+1}$ )

Fin ;

La complexité temporelle de cette algorithme étant de  $O(n)$ , on peut alors énoncer le théorème suivant, la preuve étant évidente :

Théorème 1 : Un polygone convexe à  $n$  sommets est triangularisable en  $O(n)$

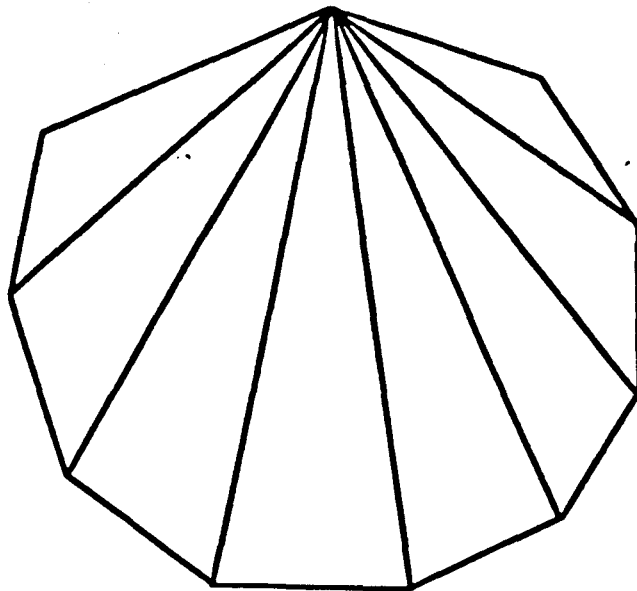


Fig 2 — Exemple de triangularisation d'un polygone convexe

#### 4 Triangularisation d'un polygone spirale

Un polygone spirale est défini comme étant un polygone possédant une et une seule chaîne de sommets concaves, reliée de bout en bout par une chaîne de sommets convexes (voir fig 1.b) . Par conséquent, la classe des polygones en spirale est une classe de polygones dont la triangularisation est rendue simple en considérant sa chaîne concave.

En effet, on remarque que le dernier ( resp. le premier ) sommet  $y$  de la chaîne concave est visible des deux sommets successeurs ( rep. prédécesseurs ) sans intersection possible avec un segment quelconque du polygone . En formant un triangle à l'aide de ces trois sommets, on diminue ainsi la valeur angulaire du dernier ( resp. premier ) sommet de la chaîne concave dans le polygone restant. Ainsi, en répétant cette opération jusqu'à ce que ce sommet concave devienne convexe, on aura d'une part éliminé un sommet concave, et d'autre part le polygone restant est lui aussi spirale — dans le cas où on a plus de deux sommets concaves . Le résultat final étant obtenu en itérant ce processus sur tous les sommets de la chaîne concave du polygone initial ( fig 3 ) .

