84/365

Sc N 84/386 A

UNIVERSITÉ DE NANCY I

FACULTÉ DES SCIENCES

MEDECIM :
UN SYSTÈME DE DESCRIPTION ET DE MANIPULATION D'IMAGES APPLIQUÉ A L'IMAGERIE MÉDICALE

BIBLIOTHEQUE * UNIVERSITE NANCY SCIENCES

THÈSE DE DOCTORAT DE 3e CYCLE
INFORMATIQUE

SOUTENANCE PUBLIQUE LE 17 DÉCEMBRE 1984

Bernard CURIEN

MEMBRES DU JURY

Président : M. J.P. HATON, Professeur
Juges : M. A. BELAID, Attaché de Recherche
M. P. BERNADAC, Professeur
M. R. MOHR, Professeur
M. J. ROBERT, Professeur

BIBLIOTHEQUE SCIENCES NANCY 1

D 095 179366 2

NIVERSITÉ DE NANCY I

FACULTÉ DES SCIENCES

MEDECIM :
UN SYSTÈME DE DESCRIPTION ET DE MANIPULATION D'IMAGES APPLIQUÉ A L'IMAGERIE MÉDICALE

THÈSE DE DOCTORAT DE 3e CYCLE
INFORMATIQUE

SOUTENANCE PUBLIQUE LE 17 DÉCEMBRE 1984

Bernard CURIEN

MEMBRES DU JURY

Président : M. J.P. HATON, Professeur
Juges : M. A. BELAID, Attaché de Recherche
M. P. BERNADAC, Professeur
M. R. MOHR, Professeur
M. J. ROBERT, Professeur

SOMMAIRE

MEDECIM

Un système de description et de manipulation d'images appliqué à l'imagerie médicale

INTRODUCTION

INTRODUCTION

L'image est par nature un véhicule privilégié de l'information. Ce constat repose d'abord sur des notions de physiologie, puisque la vision est une fonction du système nerveux particulièrement sophistiquée, mais il prend aujourd'hui une autre dimension du fait des progrès de l'informatique et de la télématique et on parle couramment de "civilisation de l'image" (Rev 3).

Après l'ère audiovisuelle qui nous a apporté la télévision et le magnétoscope, véhicules passifs de l'image, nous découvrons de nouveaux aspects relevant parfois d'améliorations techniques mais surtout de changements de nature avec l'image numérique , l'image tridimensionnelle, la synthèse d'images et l'image interactive. Cette dernière, mariant l'informatique et la télématique, fait de l'image un service qui devrait etre un des piliers de notre future société de communication.

Certaines de ces "nouveautés" sont connues depuis longtemps (l'holographie fut découverte il y a 35 ans) mais les applications quittent maintenant le domaine restreint dans lequel elles étaient confinées pour devenir des composantes de la vie économique (ainsi le visiophone et la téléconférence sont aujourd'hui des réalités).

S'agissant des images numériques les exemples abondent qui illustrent leur impact sur de nombreuses activités :

- L'industrie s'équipe en systèmes de vision servant dans les domaines de la métrologie, l'inspection et l'assemblage (FER 84), (CHE 83). On estime que le marché de la visionique, branche de la robotique, devrait voir son chiffre d'affaires multpiplié par 5 entre 1982 et 1985 atteignant alors 500 M$ en Amérique du Nord

- L'industrie cinématographique utilise désormais les images de synthèse (Rev 2). Toutefois les considérables moyens de calcul requis sont encore un frein.

- La médecine, grosse consommatrice d'images de longue date, utilise également les images numériques dans des applications comme le scanner X, la scintigraphie ou l'angiographie numérisée. A cet égard le développement d'une nouvelle structure, les départements d'imagerie, fédérant des disciplines distinctes comme la radiologie, la médecine nucléaire et des techniques assimilées comme l'échographie et la résonance magnétique nucléaire montre la place accordée à l'image (Rev 1), (Rev 3), (COM 84),(CAS 84).

Ces nouvelles images suscitent de nombreux travaux pour développer les outils de manipulation adaptés. Nous avons besoin de matériels nouveaux, de logiciels de traitement d'images et de logiciels de gestion de banques d'images.
Ainsi un service d'images utilisant une approche interactive comme celui développé par l'Architecture Machine Group du MIT demande une capacité de stockage et une rapidité de lecture qui étaient jusqu'il y a peu inacessibles (BOL 83).
De meme le traitement d'images numériques en temps réel pour des applications comme la télévison numérique ou l'inspection impose des vitesses de transfert des données et de calcul élevées.
Outre les problèmes de traitement proprement dit, se posent les problèmes de gestion et d'accès aux images et aux informations qu'elles véhiculent : ainsi les images LANDSAT du fait de leur nombre posent actuellement de gros problèmes de stockage et il est impensable dans l'immédiat de toutes les dépouiller. On trouve dans ce domaine des travaux visant à appliquer les systèmes experts à la gestion de banques d'images comme le projet EXPRIM développé au CRIN (Centre de Recherche en Informatique de NANCY) (CRE 83)

Ayant défini le contexte dans lequel nous souhaitons situer notre travail, nous rappellerons que de part notre situation de scientifique intégré au sein d'un service de Médecine Nucléaire doublé d'un laboratoire universitaire de Biophysique nous avons été naturellement sensibilisé au double problème, médical d'abord, de l'interprétation des images scintigraphiques, informatique ensuite, de leur traitement. C'est ainsi qu'est née l'idée d'un langage de manipulation d'images qui faciliterait le traitement puis l'interprétation des examens réalisés dans les services d'imagerie. La première étape de ce projet consiste à se doter d'outils de manipulation des images et c'est elle qui fait donc l'objet de notre thèse.

Le rapport de présentation qui suit comporte dans une première partie un rappel de la physiologie de la vision assorti d'un parallèle avec certaines tendances actuelles du traitement d'images. Une seconde partie présente les problèmes de la vision par ordinateur et envisage le cas des images médicales. Une troisième partie expose les principes et la réalisation du langage de description d'images que nous avons mis au point. Enfin la quatrième partie est prospective, envisageant des améliorations et l'intégration de ce langage dans un système visant à l'interprétation des images.

LA VISION

Le cerveau et l'esprit ont en commun ... l'organisation

L'organisation des sytèmes matériels est elle-meme immatérielle ... ni dimensionable ... ni réductible à de la masse ou de l'énergie

Edgar MORIN

NOTIONS DE NEUROPHYSIOLOGIE DE LA VISION

Ainsi que nous l'avons remarqué en introduction, la place de l'image se justifie essentiellement par les performances de l'oeil et du système de la vision humaine.

L'oeil est un moyen de préhension capable de manipuler des données en quantités énormes et l'ensemble du mécanisme de la vision concourt à en extraire les informations utiles à très grande vitesse. Comparativement nos autres organes sensoriels sont certainement très inférieurs par la quantité des informations qu'ils appréhendent et par la vitesse à laquelle ils les traitent.

Les outils que nous construisons pour traiter artificiellement les images gagneront en efficacité en imitant l'organisation de la vision et c'est pourquoi nous pensons qu'il est utile d'évoquer dans ce travail la neurophysiologie de la vision, domaine difficile à explorer et imparfaitement connu - <LEG 59>, <RUC 79>, <IMB 83>.

Toutefois il nous est aujourd'hui impossible d'appréhender les mécanismes de l'activité cognitive visuelle. Une réflexion d'Edgar MORIN illustre le problème : "L'esprit ne sait rien, par lui-meme, du cerveau qui le produit, lequel ne sait rien de l'esprit qu'il produit" - <MOR 84> -.

Nous limiterons notre approche à quelques idées, cherchant un parallèlisme entre le fonctionnement de la vision et la structure des systèmes informatiques de traitement d'images et d'analyse de scène.

UN RECEPTEUR PRIVILEGIE

Du point de vue de l'embryologie et de la physiologie le système de la vision se distingue des autres systèmes sensoriels par le fait que le récepteur, la rétine, se développe directement à partir de l'ébauche nerveuse elle-meme ce qui signifie qu'une partie du cerveau reçoit directement les impressions visuelles; cette observation laisse pressentir la qualité et l'acuité des perceptions visuelles comparées à celles d'autres sens comme l'ouie.

UN PARTAGE DES TACHES ET UNE SYNERGIE

Classiquement on décrit les cellules à cones et à batonnets, récepteurs spécialisés analysant, les uns l'information de luminance du stimulus lumineux , et les autres le signal de chrominance. Cette discrimination précoce des informations a servi notamment à établir des modèles de la vision des couleurs utilisant un signal achromatique et deux signaux chromatiques (FAU 76), (CLA 83.2).

Chacun des neurones impliqué dans le processus de la vision posséde au niveau de la rétine un champ récepteur; on distingue deux types de neurones dits à centre "ON" et à centre "OFF". Les uns et les autres ne sont sensibles qu'aux variations spatiales du stimulus lumineux et transmettent donc une information ne concernant pas la valeur absolue du signal mais sa dérivée (loi de WEBER). Au niveau cortical on trouve aussi des neurones sensibles au gradient de l'intensité lumineuse mais ne réagissant que pour une orientation de ce gradient.

On a également identifié des neurones donnant à un stimulus lumineux une reponse de type "tonique", stable, et d'autres donnant une réponse de type "phasique", transitoire. Ici ce sont les aspects statiques et dynamiques de la scène qui sont appréhendés; la périphérie de la rétine est plus particulierement riche en neurones "phasiques" (répondant bien au mouvement) et la fovéa où se projette l'image du point fixé est plus riche en neurones "toniques"

La vision binoculaire, responsable de la vision du relief, suppose une correspondance parfaite de la partie temporale de la rétine de l'oeil droit et de la partie nasale de la rétine de l'oeil gauche et inversement puisque ces régions recoivent les stimulus issus de la meme partie du champ visuel. La diplopie apparait lorsque cette correspondance fait défaut; on obtient alors deux images non superposées d'un objet unique. On constate donc l'existence d'une synergie mécanique et d'une relation biunivoque entre les points des deux rétines droite et gauche pour assurer la fusion des deux images.

Ces exemples montrent que la rétine se comporte comme une matrice de processeurs spécialisés dont la répartition est fonctionnellement déterminée et qui collaborent pour créer le message transmis.

UNE ARCHITECTURE COMPLEXE

Les travaux de la physiologie ont mis en évidence l'organisation des récepteurs et des voies de conduction : 160.000.000 de récepteurs spécialisés, (cellules sensorielles à cones et à batonnets), recoivent le message visuel qui subit un premier traitement lié à la spécificité des différents types de capteurs, à leur répartition et à la topologie des voies de conductions non seulement ascendantes mais aussi horizontales.