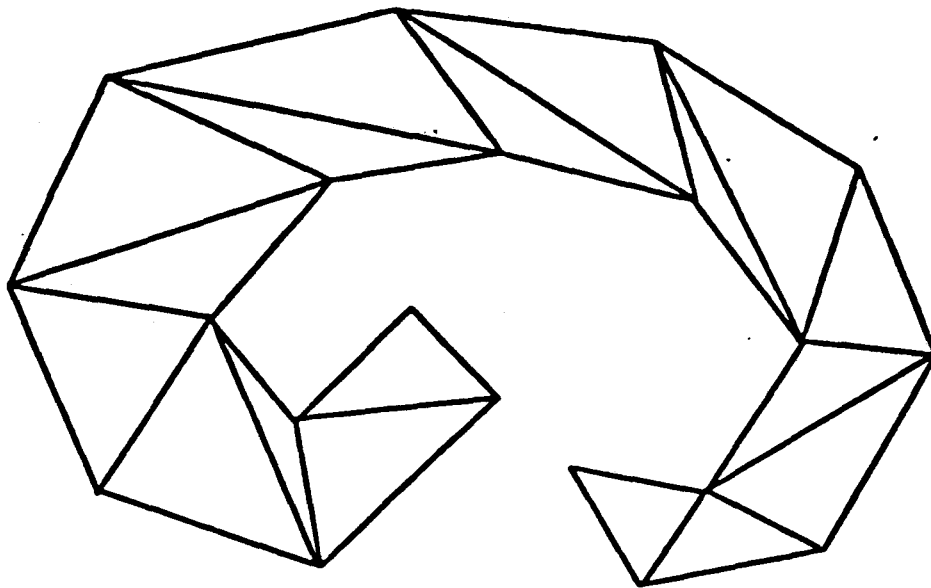


Fig3 — Triangularisation d'un polygone spirale

A noter que le dernier polygone en spirale restant après élimination successive de tous les autres sommets concaves, ne possède qu'une seule concavité . On dira alors qu'il est simplement spiralé . Dans ce cas, la triangularisation est achevée en décomposant ce polygone à l'aide de l'algorithme de triangularisation d'un polygone convexe vu précédemment, en prenant soin de le faire à partir de ce seul sommet concave . Ce sommet sera de ce fait relié à tous les sommets  $x$  restant non-adjacents . Afin de formaliser ce raisonnement, soit  $L_s$  l'ensemble des caractéristiques angulaires de tous les polygones en spirales. On a alors :

$$L_s \subset \{x\}^* \{y\}^*$$

Etant donné que tout polygone possède un nombre fini de sommets, dont au moins trois sommets convexes, et qu'un polygone spirale ne possède qu'une seule chaîne de sommets concaves . En arrangeant le polygone de manière que le premier sommet soit le sommet convexe précédent la chaîne concave, l'ensemble  $L_s$  peut s'écrire :

$$L_s = x \{y\}^m x x \{x\}^p \text{ avec } m, p \in \mathbb{N} \setminus \{\infty\}$$

On voit dès lors, que l'ensemble des caractéristiques angulaires des polygones convexes est un sous-ensemble de  $L_s$ , i.e.,

$$L_c \subset L_s$$

Ce choix est justifié dans le sens où un polygone simplement spiralé se décompose en triangles de la même manière qu'un polygone convexe, grâce à l'algorithme donné en 3.

Soit  $A_p$  un mot de  $L_s$  :

$$A_p = x y \dots y x \dots x$$

correspondant au polygone  $P$  à  $n$  sommets suivant :

$$P = s_1 s_2 \dots s_{m+1} s_{m+2} \dots s_n$$

Ayant :

$$m \geq 2 \text{ sommets concaves : } s_2, \dots, s_{m+1}$$

$$p \geq 3 \text{ sommets convexes : } s_1, s_{m+2}, \dots, s_n$$

Au fur et à mesure de la production de triangles à partir du sommet  $s_{m+1}$  on diminue la valeur angulaire de ce sommet. Lorsque  $A(s_{m+1})$  devient inférieur à  $\pi$ , la caractéristique angulaire du polygone restant  $R(P)$  est alors :

$$A_{r(p)} = x y \dots x x \dots x$$

Ayant :

$$(m-1) \text{ sommets } y : s_2 \dots s_m$$

$$(p-r) \text{ sommets } x : s_1 s_{m+1} s_{m+r+2} \dots s_n$$

où  $r$ , représente le nombre de triangles obtenus à partir de  $s_{m+1}$

On voit alors que :  $A_{r(p)} \in L_s$

on peut alors refaire le même traitement avec ce polygone.

Remarque : Si  $m = 1$ , le sommet concave correspondant ( $s_2$ ) est alors visible de tous les

autres sommets du polygone , et dans ce cas on peut appliquer l'algorithme de décomposition en triangles des polygones convexes.

L'algorithme de triangulation d'un polygone en spirale est le suivant :

En Entrée : P polygone spirale , s dernier sommet concave.

En Sortie : ensemble de triangles.

Procédure S ( P : polygone , s : sommet ) ;

Debut

si P = polygone convexe ou simplement spiralé

alors C ( P , s )

Sinon

Tant que A ( s ) >  $\pi$

Faire

a = Succ ( s ) ; b = Succ ( a ) ;

Sortir-Triangle( s , a , b ) ;

Succ ( s ) = b ; Pred ( b ) = s ;

P = P - {a}

Fait ;

s = Pred ( s ) ;

S ( P , s )

Fsi

Fin ;

La complexité temporelle de cette algorithme est de  $O(n)$  , on peut alors énoncer le théorème suivant :

Théorème 2 : Un polygone spirale à n sommets est triangulisable en  $O(n)$

Preuve : montrons, en utilisant l'algorithme S, que l'on obtient bien  $(n-2)$  triangles.

soit P un polygone spirale possédant:

m sommets concaves

p sommets convexes

soit  $t_i$  le nombre de sommets convexes de P éliminés du  $i$ -ième appel de S

soit  $r_i$  le nombre de triangles obtenus au  $i$ -ième appel de S

On a :

$r_1 = t_1$  : étant donné qu'au premier appel à S tout triangle obtenu permet d'éliminer un sommet convexe

$r_i = t_i + 1$  : étant donné qu'à l'appel suivant le premier triangle obtenu permet d'éliminer un sommet concave

Par conséquent :

Le nombre de triangles obtenus après  $(m-1)$  appels de S est :

$$\sum_{i=1}^{m-1} r_i = m - 2 + \sum_{i=1}^{m-1} t_i$$

Le polygone simplement spiralé restant possède un nombre de sommets égal à :

$$p + 2 - \sum_{i=1}^{m-1} t_i$$

et donc est triangularisé en :

$$p - \sum_{i=1}^{m-1} t_i \quad \text{Triangles}$$

D'où le nombre total de triangles obtenu est :

$$m - 2 + \sum_{i=1}^{m-1} t_i + p - \sum_{i=1}^{m-1} t_i = m + p - 2 = n - 2$$

D'autre part, l'algorithme proposé possède une complexité temporelle de  $O(n)$ .

Donc on a bien le résultat voulu.

## 5 Triangularisation d'un polygone en « T »

Un polygone en T (voir fig 1.c) est un polygone possédant deux chaînes de sommets concaves tel que :

- . le dernier sommet concave de la première chaîne est relié au premier sommet concave de la seconde chaîne par 1 ou 2 sommets convexes
- . et le dernier sommet concave de la seconde chaîne est relié au premier sommet concave de la première chaîne par une chaîne de sommets convexes en nombre quelconque.

Soit  $L_T$  l'ensemble des caractéristique angulaires des polygones en T. Alors :

$$L_T \subset \{x\}^* \{y\}^* x \{y\}^* \cup \{x\}^* \{y\}^* x x \{y\}^*$$

Etant donné qu'on ne considère que des polygones, et en réarrangeant celui-ci de manière que le premier sommet soit le sommet convexe précédant la première chaîne de sommets concaves, on a alors :

$$L_T = x \{y\}^i x \{y\}^j x \{x\}^k \cup x \{y\}^i x x \{y\}^j \{x\}^k$$

Avec  $i, j, k \in \mathbb{N} \setminus \{\infty\}$

d'où, on a :

$$L_c \subset L_s \subset L_T$$

Soit  $A_p$  un mot de longueur n de  $L_T$

$$A_p = x y \dots y x x y \dots y x \dots x$$

correspondant au polygone suivant :

$$P = s_1 s_2 \dots s_{i+1} s_{i+2} s_{i+3} s_{i+4} \dots s_k s_{k+1} \dots s_n$$

Afin de triangulariser P on considère le dernier sommet concave de la première chaîne,  $s_{i+1}$ , et le premier sommet concave de la seconde chaîne,  $s_{i+4}$ . On peut avoir deux cas:

cas 1 : (fig 4.a)

si,  $s_{i+1}$ , est visible sans intersection possible avec tout segment de P, de  $s_{i+4}$ ; alors les 4 sommets :  $s_{i+1}$ ,  $s_{i+2}$ ,  $s_{i+3}$ ,  $s_{i+4}$ , forment un quadrilatère qui est facilement décomposable



en 2 triangles . Le polygone restant  $R(P)$  peut avoir l'un des 4 cas de figures suivantes , dépendant des nouvelles valeurs angulaires de  $s_{i+1}$  et  $s_{i+4}$  :