> Les cellules sensibles au mouvement sont nombreuses en périphérie de la rétine et rares dans la fovea, répartition inverse de celle des cellules donnant une réponse stable (tonique).
> Le phénomène de sommation résulte d'une concentration verticale - de 160.000.000 de récepteurs à 1.000.000 de neurones dans le nerf optique - et horizontale dès la rétine sous l'action des cellules amacrines.
> L'inhibition est un mécanisme qui renforce les contrastes et aide à distinguer les contours dès que le contraste différentiel est supérieur à un seuil constant, ainsi que MACH l'a décrit.

Ainsi donc l'image est d'emblée discrétisée et analysée : les capteurs travaillant en parallèle assurent un filtrage et ne laissent remonter vers les structures supérieures qu'une partie de l'information (on estime à 1/100 le rapport du nombre des récepteurs au nombre des stimuli que comporte le message visuel transmis). On pense que le message progresse en abstraction au fur et à mesure de son intégration dans les niveaux supérieurs (IMB 83).
Toutefois, après les corps genouillés, le nombre des neurones augmente à nouveau jusqu'à 1.400.000.000 dans le cortex visuel.

> Les corps-genouillés jouent d'ailleurs un role organisateur: recevant des afférences des deux yeux, ils sont le siége de connections horizontales entre les différentes couches de neurones et initient sans doute le mécanisme de fusion nécessaire à la vision binoculaire (RUC 79)

Dans les niveux supérieurs, l'organisation devient très complexe et nous noterons seulement qu'actuellement nous connaissons l'existence de champs récepteurs spécialisés dans la détection de figures de contraste symétriques, asymétriques ou orientées.

UN CENTRE D'INTERET

La topographie privilégie la région maculaire, centre du champ de vision, par la densité des cones et leur innervation puisque chacun d'entre eux est relié à un neurone alors qu'à la périphérie il n'y a plus qu'un neurone pour 250 récepteurs.

Privilégiée par la répartition des récepteurs, la macula l'est aussi au niveau cortical. En effet, l'arrangement des voies optiques (croisement partiel au niveau du chiasma) détermine une latéralisation des hémisphères, chacun ne voyant que la partie du champ controlatérale. Mais on a montré que, pour la seule région maculaire, il existait une représentation bilatérale, étendue, au niveau de l'aire visuelle primaire (aire striée n°19 de BRODMANN); ceci peut se traduire en disant que la "résolution corticale" serait 10 fois meilleure que celle de la rétine - (RUC 79) -. Ainsi chaque hémisphère dispose de l'information relative à un des deux cotés du champ de vision et de l'information correspondant au centre d'interet situé dans l'axe optique.

EN CONCLUSION

Les quelques acquis de la neurobiologie de la vision que nous venons de voir ont montré la complexité de l'organisation de l'oeil et surtout notre ignorance de "l'activité cognitive que constitue une perception visuelle,..., mégacomputation,..., computations de computations" (MOR 84).

Ces constatations nous montrent l'ambition de l'objectif assigné aux systèmes de vision par ordinateur. Toutefois la réalisation de ceux-ci est possible si on admet de limiter suffisament le domaine pour intégrer aux mécanismes de compréhension des informations contextuelles en nombre compatible avec nos moyens actuels.

L'IMAGERIE

LE TRAITEMENT D'IMAGES

ROLE ET PROBLEMES DE L'IMAGERIE MEDICALE

Le médecin est plus que d'autres, consommateur et producteur d'images. A l'hopital d'abord, où la concentration des moyens facilite le recours à l'auxiliaire précieux qu'est l'image, mais aussi chez les praticiens qui peuvent se doter d'appareillages plus légers, échographes, thermographes ou appareils Doppler.
Historiquement la Radiographie, la première, s'est imposée comme une discipline médicale à part entière. Ensuite la Médecine Nucléaire s'est individualisée et aujourd'hui les moyens d'imagerie abondent avec notamment le Scanner X devenu indispensable après 10 années d'existence, la Résonance magnétique nucléaire, très prometteuse, et l'Angiographie numérique qui banalise plus encore les examens radiologiques.

INTERPRETATION DES IMAGES MEDICALES

Si l'imagerie médicale est un enjeu industriel ainsi que nous l'avons remarqué en introduction, l'image est, elle, un enjeu médical.

Face à l'inflation des images il est difficile au médecin d'extraire de la masse des informations à sa disposition les renseignements utiles; en raison de leur nombre (à cet égard le poids du dossier d'un patient hospitalisé type est édifiant) mais aussi en raison de leur contenu toutes ces images requièrent des outils de gestion et de manipulation.

Citons quelques exemples afin d'expliciter ces problèmes:

- Une série de tomographies du thorax représente de 5 à 40 images dont la lecture nécessite un oeil averti; en outre les dossiers sont constamment réinterprétés par les différents médecins intervenant au décours de l'évolution de la maladie.

- Les techniques comme la scintigraphie ou la thermographie fournissent des images qui sont des cartographies exprimant la répartition spatiale de paramètres non pas morphologiques mais physiques, chimiques ou fonctionnels. Il ne faut plus alors parler d'image mais de cryptogramme : l'oeil est privé du secours d'un modèle morphologique pour guider sa lecture et l'image devient pour le béotien un patchwork joliment coloré. Remarquons à cet égard que, dans leurs utilisations médicales, les images RMN parlent plus par leur aspect morphologique que par la nature du paramètre physique qu'elles représentent; en effet il est difficile d'interpréter, au niveau macroscopique auquel se situe le médecin clinicien, un temps de relaxation spin-réseau. Ici transparait l'utilité d'une aide à l'interprétation pouvant prendre en compte le facteur iconographique sans perdre de vue la réalité du phénomène physique décrit par l'image.

- Dans d'autres cas, comme en échocardiographie doppler-pulsé, l'image ne se situe plus dans un plan de projection ou de coupe mais l'un des axes représente le temps. Il faut alors éduquer le lecteur peu familier avec ces représentations dans un plan temps-espace.

- Enfin la lecture de certaines échographies n'est possible, meme pour un médecin échographiste, qu'en sachant quel est l'organe exploré et quelles ont été les conditions dans lesquelles l'examen s'est déroulé. En bref, seules des informations contextuelles permettent le déchiffrage des images.

Nous avons évoqué la diversification des techniques des examens paracliniques. Ceci pose un problème et il faut distinguer, face à l'image, trois intervenants :

- Le médecin spécialiste qui réalise l'examen; lui a appris par une longue pratique à interpréter ses images. Il connait la place de l'examen dans l'arbre de décision diagnostique et c'est lui qui produira une interprétation destinée au prescripteur de l'examen. Il sait ce que représentent les images, par quelles manipulations elles ont été obtenues.

- Le médecin hospitalier, clinicien ; c'est lui qui demande l'examen, qui juge de son opportunité. Il reçoit avec les images un compte rendu; toutefois il doit etre capable de lire les images par lui-meme. Ici le statut des images est variable, et, si tous apprennent à lire les radiographies du thorax par exemple, il n'en est plus de meme s'agissant d'examens dont le principe est moins bien connu parce que nouveau ou ininterprétable en terme de médecine praticienne. En outre, les traitements informatiques peuvent créer de nouvelles images dont le contenu est différent de celui initialement présent lors de l'acquisition des images. Nous touchons ici à un réel problème puisqu'il importe que le clinicien domine l'ensemble du dossier alors que les aspects techniques de certaines composantes lui échappent de plus en plus.

- Le médecin praticien hors du cadre hospitalier vit la meme situation avec peut-etre le handicap supplémentaire d'etre plus loin de la source d'informations.

Nous ne ferons que signaler qu'outre le problème de l'interprétation se pose le problème plus difficile encore de la stratégie diagnostique qui impose aussi la connaissance et la maitrise des techniques. Finalement ces quelques lignes feront sentir, nous l'espérons, la nécessité d'outils de traitement et de communication afin de ramener l'information plus près des utilisateurs.

LES BESOINS DU TRAITEMENT DE L'IMAGE MEDICALE

De ces observations nous retirerons les leçons suivantes :

- Les moyens de calcul doivent gagner en puissance pour permettre à l'image de prendre toute la place qu'elle mérite, en levant les difficultés actuelles que pose l'interprétation immédiate de la masse des examens quotidiens.

- Le traitement d'images est un auxiliaire précieux mais il devrait etre transparent pour le médecin qui interprète l'examen; ce serait alors le signe de son efficacité.
Toutefois les images et autres objets résultant du traitement doivent impérativement etre transmis, au jour le jour, au service clinique prescripteur de l'examen.

- L'interprétation de ces images repose sur la connaissance du contexte : organe exploré, nature des informations représentées, mode de représentation adopté.

- Elle fait appel à des informations de nature différente : données anatomo-cliniques, séméiologiques, ..., capacité d'analyse et de synthèse, et donc, à un savoir et à un savoir-faire

- Le nombre des images et leur diversité suggérent la création d'une base d'images documentée permettant de confronter les images, issues des nombreux moyens d'imagerie disponibles, aussi bien entre-elles qu'à des données de nature différente.

Du point de vue systèmique, il apparait que le dépouillement des images recourt à une trilogie analyse-interprétation- contexte, fille du macro-concept cerveau-esprit-culture développé dans <MOR 84>. Reformulant en utilisant le vocabulaire de l'informatique nous dirons que les besoins existent pour un traitement d'images et une analyse de scène, le premier n'étant pas seulement subordonné à la seconde.

LES MOYENS DU TRAITEMENT D'IMAGES

En préambule nous noterons que le concept de "moyen" implique, pour la mise en oeuvre des outils le concrétisant, une formulation explicite des étapes nécessaires; il faut en effet matérialiser chaque étape du traitement, fusse par l'écriture, en se pliant donc aux contraintes - en utilisant les facilités ? - du matériel et de l'environnement logiciel. Ce concept repose donc sur la possibilité de décomposer toute action en opérations élémentaires binaires booléennes qui sont les atomes de tout traitement <DUF 82>.
Ainsi l'intrication étroite matériel-logiciel-algorithme apparait nettement : les limites des traitements possibles sont présentes dans les possibilités offertes par le matériel <WOO 81>. A ce propos DUFF note qu'il faut pour s'adapter aux moyens, les machines parallèles dans son exemple, redéfinir les algorithmes pour les adapter au matériel. Ainsi l'organisation du matériel, conçue pour supporter la fonction, autorisera une plus grande efficacité dans la spécification et l'exécution des processus mais imposera ses éventuelles faiblesses.

C'est pourquoi nous évoquerons les aspects matériel et logiciel de quelques notions actuellement développées comme le parallèlisme, la spécialisation des processeurs et, pour le logiciel principalement, la structuration des données, permettant une hiérarchisation des informations en vue d'accéder à une méta-connaissance seule capable d'aider à l'interprétation.

DES MACHINES CONCUES POUR LE T.I.

Tout d'abord il convient de citer la typologie en vigueur pour classer les machines de traitement de l'information:

Type dans la classification de FLYNN	Application
SISD Single Instruction, Single Data stream	Processeur classique
SIMD Single Instruction, Multiple Data stream	Processeur vectoriel
MISD Multiple Instruction, Single Data stream	/ (Non-sens)
MIMD Multiple Instruction, Multiple Data stream	Multiprocesseur

Sans oublier d'autres machines telles ILLIAC, CLIP4, DAPP et d'autres, nous citerons maintenant quelques exemples, dans le seul but de présenter les mécanismes des structures parallèle et pipeline et les particularités des chemins de données facilitant l'utilisation de ces machines pour le traitement d'images.

PROPAL II

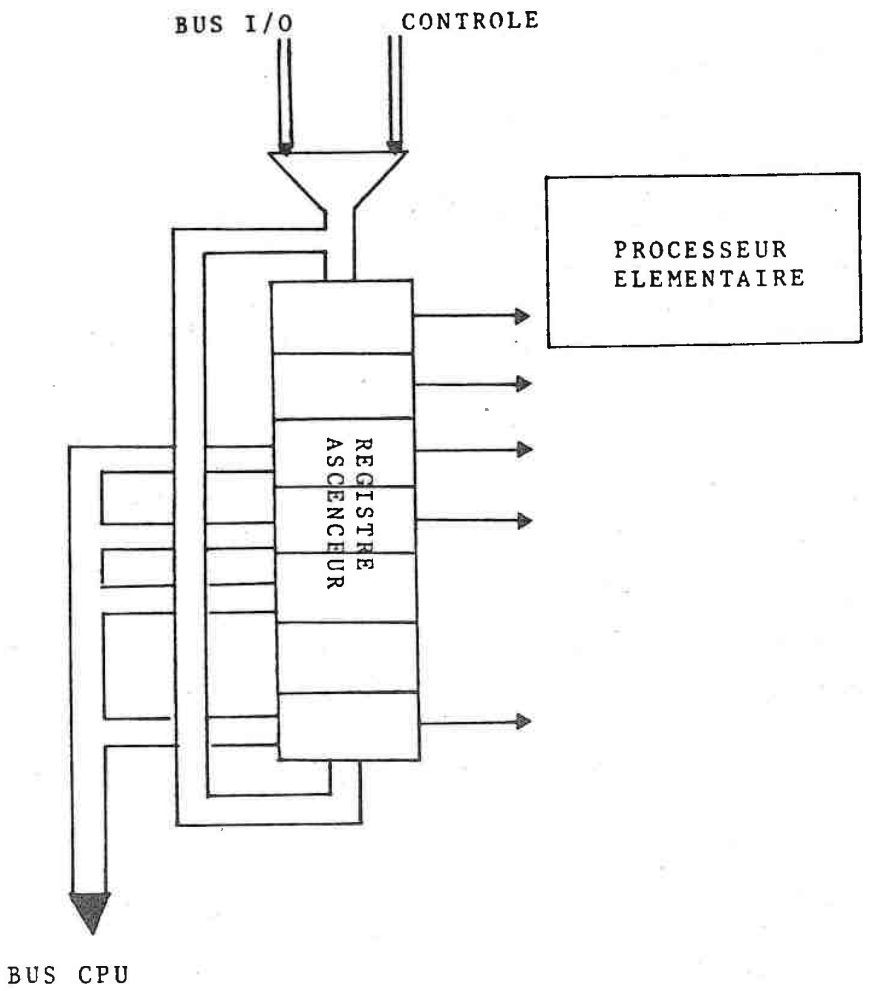
Cette machine a une architecture purement parallèle comportant de 128 à 2048 processeurs élementaires (TIM 78). Elle est du type SIMD et les processeurs ne sont donc pas spécialisés.

En revanche, les chemins de données sont adaptés à la fonction et un "registre ascenseur" permet la circulation des données d'un processeur à l'autre concrétisant de manière très imparfaite la notion de voisinage, essentielle en T.I (fig III.1).

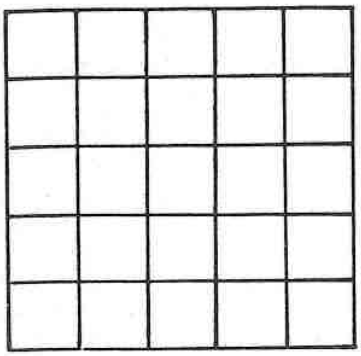
Le logiciel utilise la notion de zone, bloc de donnée contenant une unité d'information pour chaque processeur élementaire. Ainsi cette organisation se prete à la mise en oeuvre de notions comme la fenetre qui permet de focaliser l'intéret sur une partie de l'image.

En outre, PROPAL II comporte une unité séquentielle qui effectue les traitements série et controle les processeurs parallèles, assurant leur alimentation en données et le chargement des microprogrammes, comme c'est classiquement le cas sur ces machines.

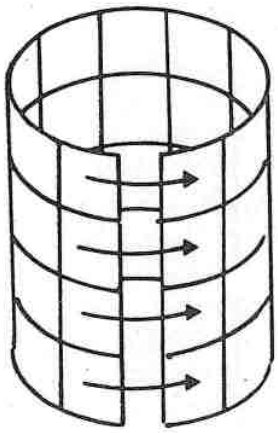
Il faut remarquer que le désavantage de cette structure parallèle réside dans la duplication des logiques de controle et la complexité des chemins de données.



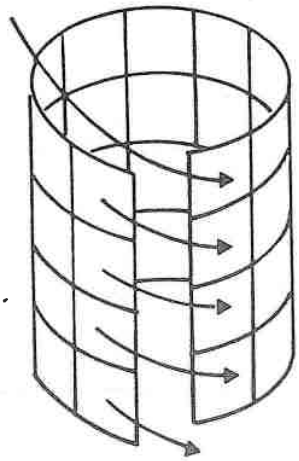
PROPAL II - Fig. III.1. d'après ⟨TIM 78⟩



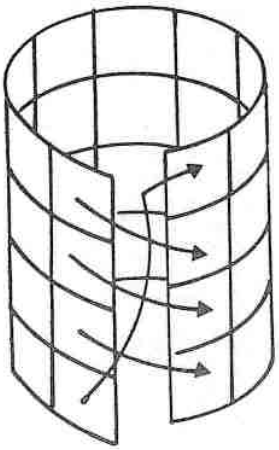
OUVERT



BOUCLAGE
CYLINDRIQUE



SPIRALE OUVERTE



SPIRALE FERMEE

TOPOLOGIE UTILISEE DANS MPP

Fig. III. 2.

MPP : Massively Parallel Processor - ⟨KEN 80⟩

Il s'agit d'une matrice carrée de 16384 processeurs ⟨SIMD⟩ communicant chacun avec leurs 4 voisins.
Les chemins de données présentent la particularité de pouvoir connecter par programmation les bords du carré 128.128 avec le bord opposé. Précisement il existe quatre type de connectivité entre les bords droits et gauches : ouverte, cylindrique, spirale ouverte, spirale fermée ⟨fig III.2⟩.
Les sorties de données des processeurs peuvent etre combinées en un arbre "OU" permettant l'extraction d'une valeur unique ⟨maximum par exemple⟩ à partir d'une matrice en entrée.

CYTOCOMPUTER

CYTOCOMPUTER est une machine pipeline spécialisée explicitement dévolue au traitement d'images ⟨STE 82⟩.
La structure comporte deux niveaux :

- le premier est un pipeline série faisant communiquer les modules du second niveau ⟨fig III.3⟩.

L'avantage de ce type d'organisation est que, lorsque le pipeline est chargé ⟨temps de latence du pipeline⟩, le temps d'entrée-sortie des données se fusionne avec le temps de calcul. On démontre - ⟨LOU 80⟩ - qu'à la latence du pipeline près le temps de calcul ne dépend plus de la complexité de l'algorithme ce qui n'est pas le cas des processeurs vectoriels.
Si de plus la chaine du pipeline comprend un nombre d'étages égal au nombre de lignes de l'image, il n'est pas nécessaire d'utiliser une stockage externe, quelle que soit la complexité du traitement à condition de reboucler la sortie du dernier étage sur l'entrée du premier ⟨STE 82⟩.
Le pipeline, bien adapté aux manipulations répétitives, est donc un exemple parlant du lien entre l'organisation et la fonction ⟨la syntaxe et la sémantique⟩; cette structure plus économique en chemin de données devient très performante lorsque la complexité des algorithmes croit.

- les modules constituant le second niveau reposent sur la notion de voisinage 9 points. Des registres à décalage dimensionnés pour contenir 2 lignes contigues de l'image alimentent des registres "fenetre" contenant les 9 points d'un voisinage; grace à un décalage synchrone ils réalisent ainsi une fenetre mobile se déplaçant avec le balayage séquentiel de l'image originale (fig III.2). Chaque module effectue une transformation préprogrammée affectant le résultat au pixel centre de la fenetre. A la fin de chaque cycle la sortie du module est transmise au module suivant, propageant ainsi le balayage séquentiel. Cette fenetre 9 points matérialise la notion de masque en T.I.

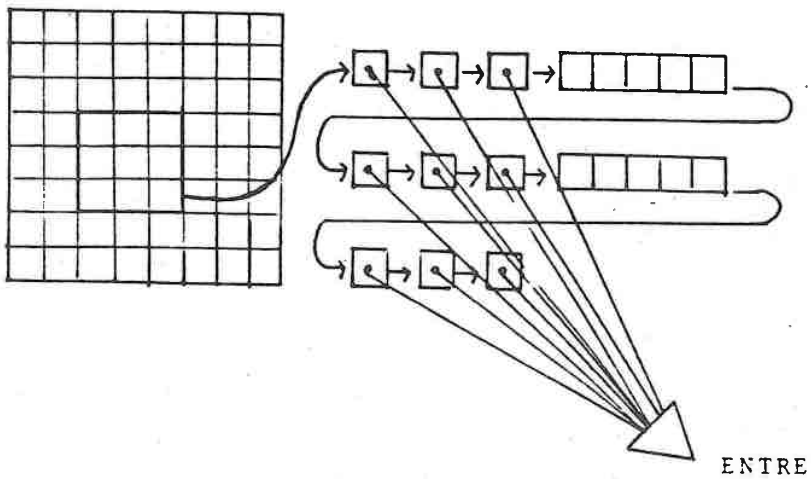
Le pipeline et la fenetre véhiculent des notions sémantiques - traitement répétitif sur des données à un accés séquentiel pour le premier et opérateurs de voisinage pour la seconde -. Apportant une aide à la mise en oeuvre de concepts abstraits, ils imposent en contrepartie de respecter cette sémantique faute de perdre leur efficacité.