- Soit  $A(s_{i+1}) > \pi$  et  $A(s_{i+4}) > \pi$   
 dans ce cas  $R(P)$  est en spirale , donc  $A_{R(P)} \in L_p$ .  
 $R(P)$  peut être alors triangularisé par  $S(R(P), s_2)$
- Soit  $A(s_{i+1}) > \pi$  et  $A(s_{i+4}) < \pi$   
 Dans ce cas :  
 si  $A(s_{i+5}) > \pi$  alors  $A_{R(P)} \in L_T$ .  
 On refait alors la même opération avec les sommets  $s_{i+1}, s_{i+5}$ .  
 sinon,  $A_{R(P)} \in L_s$ , d'où  $R(P)$  peut être triangularisé par  $S(R(P), s_2)$ .
- Soit  $A(s_{i+1}) < \pi$  et  $A(s_{i+4}) > \pi$   
 dans ce cas :  
 si  $A(s_i) > \pi$  alors  $A_{R(P)} \in L_T$   
 sinon  $A_{R(P)} \in L_s$  et la triangularisation est achevée par  $S(R(P), s_{i+4})$
- Soit  $A(s_{i+1}) < \pi$  et  $A(s_{i+4}) < \pi$   
 dans ce cas :
  - si  $A(s_i) < \pi$  et  $A(s_{i+5}) < \pi$   
 alors  $R(P)$  est convexe et la triangularisation est obtenue par  $C(R(P), s_{i+1})$
  - si  $A(s_i) < \pi$  et  $A(s_{i+5}) > \pi$   
 ou  $A(s_i) > \pi$  et  $A(s_{i+5}) < \pi$   
 alors  $A_{R(P)} \in L_s$  et la triangularisation est obtenue par  $S(R(P), s_{i+5})$  ou par  $S(R(P), s_i)$
  - si  $A(s_i) > \pi$  et  $A(s_{i+5}) > \pi$   
 alors  $A_{R(P)} \in L_T$  et on refait le même raisonnement

Remarque : Lorsqu'on n'a qu'un seul sommet convexe séparant deux chaînes, les deux sommets concaves adjacents sont visibles entres eux et peuvent être reliés sans intersection avec un autre segment de P (fig 4.b).

Cas 2 : (fig 4.c)

si  $s_{i+1}$  n'est pas visible de  $s_{i+4}$  alors :

- soit  $s_{i+1}$  est visible de  $s_{i+3}$
- soit  $s_{i+4}$  est visible de  $s_{i+2}$

Dans les deux cas , on peut former un triangle et le polygone restant  $R(P)$  est tel que :

$$A_{T(p)} \in L_T$$

Par conséquent on peut refaire le même traitement avec  $R(P)$

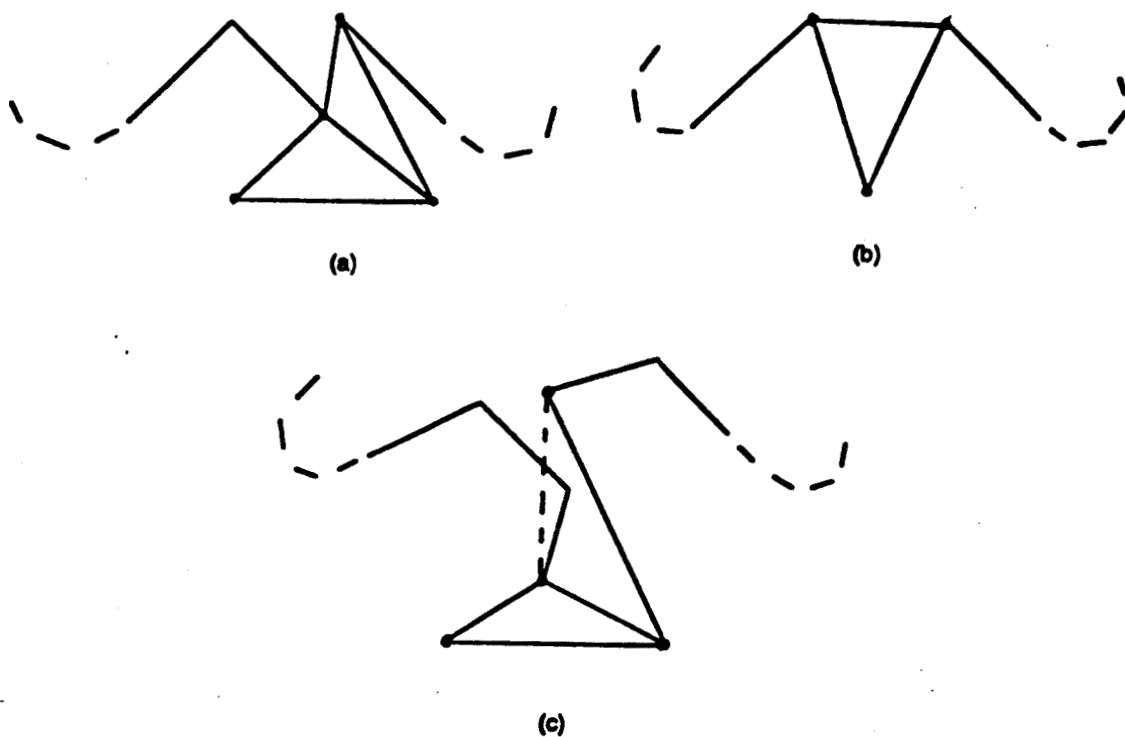


Fig 4— Triangularisation partielle d'un polygone en T

L' algorithme de triangularisation est le suivant :

En Entrée :

- P polygone en T
- d dernier sommet concave de la première chaîne
- p premier sommet de la deuxième chaîne

En Sortie : ensemble de  $( n - 2 )$  triangles

**Procédure**  $T(P, d, p)$ ;

**Debut**

**Si**  $P =$  polygone convexe **ou** spirale

**Alors**  $S(P, d)$

**Sinon**

**Si**  $\text{Test-seg}(p, d)$

**Alors**

triangulariser le sous-polygone obtenu par  $S$

$\text{Succ}(d) = p$  ;  $\text{Pred}(p) = d$

**Sinon**

**Si**  $\text{Test-seg}(d, \text{Pred}(p))$

**Alors**

$\text{Sortir-Triangle}(d, \text{Succ}(d), \text{Pred}(p))$ ;

$\text{Succ}(d) = \text{Pred}(p)$ ;  $\text{Pred}(\text{Pred}(p)) = d$

**Sinon**

$\text{Sortir-Triangle}(p, \text{Pred}(p), \text{Succ}(d))$  ;

$\text{Pred}(p) = \text{Succ}(d)$ ;  $\text{Succ}(\text{Succ}(d)) = p$

**Fsi**

**Fsi**;

**Si**  $A(p) < \pi$  **Alors**  $p = \text{Succ}(p)$  **Fsi**;

**Si**  $A(p) < \pi$  **Alors**  $d = \text{Pred}(d)$  **Fsi**;

$T(R(P), d, p)$

**Fsi**

**Fin**;

**Remarques** :

- 1— si  $P$  est spirale, alors :  
     $p$  désigne le premier sommet de la chaîne  
     $d$  désigne le dernier sommet de la chaîne
- 2— la fonction  $\text{Test-seg}(*, *)$  peut être réalisée en ne considérant que l'angle formé par les deux paramètres et leurs successeurs respectifs (voir Fig 4).

**Proposition 1** : un polygone en  $T$  de  $n$  sommets est triangularisable en  $O(n + r)$ .  
où  $r$  représente le nombre de sommets constituant le « tronc » du polygone.