GOP

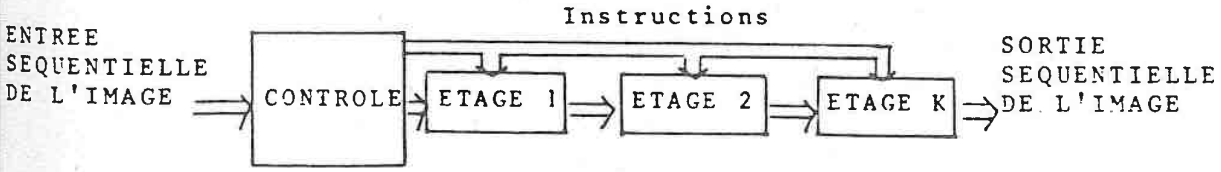
Ici on trouve une machine MIMD structurée en deux étages (fig III.4), le premier parallèle et le second séquentiel - (GRA 81) -.

Le premier étage comporte 4 pipelines parallèles, une mémoire image, une mémoire contenant le masque et désignant les points à utiliser dans la mémoire image ce qui est une réalisation des concepts de filtre ou de tache développés abstraitement dans certains langages T.I. - (BEL 84) - . On peut alors sélectionner une partie de l'image hors de tout critère de voisinage, méthode relative, en la désignant de manière globale soit par sa géométrie soit par d'autres propriétés.
La lecture des données se fait ici ligne par ligne.

Le second étage, où le flux de données est considérablement plus faible a une organisation série, utilisant des processeurs spécialisés. Elle initie les taches dévolues à la première partie et analyse les résultats. Il existe donc une hiérarchie dans les processus, les uns pouvant controler les autres.

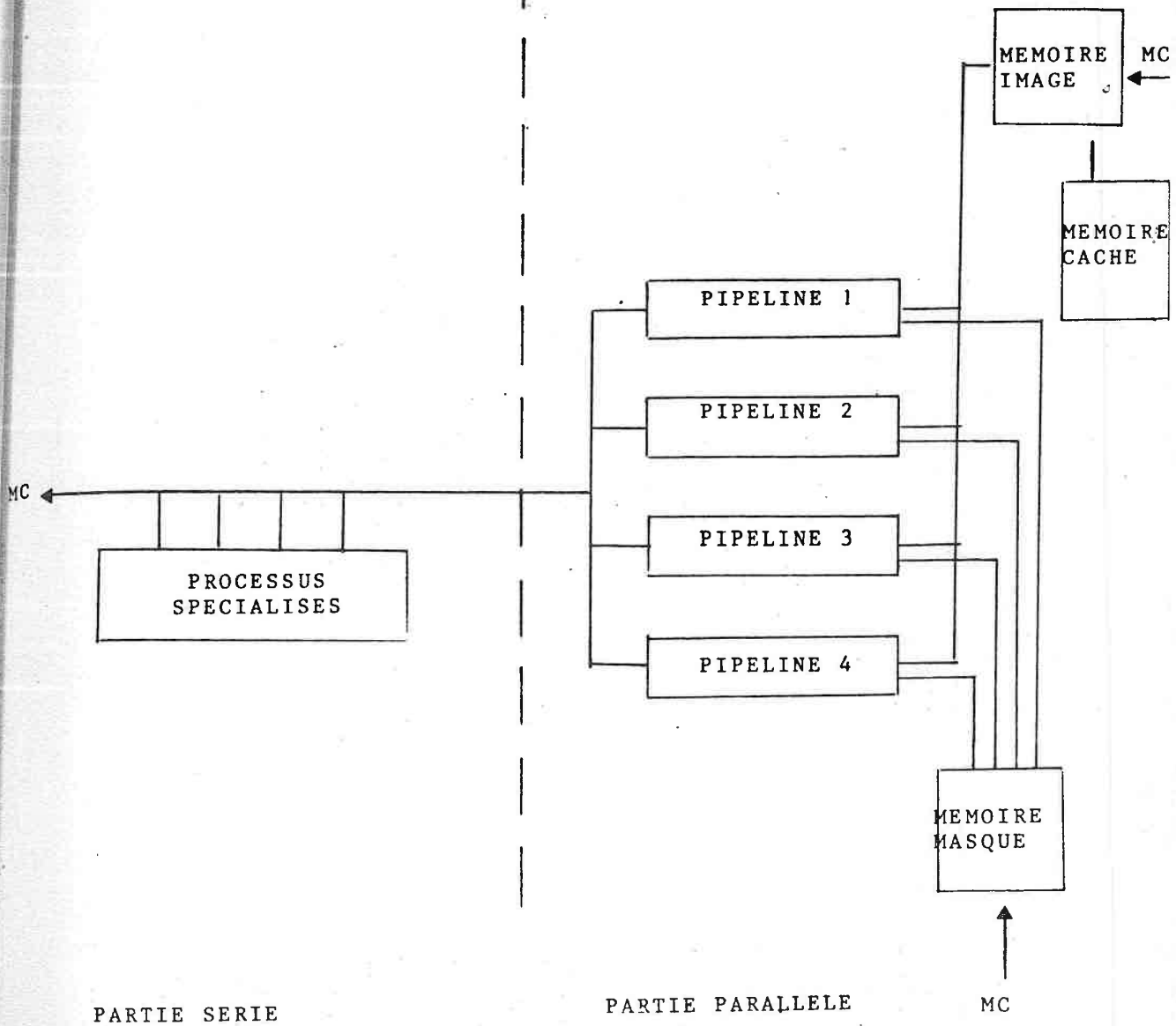


STRUCTURE D'UN ETAGE



PIPELINE

CYTOCOMPUTER - Fig.III.3. - d'après ⟨LOU 80⟩


MEMOIRE IMAGE
MC
MEMOIRE CACHE
PIPELINE 1
PIPELINE 2
PIPELINE 3
PIPELINE 4
MC
PROCESSUS SPECIALISES
MEMOIRE MASQUE
PARTIE SERIE
PARTIE PARALLELE
MC

GOP Fig. III.4.

Le projet ICOTECH

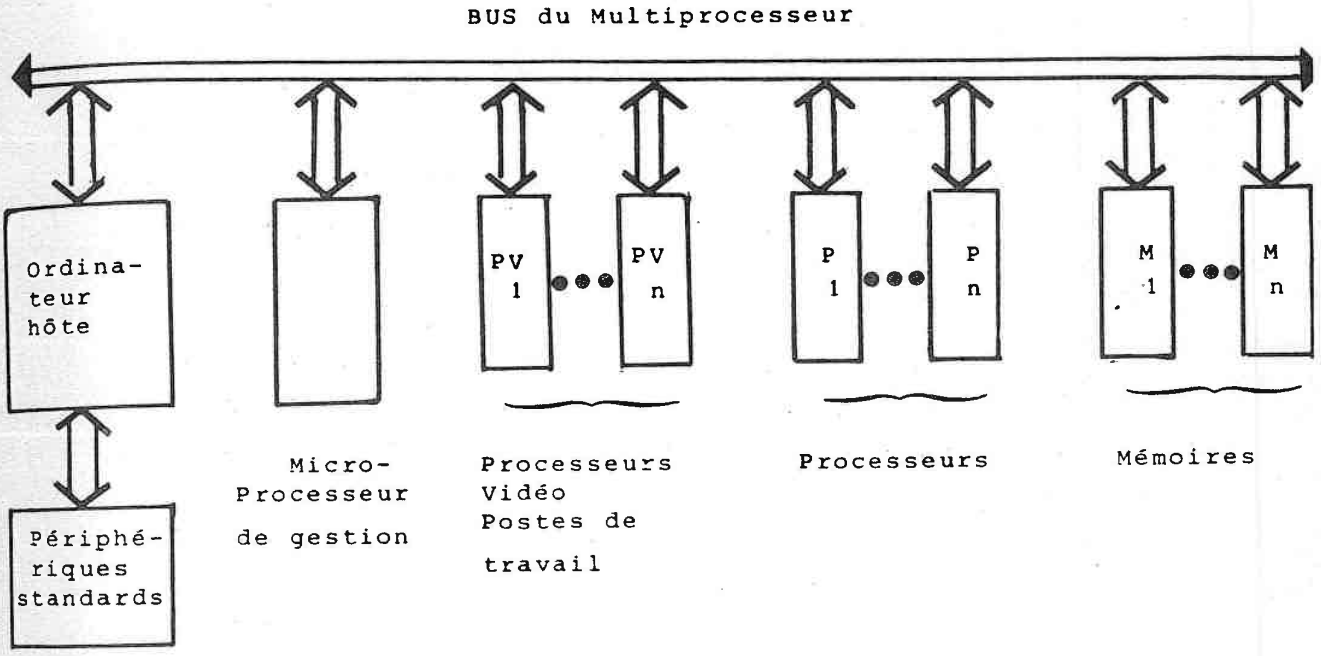
Ce projet, en cours d'industrialisation, est développé principalement par les équipes du GREPA de Strasbourg et du CRIN de NANCY associées à d'autres laboratoires. L'organisation retenue ici diffère sensiblement de celles que nous venons de voir - <MEG 82> -.

Le coeur du système est um multiprocesseur, PRIM (fig III.5), architecturé autour d'un bus bi-directionnel auquel sont connectés des modules divers :

- modules d'entrée-sortie, dit "processeur vidéo", pour l'acquisition et/ou la restitution des images.
- modules de mémoire image
- modules de traitements spécialisés dans l'éxecution de fonctions particulières. Les modules proposés réalisent les transformations suivantes : convolution, transformée de FOURIER rapide, diverses transformations géométriques (translation, zoom, rotation), combinaisons linéaires d'images.

Enfin le processeur-vidéo active les processeurs spécialisés et gére les données.

Cette architecture, que l'on retrouve sur des machines commercialisées comme le PERICOLOR de la société NUMELEC par exemple, présente l'avantage d'un cout modéré et de hautes performances dues aux processeurs spécialisés. La pierre d'achoppement est l'organisation centralisée des chemins de données et c'est le bus qui limite pratiquement la rapidité de ces systèmes.



Projet Icotech - Fig.III.5 d'après ⟨MEG 82⟩

COMMENTAIRES

Les machines que nous avons passé en revue représentent un effort considérable pour adapter les moyens aux buts. Elles répondent aux objectifs suivants :

- mise à disposition de l'utilisateur d'une puissance de calcul importante, permettant un traitement en "temps réel" (à la cadence vidéo ?) des images,
- implémentation des concepts de centre d'intéret (fenetre), de sélection de région (filtre), de voisinage (masque).

Toutes supposent une image organisée en tableau dont l'atome est le pixel, et la fonction d'accès séquentielle - toutefois GOP saisit les données ligne par ligne -.