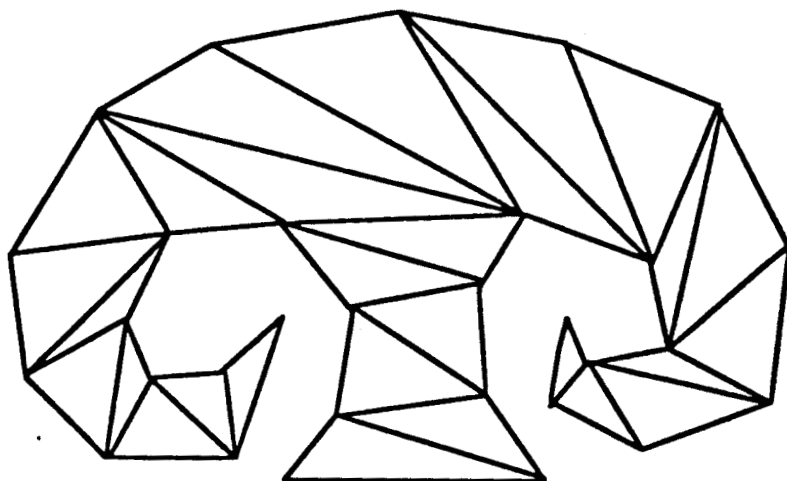
**Preuve** : soit  $t$  le nombre de triangles nécessaire pour triangulariser le tronc du polygone .  
Comme, tout triangle obtenu, dans l'algorithme, permet d'éliminer un sommet de  $P$  . Par conséquent, le polygone restant possède  $(n - t)$  sommets. Ce polygone comme on l'a vu est soit spirale soit convexe , et donc est triangularisé en  $(n - t - 2)$  triangles par  $S$  ou  $C$  .

De plus on remarque que :  $t = r - 2$  , où  $r$  représente le nombre de sommets constituant le tronc du polygone.

D'où, le nombre total de triangles obtenus par  $T$  est:

$$t + n - t - 2 = n - 2$$

Un exemple de résultat de l'application de cet algorithme à un polygone en  $T$  est donné par la figure 5.



*Fig 5— Triangularisation d'un polygone en T*

Après avoir vu la triangularisation des polygones convexes , spirales et en T. Nous allons voir comment, à l'aide de ces trois types de polygones, triangulariser un polygone simple quelconque.

Etant donné que les sommets concaves réduisent la visibilité de certains sommets à d'autres sommets du polygone . Il est alors naturel de considérer la triangularisation à partir de ces sommets .

Comme nous l'avons dit, pour triangulariser un polygone simple quelconque on va dans une première étape réaliser la décomposition en polygones convexes , spirales et en T. On utilisera par la suite les différents algorithmes vus précédemment, pour triangulariser les sous-polygones obtenus, et achever ainsi la triangularisation du polygone considéré .

Cependant, par les contraintes strictes relatives à la définition des polygones convexes et spirales, il est pour le moins difficile d'isoler ces derniers . Par contre, il est évident qu'on peut plus facilement isoler les polygones en T , particulièrement en présence de plusieurs chaînes de sommets concaves. Il suffit pour cela de pouvoir relier deux chaînes de sommets concaves consécutives . D'autre part, la triangularisation d'un polygone en T utilise systématiquement la triangularisation des polygones convexes et spirales .

Par conséquent, on recherchera plutôt les polygones en T pour la triangularisation du polygone simple.

Dans ce cas, soit  $c_1$  et  $c_2$  deux chaînes de sommets concaves consécutives dans P. S'il existe deux sommets,  $s_i$  de  $c_1$  et  $s_j$  de  $c_2$ , tel que le segment  $( s_i , s_j )$  soit complètement inclus dans P alors :

- le sous-polygone obtenu en reliant  $s_i$  à  $s_j$  est soit convexe, spirale ou en T
- le polygone restant ,  $R(P)$  , possède moins de sommets concaves

En effet, soit  $A_q$  la caractéristique angulaire du sous-polygone Q déterminé en reliant  $s_i$  à  $s_j$  :

$$A_q = A(s_i) y \dots y x \dots x y \dots y A(s_j)$$

On voit alors que, quelles que soient les valeurs de  $A(s_i)$  et de  $A(s_j)$  , la caractéristique angulaire de Q est tel que :

$$A_q \in L_T$$

Et Q est :

- convexe si  $A(s_i) < \pi$  et  $A(s_j) < \pi$ , et tel que la chaîne de sommets reliant  $s_i$  à  $s_j$  est convexe.
- spirale si  $A(s_i) > \pi$  et  $A(s_j) > \pi$

- en T dans les autres cas

Cependant, la recherche de  $s_i$  et  $s_j$  peut être une opération onéreuse si le nombre de sommets concaves composant les chaînes  $c_1$  et  $c_2$  est important.

Pour cela, on caractérisera chaque chaîne de P, en plus des sommets de début et de fin de chaîne, par les quatre sommets suivants :

- le sommet concave de la chaîne le plus haut : ph
- le sommet concave de la chaîne le plus bas : pb
- le sommet concave de la chaîne le plus à gauche : pg
- le sommet concave de la chaîne le plus à droite : pd

Ces sommets seront donnés en respectant le sens de parcours de la chaîne (ici le sens des aiguilles d'une montre).

Par conséquent, la recherche des deux sommets reliant les deux chaînes concaves,  $c_1$  et  $c_2$ , se fera en consultant les sommets caractéristiques de chaque chaîne. Cette recherche devant être réalisée de manière à englober le maximum de sommets concaves des deux chaînes dans le sous-polygone. Ces deux sommets doivent être alors, les premiers sommets caractéristiques visible de chaque chaîne.

Dans notre algorithme on utilisera les fonctions suivantes :

- Sommet-Charact ( \* ), qui permet de déterminer, à partir du sommet de début de chaîne, les sommets caractéristiques restitués dans l'ordre de leur apparition.
- Consult ( \* , \* ), fonction à résultat booléen qui teste la visibilité des sommets caractéristiques de la seconde chaîne aux sommets de la première chaîne.
- Succ-chain ( \* ), permet de donner le sommet de début de la chaîne concave suivante.

L' algorithme de triangularisation est le suivant :

En Entrée :

- P polygone simple quelconque .
- deb1 sommet de debut de la 1<sup>ière</sup> chaîne
- deb2 sommet de debut de la 2<sup>ième</sup> chaîne

En Sortie : ensemble de ( n - 2 ) triangles

Procédure Decomp ( P , deb1 , deb2 )

Debut

Si AC(P)  $\in$  L<sub>T</sub>

Alors

p = premier sommet concave de la 1<sup>ière</sup> chaîne;

d = dernier sommet concave de la 2<sup>ième</sup> chaîne;

T(P , d , p)

Sinon

c<sub>1</sub> = Sommets-caract( deb1 );

c<sub>2</sub> = Sommets-caract( deb2 );

Si Consult( c<sub>1</sub> , c<sub>2</sub> )

Alors

p = premier sommet concave de la 1<sup>ière</sup> chaîne du sous-polygone Q;

d = dernier sommet concave de la 2<sup>ième</sup> chaîne du sous-polygone Q;

T(Q , d , p);

deb1 = Succ-chain( deb2 ); deb2 = Succ-chain( deb1 )

Sinon

deb1 = deb2;

deb2 = Succ-chain( deb2 )

Fsi ;

Decomp( R(P) , deb1 , deb2 )

Fsi

Fin ;

Remarques :

- . R(P) désigne le polygone restant après une éventuelle triangularisation partielle.
- . Q désigne le sous-polygone en T déterminé.

Cet algorithme permet de triangulariser un polygone simple quelconque en  $(n-2)$  triangles.

En effet, soit  $s$  le nombre de sous-polygones en  $T$  obtenus en décomposant  $P$ ,  
et soit  $n_i$   $i = 1, \dots, s$  le nombre de sommets de chaque sous-polygone.

Tout sous-polygone étant triangularisé en  $(n_i - 2)$  triangles, par l'un des algorithmes donnés précédemment.

Par conséquent, le nombre total de triangles obtenus est alors :

$$\sum_{i=1}^s (n_i - 2) = \sum_{i=1}^s n_i - 2s$$

Or, en remarquant que les deux sommets concaves de  $P$  permettant de relier les deux chaînes de sommets concaves consécutives, se retrouve dans le polygone restant  $R(P)$  et dans le sous-polygone déterminé.