Structure des données

La remarque précédente nous amène à discuter du problème de l'organisation des données.
Comme le notent de nombreux auteurs, l'interprétation des images implique de manipuler des objets hiérarchiquement organisés : partant du point image (Pixel) on extrait des lignes puis des régions et enfin des objets. A ce sujet NAGIN - (NAG 81) - parle de résolution décroissante et DEBORD - (DEB 82) - distingue les notions de segmentation picturale (extractation d'objets T.I.) et de segmentation sémantique (extraction des concepts dénotés par les objets T.I.).

A titre d'exemple, voici quelques notions que présente NORTON-WAYNE - (NOR 81) - :

ligne:

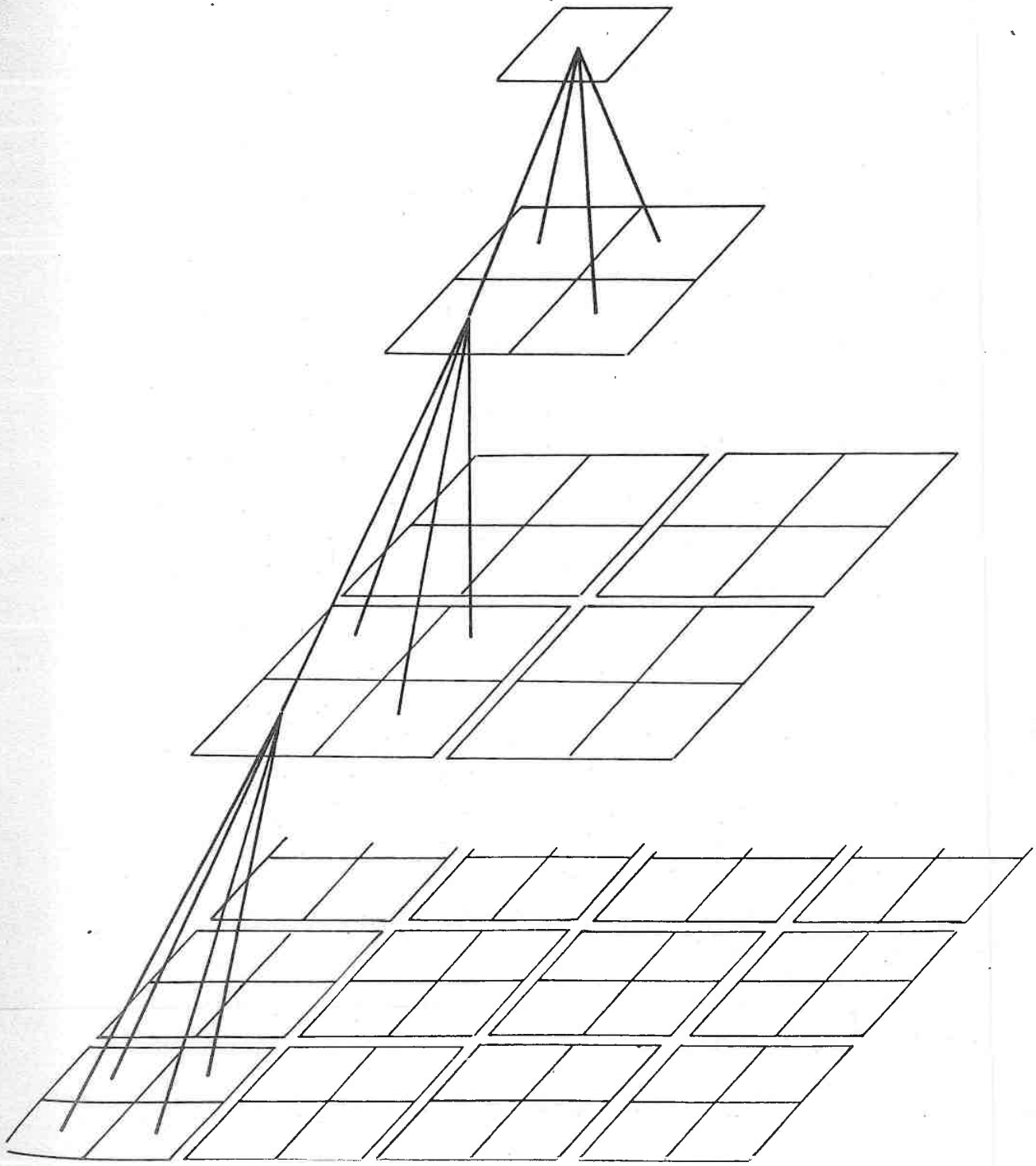
définie par une longueur, une direction et une origine,

frontière:

contour fermé, dont l'intérieur est un objet et dont chaque élément a deux voisins, et deux seulement, appartenant à la frontière,

balayage clairsemé:

certaines opérations ne demandent d'accéder qu'à une fraction des points de la grille,



ARBRE QUATERNAIRE

Fig. III. 6

accès aléatoire:

l'acquisition des données ne se fait que lorsque les informations apparaissent utiles,

accès sequentiel:

sur les caméras du type barrette CCD, par exemple, acquérant les données ligne par ligne, il serait intéressant de n'utiliser que des traitements portant sur deux ou trois lignes consécutives.

ligne de force:

notion définie dans (ENO 81).

Mais on remarque que ces notions intrinséquement ne comportent aucune hiérarchie des informations; les mécanismes de stratification du traitement restent dévolus à l'algorithmique. Par ailleurs, il importe de définir la finalité du traitement : une frontière est un moyen économique de désigner un objet en vue d'une interprétation de l'image, mais, pour une amélioration d'image par exemple il est plus efficace de désigner la région par l'ensemble de ses points.

D'autres auteurs - (NAG 81), (UHR 82) - présentent la notion de cone perceptuel. Il s'agit d'une généralisation de la structure d'arbre quaternaire, (quad-tree; fig III.5) dans lequel l'unité d'information se projette au niveau inférieur sur une région image de taille variable. Cette organisation stratifiée permet de définir des opérations horizontales (les données et le résultat appartenant à un meme niveau), des réductions et des projections. Ainsi partant d'un objet, sommet du cone, il est possible de le modéliser en décrivant ses composantes, lesquelles sont à leur tour détaillées, ... Ici l'outil est bien adapté à la modélisation et à la représentation d'informations d'abstractions croissantes.

Enfin, il faut noter que ces structures sont en règle définies au niveau logiciel. Le matériel lui supporte les notions de voisinage et de parallélisme adaptées au traitement des tableaux (quelques moyens existent, comme dans MPP, pour restituer un scalaire à partir d'une image permettant une condensation de l'information et un niveau d'abstraction plus poussé).

DES LANGAGES POUR LE T.I.

Nous aborderons ici plusieurs thèmes, touchant aux notions de parallèlisme, de langage de traitement d'images et de système d'analyse de scènes. Il ne s'agit pas dans ce paragraphe d'une revue des langages mais plutot d'évoquer brièvement, les tendances du traitement d'images et les solutions apportées aux problèmes posés.

LES LANGAGES DE PROGRAMMATION - <MAG 81>, <PRE 81> -

Compararant des langages existants, MAGGIOLO-SCHETTINI note qu'une majorité des auteurs conçoit ces langages comme une collection de modules, écrits en langage d'assemblage ou en FORTRAN souvent. La tendance est alors d'utiliser le système d'exploitation de la machine, un interpréteur de commandes permettant l'accès à la librairie des sous-programmes. Les domaines couverts par ces librairies sont la visualisation, l'arithmétique, les manipulations géomètriques, l'amélioration d'images, l'extraction de primitives T.I. et la décision.

Pour développer des langages exploitant le parallèlisme certains auteurs ont choisi ALGOL - <LEV 81.1> - ou FORTRAN mais le recours à PASCAL apparait s'imposer compte tenu de la diffusion de ce langage et de la possibilité de définir des objets typés.

Les caractéristiques de ces langages reposent sur la possibilité d'effectuer des traitements parallèles sur des tableaux, la sélection de sous-tableaux, les fonctions d'accès aux voisins d'un point, les instructions parallèles avec controle global ou local.

On trouve dans ces langages, destinés à la manipulation des tableaux, des constructions ad-hoc :

- Traitements parallèles :

déclaration des tableaux soumis aux traitements parallèles (déclaration standard du langage de base pour PIXAL - <LEV 81.2> - et PAL <KUL 81> - ou déclaration explicite dans le cas de PASCAL L - <RAD 81> - et de PASCAL PL - <UHR 81> -)

assignation parallèle

ajout d'une dimension au tableau

- Sélection des parties d'une image :

déclaration des frontières de la région à traiter (type **border** de PASCAL PL, **edge-of** de PIXAL, **filtre** de SAPIN - <BEL 84> -)

déclaration des limites de variation des indices du tableau permettant d'utiliser implicitement la notion de fenetre

notion de **fenetre**, rectangle dont la taille et la position sont variables, et servant à définir la partie de l'image soumise à traitement

- Notion de voisinage :

masque définissant un opérateur local à appliquer à l'image avec des possibilités de sélection des points du voisinage participant à l'opération

- Instructions de controle :

IF...THEN...ELSE...,

WHILE ANY...DO...,

IF ALL... THEN...,

la condition pouvant porter sur une grandeur scalaire (controle local) ou un tableau (controle global)

itérations du type FOR ALL...DO...

A titre d'exemple nous citerons deux réalisations :

PIXAL - (LEV 81)

Ce langage veut offrir la possibilité de définir des processus parallèles et/ou séquentiels. Explicitement il vise à la manipulation de PIXels et au développement d'ALgorithmes. Il s'appuie sur ALGOL 60 en raison de la richesse de ses structures de controle et de l'effort de formalisation dont il a été l'objet.
Il définit les types de frame, de masque et de frontière adaptés au T.I.:

- Le masque et le frame définissent des structures multidimensionnées que l'on peut positionner sur l'image. Le frame sert à définir une sous-image.
- La frontière permet de partitionner l'image en bruit de fond et signal sur des critères non pas géométriques mais guidés par le contenu de l'image.

Les instructions spéciales permettant le parallèlisme sont FOR UNTIL STEP, ASSIGNATION, PROCEDURE, IF THEN, WHILE DO.
A noter qu'il est possible de combiner les instructions parallèles et séries IF A=0 THEN goto Stop ELSE... (A=0 teste l'ensemble de la matrice A)
Les constructions spécialisées permettent l' accés aux lignes et aux colonnes, la somme des éléments (d'un tableau, vecteur), la comparaison d'un masque à un voisinage, la multiplication par un masque ou l'écriture d'un masque sur un voisinage.