On peut alors en déduire le nombre  $n_s$  de sommets du dernier sous-polygone :

$$n_s = n - \sum_{i=1}^{s-1} (n_i - 2) = n + 2(s-1) - \sum_{i=1}^{s-1} n_i$$

D'où, en reportant ce résultat dans la première équation, le nombre total de triangles obtenus est alors :

$$\sum_{i=1}^s n_i - 2s = n + 2(s-1) - 2s = n - 2$$

Un exemple de résultat obtenu par l'application de cet algorithme est donné par la fig 6.



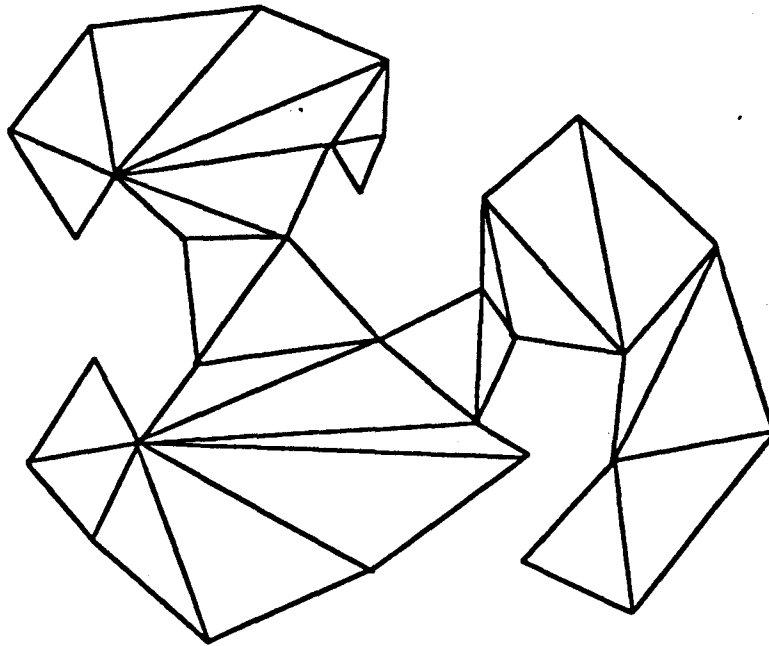


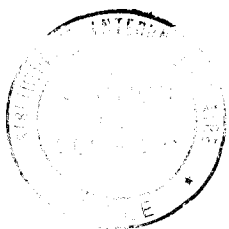
Fig 6— Exemple de triangularisation d'un polygone simple quelconque

## 7 Conclusion

L' algorithme de triangularisation d'un polygone simple quelconque proposé, a été implémenté et testé sur une station graphique relié au SPS7 , à l'aide de GDS ( Graphics Data Syntax for a multiple workstation interface ).

Les résultats pratiques obtenus sont très concluant et montrent que l'algorithme est quasi-linéaire pour un grand nombre de polygones. De plus, l'algorithme est particulièrement adapté aux polygones dont les chaînes de sommets concaves sont visibles deux à deux.

L'avantage offert par cet algorithme est qu'une implémentation en mode pipe-line est possible, en séparant la recherche des polygones en T et leur triangularisation. Ces deux opérations pouvant être alors menées en parallèle , ce qui accélérerait la convergence de l'algorithme.



Une manière classique et conventionnelle de représenter une image numérique est de la considérer comme une juxtaposition, dans une grille cartésienne, de points images appelés pixels (pour picture elements).

Nous proposons dans ce cadre, un nouveau modèle de représentation d'images basé non plus sur une description pixel, mais plutôt sur une description globale et macroscopique de celles-ci.

Le but recherché étant alors d'offrir à l'utilisateur un outil général permettant de décrire de manière simple, globale et concise tout objet graphique, atténuant ainsi les redondances et les imperfections du modèle classique.

Dans un premier temps, nous définissons de nouveaux objets élémentaires appelés "Macro-pixels", dans leurs aspects morphologique et visuel. Ensuite, nous donnons un modèle de description d'objets basé sur ces éléments, et définissons les opérateurs nécessaires à la construction de ces objets à partir des macro-pixels. Enfin, nous proposons une implémentation matérielle possible de tels objets dans le cadre d'une "architecture-objets".

Mots clés : Synthèse d'image, traitement d'image, modélisation, représentation pixel, décomposition, système graphique parallèle.