PICASSO - (KUL 81)

Le langage repose sur une bibliothèque écrite en assembleur implémentant la plupart des algorithmes T.I.(PICASSO). A cela s'ajoutent deux outils créant un environnement facilitant l'usage de la librairie :

- Un langage interactif PICASSO-SHOW, doté d'un interpréteur, permettant l'accès à la librairie et l'exécution, interactive ou non, des programmes.
- Un langage de haut niveau PAL, s'appuyant sur ALGOL 68, pour l'écriture de programmes structurés et l'utilisation de variables typées.

LES SYSTEMES DE TRAITEMENT D'IMAGES

Aux langages de programmation sont associés des outils permettant l'utilisation des primitives de T.I.. Dans ces systèmes l'accent est mis sur la finalité du traitement et VROLIJK au sujet de TAL rappelle que l'image n'est pas destinée aux "traiteurs d'images" mais aux utilisateurs finaux, notamment médecins (LEYTAS, le système surlequel se greffe TAL est un processeur d'images associé à un microscope - <VRO 81> -). Ces consommateurs utilisent l'informatique mais ne veulent pas la connaitre. Il faut leur offrir des facilités de mise en oeuvre ce qui est le but de systèmes qui proposent des modes de fonctionnement immédiat programmé, tels que TAL; LAMPION <M. REBUFFET, ECTA) ou SAPIN - <CAS 83>, <BEL 83> -.
Des trois systèmes présentés nous ne décrirons que quelques aspects utiles à la présentation de notre travail.

TAL

Le système se présente sous forme de menus chainés. Ceci permet de définir par raffinements successifs la fonction à exécuter, approche interactive qui facilite la mise au point des algorithmes de traitement. L'opérateur est guidé, choisissant les paramètres successifs sous controle du sytème, jusqu'à la rédaction correcte d'une commande. Celle-ci est alors exécutée et le résultat disponible dès la fin de la commande.
Une option PROGRAM CONTROL permet l'écriture de programmes grace à un éditeur; l'exécution peut ensuite etre lancée avec la possibilité de revenir en mode pas à pas ou d'arreter l'exécution pour corriger certains paramètres avant de relancer le programme. Enfin l'option PERIPHERAL CONTROL autorise le transfert des programmes de la mémoire tampon de TAL à un périphérique et inversement.

LAMPION et SAPIN

Il s'agit de systèmes à menus qui offrent des commodités d'utilisation comme :

- l'accès à une documentation des modules exécutables à partir du menu.
- le déplacement dans l'arborescence des menus sur des critères de similitude algorithmique ou fonctionnelle.
- un mode TRACE mémorisant les traitements (les déplacements dans l'arborescence) et permettant de rejouer une séquence de traitements.
- un langage interprété, doté de structures de controle, pour la rédaction des commandes.
- un générateur de commandes permettant d'insérer de nouvelles fonctions dans les menus.

Dans ces deux systèmes, il existe des objets T.I. typés prédéfinis. Ceci facilite l'écriture de nouveaux algorithmes de traitement, le format des objets utilisés tant en entrée qu'en sortie étant bien spécifié, et autorise l'ajout de feuilles nouvelles à l'arborescence (commandes compilées). Ainsi, dans le cas de SAPIN, un chapeau standardisé sert de boite aux lettres pour le passage de la zone documentaire et l'énumération des objets manipulés. Après compilation séparée du module, ces informations sont récupérées par le système SAPIN grace à une commande INSERT qui ajoute la nouvelle fonction aux menus.

Nous avons dit l'intéret de l'approche proposée par ces systèmes pour le consommateur d'images, mais n'oublions pas avant de conclure ce paragraphe que le prix de l'interactivité est une certaine lenteur (qui se justifie en raison de leur vocation déclarée au développement). En outre, ces systèmes sont encore orientés vers le traitement d'image et non pas l'analyse de scène dont l'objectif, plus ambitieux, est la description du contenu des images.

PRINCIPES DE L'ANALYSE DE SCENE

BUT

L'analyse de scène a recours à la modélisation. Son but est l'extraction d'objets puis l'identification à un modèle. Dans cette optique il est utile de concevoir des logiciels capables de générer leur propres modèles qu'ils peuvent extraire de l'analyse d'une scène réelle - <PER 81> -. Cette capacité, fondamentale, d'apprentissage sera le garant de l'adaptabilité des systèmes de vision à des taches diverses.

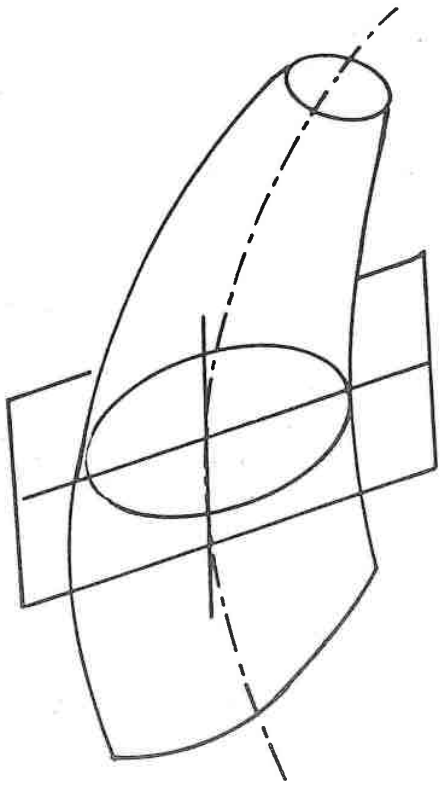
L'objectif ainsi défini est toutefois trop ambitieux et si des approches théoriques - <MAR 82>, <WIN 75> - envisagent le problème de la vision dans sa généralité, les systèmes effectivement mis en oeuvre restreignent le domaine. A coté de réalisations comme ACRONYM utilisant une méthode dite des cones généralisés qui autorise une très grande souplesse dans la description d'objets tridimensionnels, la plupart utilisent une vision 2D. En outre ils incluent des modèles en nombre limité.

ORGANISATION

Elle repose sur trois notions - <BAL 77>, <FRE 77.2> - :

- bas-niveau, chargé de l'extraction des primitives,
- haut-niveau, décrivant le modèle,
- interprète, assurant l'adéquation entre ces deux niveaux

Le bas-niveau permet de manipuler l'information brute pour créer ou modifier une base de faits contenant les primitives extraites, objets T.I., qui sont des informations spécifiques du cas d'espèce traité. Cette base, modifiée à chaque session, est la mémoire à court-terme du système.



Méthodes des Cônes généralisés

Système ACRONYM (BINFORD) Fig. III.7

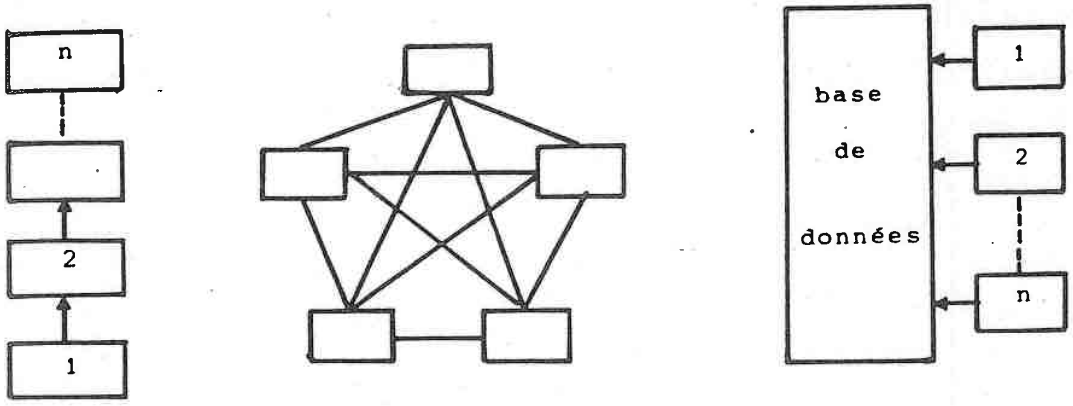
La notion de paradigme segmentation - interprétation traduit l'idée selon laquelle ces deux étapes sont deux aspects d'un meme processus cognitif. LUX présente la contreverse qui est née à ce sujet - <LUX 84> - : face aux partisans de ce paradigme, on trouve les tenants de la thèorie de MARR - <MARR 82> - selon laquelle la vision humaine permet d'interpréter des scènes dont les composantes ne sont pas des objets connus. Ainsi la connaissance d'un modèle à-priori ne serait pas nécessaire. Toutefois la théorie de MARR ne débouche pas aujourd'hui sur des réalisations opérationnelles.

A l'inverse, on admet dans les systèmes fonctionnant que l'extraction des primitives ne doit pas etre considérée comme une procédure monolithique. Ainsi LEVINE note qu'au bas niveau une segmentation complète n'est pas possible - <LEV 82> -. Il propose donc une approche possible : initialiser une segmentation partielle qui est complétée dans le cadre d'une analyse hiérarchisée grace à une rétro-action descendante (le niveau supérieur guidant la segmentation par le bas-niveau).
Cette rétro-action entre les niveaux s'exprime sous la forme de règles condition-action; les conditions s'appliquent à un niveau supérieur à celui sur lequel s'exerce l'action (cf fig III.6) ; l'activation des règles est sous le controle de méta-règles définissant, dans une situation donnée, les critères de choix de la règle particulière à appliquer (exprimant une connaissance de la connaissance).

L'interaction entre les différents niveaux est donc nécessaire, comme nous venons de le voir. Elle repose selon les réalisations sur trois types de structures (Fig III.8) :

- Hiérarchique linéaire. Dans ce cas simple, lorsque le contenu des scènes à analyser est connu par avance, l'enchainement des opérations peut etre figé. On trouve dans <YAC 79> une application au cas des ventriculographies.

- Hétérarchique. Les règles définissant le comportement du système sont alors cachées dans les algorithmes ce qui nuit à leur clarté et gène les modifications.



a. Structure hiérarchique linéaire ascendante

b. Structure hétérarchique

c. Structure sur base de donnée

Fig.III.8 Les différentes structures

- Structure sur une base de données, statique et donc mieux maitrisée. Ici on trouve le domaine des systèmes à règles de production (SRP), utilisé pour la réalisation des systèmes experts. Cette approche est présentée en détail dans <LAU 82.1> et <LAU 82.2>.

LES SYSTEMES A REGLES DE PRODUCTION

Dans ces systèmes, l'approche algorithmique est délaissée au profit d'une organisation comportant de règles décrivant explicitement les connaissances de manière déclarative et d'un moteur d'inférence utilisant ces règles pour modifier la base de faits connus du système.
Les règles de production forment une base de connaissances, la mémoire à long terme du système.
Elles expriment soit un savoir (règles de production) :

Si condition portant sur les faits
Alors action sur la base des faits,

soit un savoir-faire (méta-règles) :

Si condition sur les règles
Alors action sur le controle

Une règle est donc un couple **condition-action:** l'action ne sera entreprise que si la condition est vérifiée.
Le choix des règles à utiliser est sous le controle d'un module qui initie des cycles comportant :

- un filtrage : sélection des règles utilisables eu égard aux faits actuellement connus (ces faits constituent la base des faits). Un module interprète, dans une phase d'identification, établit un lien entre les faits (résultats partiels obtenus) et le modèle, affectant à la correspondance ainsi établie un coefficient de plausibilité - <SHN 77>, système MYCIN : <LAU 82.1> -. Cette correspondance sert à déterminer quelles sont les règles dont les prémisses sont satisfaites.

- le choix : c'est la phase sensible qui conditionne "l'intelligence" du systéme. Elle sélectionne la règle à appliquer en utilisant un savoir-faire qui peut lui-meme etre contenu dans des règles (méta-règles). Les informations disponibles à un instant donné modifient donc le comportement ultérieur du système (notion de connaissance active - (FRE 77) -)

- l'exécution de la règle : la partie action est mise en oeuvre et modifie la base des faits.

ECRITURE DES REGLES

Pour décrire les règles de production il est nécessaire d'utiliser un langage adapté. Ainsi, aux notions pratiques que nous venons de décrire, on associe par dualité les concepts suivants relatifs au langage de description utilisé :

- syntaxe définissant le lien formel entre les signes du langage
- sémantique précisant les relations entre les signes et les objets utilisés par la structure proposée pour le langage - (PAB 74) -
- pragmatique gérant les relations entre les signes et l'interprétation proposée - (BAL 77), (NIE 81) -.

Dans ce formalisme une phrase du type sujet,verbe,compléments est une construction conforme à la syntaxe dans laquelle le sujet est un objet, le verbe une relation et les compléments des objets en relation avec le sujet : "la cheminée est sur le toit".

Remarquons que la distinction entre syntaxe et sémantique prete à discussion : la syntaxe décrit des aspects sémantiques statiques - (LIV 78, chapV), (CHA 81) -. Par exemple, dans le cas d'objets typés, la syntaxe precise le type des objets manipulés par les relations ce qui implique des restrictions quand à la sémantique possible des relations.

AVANTAGES DES SYSTEMES A REGLES DE PRODUCTION

La particularité de ces systèmes réside dans la forme déclarative utilisée pour représenter le savoir; par opposition aux structures hétérachiques où le controle de l'exécution est caché dans divers modules, ici on a une description explicite du savoir utilisé. Ainsi les systèmes experts développés selon cette méthode sont modulaires, aisément modifiables (par ajout, retrait ou modification d'une règle). Le problème principal reste alors la possibilité de donner, pour le domaine d'application considéré, des règles générales, correctement formalisées.

EN CONCLUSION

A l'issue de cette revue nous retirerons quelques principes qui nous ont guidés dans la conception de notre langage :

- Le traitement d'images est un moyen pour l'analyse de scène; il est un but lorsqu'il produit des images déstinées à l'utilisateur final (le clinicien dans le domaine médical).

- L'analyse de scène est un technique en soi insuffisante si elle n'utilise que les données iconographiques. Le recours au contexte est nécessaire.

- Pour les besoins de l'évaluation de la valeur diagnostique des examens, pour l'amélioration de leur interprétation, il est souhaitable d'intégrer des techniques d'apprentissage.

- Les futurs systèmes d'imagerie médicale devront communiquer entre-eux et intégrer des informations de toute nature. Il faut donc exporter à l'issue du traitement des informations sous une forme standardisée, indépendante des données qui ont servi à les obtenir.

Le langage que nous allons présenter dans la partie suivante veut donc etre une première étape tenant vers cet objectif en facilitant l'écriture de règles relatives à la partie traitement d'image d'un futur système expert d'aide au diagnostic utilisable dans les départements d'imagerie.

MEDECIM :

UN LANGAGE DE DESCRIPTION

LE LANGAGE DE DESCRITION DES IMAGES

Observant la démarche utilisée pour lire une image nous avons constaté qu'elle se décompose schématiquement en deux temps : extraction des informations utiles (objets normalement présents et anomalies) puis interprétation des ces données en faisant appel à des notions non iconographiques. En pratique il y a imbrication de ces deux étapes au niveau de leur mise en oeuvre.

Le langage que nous proposons permet de manipuler les informations extraites au fur et à mesure des besoins de l'interprétation. Il s'agit donc de décrire d'abord le contenu des images plutot que de traiter les images au sens T.I..

Nous avons choisi de matérialiser ces connaissances sous la forme de prédicats ce qui normalise la représentation utilisée et apporte une grande souplesse puisqu'à l'issue de l'étape de traitement d'image proprement dite, l'information circule sous une forme banalisée permettant d'injecter à ce niveau (qui est celui de l'interprétation) des connaissances de quelque nature que ce soit. L'utilisation du contexte clinique, épidémiologique et paraclinique devient possible.

L'accés aux primitives de traitement, nécessairement présentes pour accéder aux informations contenues dans les images, est subordonné à la définition préalable des objectifs. On impose par là de finaliser le traitement en définissant, avant l'exécution, son aboutissement . Cette contrainte pratiquement se traduit par une écriture des commandes débutant obligatoirement par la qualification, à l'aide de prédicats, des objets crées lors de l'éxecution de la commande. On retrouve là la démarche des utilisateurs d'images qui, à la différence des traiteurs d'images, ne veulent pas s'intéresser à la technique du traitement d'images mais seulement au résultat.

Le traitement d'image n'est toutefois pas seulement objet, il aussi sujet créant les documents finaux dont nous avons déjà dit l'importance pour les prescripteurs d'examens, intéressés à l'interprétation des résultats mais aussi à la lecture directe des documents.

Avant d'aborder la description du langage, concluons en disant qu'en d'autres termes il s'agit d'un langage orienté objet, non procédural (prenant en compte le "QUOI" et pas le "COMMENT").

LA STRUCTURE DU LANGAGE

Le langage comporte deux niveaux distincts, le bas-niveau relevant du traitement d'images et le haut-niveau manipulant les notions sémantiques.

LE NIVEAU LOGIQUE

Le langage repose sur le calcul des prédicats du premier ordre c'est à dire qu'une commande du langage est une expression logique. Exécuter une commande c'est calculer une valeur de vérité, évaluant pour cela les différents termes apparaissant dans l'énoncé de l'expression - <PAB 74> -.
Les termes sont des prédicats, c'est à dire des assertions décrivant les objets T.I.. On manipule donc non pas des images mais des épithètes pouvant leur etre attachées ce qui permet de définir un objet par l'ensemble de ses propriétés.

Dans le cadre de cette inroduction il faut noter que d'autres langages de programmation utilisent la logique pour représenter des connaissances et nous allons donc tenter de nous situer à cet égard en évoquant le plus connu d'entre-eux.

PROLOG

La classé des langages fondés sur la logique du premier ordre posséde un représentant bien connu avec PROLOG - <COL 83>, <COL 84> - qui a été développé il y a plus de 10 ans. Notre langage partage évidemment de nombreuses caractéristiques avec lui - <LAU 82.2> - :

- uniformité de la représentation des faits, règles d'inférence et structures de controle
- confusion de la base de règles et de la base de faits

Il s'en distingue notamment par l'absence de moteur d'inférence. PROLOG utilise le principe de résolution pour fournir d'un problème donné toutes les solutions; il met en oeuvre pour cela une approche combinatoire, retournant les propositions (clauses de HORN) en partant du résultat pour en démontrer la validité. Dans notre cas il existe sous-jacent un niveau T.I et la longueur des traitements asociés exclut d'entreprendre des traitements pouvant se révéler inutiles. Surtout nous travaillons dans un domaine tel que la structure associée au langage garantit la validité des formules manipulées - <PAB 74>, <LAU 84> -. Ceci veut dire, par exemple, que nous ne prenons pas en compte le cas hypothètique d'une requete utilisant une formule adaptée au traitement des images pulmonaires et initiée sur des images d'un autre organe. Il ressort de cette remarque que la solution est unique : une requete ne fournit qu'un résultat, vrai ou faux, au niveau logique; en revanche, elle modifie la segmentation de bas-niveau et peut créer plusieurs objets T.I..

Autre différence : nous travaillons sur des informations de deux types distincts (images et assertions) et nous utilisons pour le T.I. des objets typés. Les primitives T.I. doivent de plus etre compatibles avec le logiciel antérieurement fourni sur la machine nous disposons - PDP 11 sous RT11 modifié, FORTRAN IV -.

C'est donc pour toutes ces raisons que nous avons développé notre propre système, écrit en FORTRAN, conscients des avantages et des inconvénients de ce choix.

LE BAS-NIVEAU

Au niveau inférieur on retrouve des objets T.I. classiques et des procédures effectuant les traitements.

Comme nous avons dit, une commande du langage est une formule logique et c'est pourquoi ce niveau T.I. n'est accessible au programmeur que s'il a précédemment défini la structure du niveau logique de la formule selon les contraintes inposées par la grammaire.

Ici il n'y a que peu de particularités démarquant le langage MEDECIM de ceux que nous avons évoqués dans la partie III. Les modules assurant le T.I. peuvent etre rajoutés, au fur et à mesure des besoins et après compilation séparée, à la base des connaissances contenant le savoir-faire en T.I. (celle-ci consiste en un BLOCK DATA du langage FORTRAN résident en mémoire centrale pendant l'analyse syntaxique de la formule en cours d'interprétation).

```
EXP             ::=  <CONNECTEUR><EXP><EXP>
                    /<QUANTIFICATEUR><PRIMAIRE><EXP>
                    /<TERME>
TERME           ::=  <PREDIC2><OBJET><OBJET>
                    /<PREDIC1><OBJET>
                    /<BOOLEEN>
OBJET           ::=  <OPER2><OBJET><OBJET>
                    /<OPER1><OBJET>
                    /<PRIMAIRE>
CONNECTEUR      ::=  ET/OU
QUANTIFICATEUR  ::=  QUEL_QUE_SOIT/IL_EXISTE
BOOLEEN         ::=  VRAI/FAUX
PREDIC2         ::=  EST_INTERIEUR/EST_EXTERIEUR
                    /EST_A_DROITE/EST_AU_DESSUS
                    /EST_CONFONDU/EST_CONNECTE/EST_VOISIN
                    /APPARTIENT
                    /EST_SOMBRE
                    /EST_EGALE/EST_SUPERIEURE
PREDIC1         ::=  NOT
                    /ECRIRE/EDITER/TRACER/VISUALISER
                    /EST_CIRCULAIRE/EST_CONNEXE
OPER2           ::=  ARITH_IMAGE/ARITH_VALEUR/MASQUE
                    /ZOOMER/SEUILLAGE/FENETRER
                    /INTERSECTION/REUNION/EROSION/DILATATION
                    /DISTANCE
OPER1           ::=  SEL_ETUDE/SEL_IMAGE/EFACE_IMAGE
                    /LAPLACIEN/LISSAGE/PREWITT/SOBEL/SEGMENTE
                    /FERMETURE/OUVERTURE/ENCADRE/RECTANGLE
                    /EFFACE_PIXEL/CENTRE
PRIMAIRE        ::=  PATIENT/ETUDE
                    /IMAGE/FENETRE/REGION/PIXEL
                    /HISTOGRAMME/LIGNE/COURBE/VALEUR
```

DEFINITION DE LA GRAMMAIRE

Fig IV.1

LA GRAMMAIRE

Une formule du langage est un assemblage comportant des symboles logiques, des objets et des primitives T.I. et respectant les règles imposées par la grammaire (fig IV.1).
Celle-ci utilise une notation préfixée. Ce choix fait apparaître la structure des expressions en commençant par le niveau logique. On impose ainsi au programmeur un mode de raisonnement descendant, partant de la racine vers les feuilles. Par analogie on peut penser au processus de la vision dans lequel l'oeil effectue des traitements de bas-niveau alors que seule l'interprétation relève du domaine conscient. Ultérieurement nous prévoyons de modifier cette grammaire afin de faciliter la rédaction des formules si cette notation se révèle trop difficile à manipuler (à ce sujet, rappelons qu'un langage n'impose pas le choix d'une grammaire). Toutefois dans l'arborescence des formules cette stratification séparant la logique du traitement d'images subsistera bien sur.

A noter l'existence de quantificateurs, fonctionnelles équivalentes aux structures de controle, itérateurs et conditionnelles, dans les langages algorithmiques.

On trouvera une description des constantes, prédicats opérateurs et primaires, dans le paragraphe consacré au langage T.I..

LES SYMBOLES LOGIQUES

Les symboles du langage comportent :

- des connecteurs : ET, OU, XOR, EQV
- des quantificateurs : quantificateur existenciel et universel
- des variables :
- des prédicats : ce sont les constantes du langage,
- des individus : spécifiques du domaine (du T.I.).

A chaque prédicat et à chaque groupe fonctionnel est attachée son arité qui est le nombre des arguments qu'il manipule.

Notre langage est donc calqué sur le calcul des prédicats du premier ordre mais il faut apporter quelques précisions :

LES CONNECTEURS

Afin de faciliter la spécification des propriétés des objets nous avons ajouté, aux connecteurs ET et OU, le OU exclusif (noté XOR) et l'équivalence logique (notée EQV). En toute rigueur il suffirait d'un connecteur et de la négation pour signifier tous ces connecteurs mais leur ajout est une facilité pour l'utilisateur.

LES QUANTIFICATEURS

Les quantificateurs universel et existenciel ne doivent pas s'entendre au sens de la logique classique. En effet le domaine auquel appartient la variable quantifiée est actuellement restreint à l'ensemble des objets de meme type définis dans la formule courante. Cette approche intuitionniste se justifie actuellement par l'utilisation du langage pour le traitement automatisé de séquences d'images; dans cette optique le seul contexte à prendre en compte se limite aux images soumises à traitement et aux objets T.I. associés, puisque nous n'utilisons pas de base d'images.

Précisement, une formule du type

$$\forall x \in D, P(x)$$

signifie "pour tout x dans le domaine D, la propriété P(x) est vérifiée". Dans notre cas ceci nous amènerait à tester de manière exhaustive tous les individus de l'ensemble D pour évaluer la valeur de verité du prédicat P. Avec la restriction adoptée sur la portée des quantificateurs, limitant le domaine de la variable liée à l'ensemble des individus apparaissant dans la formule, on n'évaluera le prédicat P que pour les individus x effectivement crées lors de l'éxecution de la formule.

Le corollaire de cette restriction est que les règles de dualité liant les quantificateurs disparaissent:

$\langle \exists x , P(x) \rangle$ n'est plus synonyme de $\langle \neg \langle \forall x , \neg P(x) \rangle \rangle$

$\forall$ Quantificateur existenciel
$\exists$ Quantificateur existenciel
$\neg$ Négation

Dans une future utilisation intégrant des mécanismes de résolution on envisagera l'extension du domaine des variables quantifiées en les recherchant dans une base d'objets T.I. formant un contexte élargi; ainsi pour un type de scène donné, constituant le domaine d'une réalisation du langage, on pourrait vérifier qu'une formule est valide. Pour illustrer notre propos disons qu'étant donné le domaine des radiographies thoraciques de face, une formule décrivant au centre de l'image un empilement vertical de structures radio-opaques, grossiérement carrées, dont les bords verticaux sont concaves, dont la section... (la colonne vertébrale ?) serait certainement valide. Les quantificateurs autoriseront alors la validation de modèles décrivant un type d'image donné. Enfin cette extension impliquera des contraintes sur la portée des identificateurs sur lesquelles nous reviendrons.

NOT	BOOLEEN		BOOLEEN
TRACER	COURBE		BOOLEEN
VISUALISER	IMAGE		BOOLEEN
ECRIRE	IMAGE		BOOLEEN
EST_CONNEXE	REGION		BOOLEEN
EST_CIRCULAIRE	REGION		BOOLEEN
EDITER	VALEUR		BOOLEEN
EST_SOMBRE	REGION	IMAGE	BOOLEEN
APPARTIENT	PIXEL	LIGNE	BOOLEEN
EST_VOISIN	PIXEL	PIXEL	BOOLEEN
EST_A_DROITE	PIXEL	PIXEL	BOOLEEN
EST_AU_DESSUS	PIXEL	PIXEL	BOOLEEN
EST_CONNECTE	PIXEL	PIXEL	BOOLEEN
EST_CONFONDU	PIXEL	PIXEL	BOOLEEN
EST_INTERIEUR	PIXEL	REGION	BOOLEEN
EST_EXTERIEUR	PIXEL	REGION	BOOLEEN
EST_SUPERIEURE	VALEUR	VALEUR	BOOLEEN
EST_EGALE	VALEUR	VALEUR	BOOLEEN

PROFIL DES PREDICATS DU LANGAGE

Fig IV.2

L'EGALITE

L'égalité est utilisée au niveau de la manipulation des formules dans la logique (niveau du méta-langage, de l'exposition). Elle n'a pas sa raison d'etre ici puisque l'utilisation qui est actuellement faite des formules se réfère toujours à une réalisation du langage; en effet comme nous l'avons il s'agit, à ce jour, simplement d'extraire les éléments distingués, l'interprétation étant explicitement présente dans la formule.

LES PREDICATS

Les prédicats sont les opérateurs assurant la charnière entre la logique et le traitement d'images : acceptant des arguments de type objets T.I., ils rendent un résultat booléen. Ce sont eux qui expriment toutes les propriétés que l'on peut mettre en évidence pour caractériser les objets.
La liste des prédicats est citée dans la grammaire où nous notons PREDIC1 les prédicats d'arité 1, PREDIC2 les prédicats d'arité 2. De plus nous donnons (fig IV.2) le profil des prédicats existants à ce jour. La liste citée peut bien entendu évoluer s'il apparait des manques à ce niveau.

Les prédicats unaires décrivent des propriétés d'objets T.I. et les prédicats binaires s'adressent aux relations pouvant exister entre eux.

Exemples :
EST_CIRCULAIRE(REGION), EST_CONFONDU(PIXEL,PIXEL)

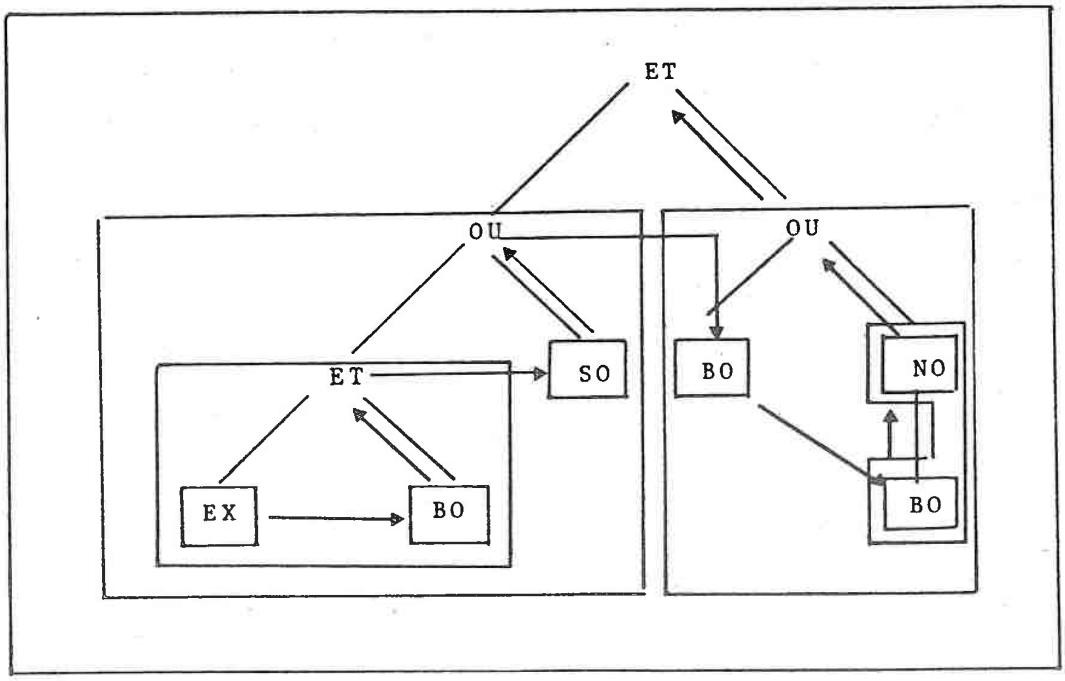
On remarque enfin, parmi les prédicats unaires, une classe à part comportant ECRIRE, EDITER, VISUALISER . Ces prédicats assurent une fonction de service et ne créent pas d'objets; ils agissent par leurs effets de bord et rendent toujours un résultat vrai. Ils ne sont là que pour donner plus de souplesse au langage.

EXEMPLE

Enfin pour clore cette présentation nous donnerons un exemple de formule logique du langage (fig IV.3). Cette formule pourrait s'écrire en notation infixée :

(((EX) ET (BO)) OU (SO)) ET ((BO) OU (NO (BO))) (1)

ET et OU sont les connecteurs - conjonction et disjonction -
NO est la négation
BO est un individu booléen
EX et SO sont des prédicats manipulant des objets T.I.
et dont les arguments ont été omis



Structure logique : exemple

fig IV.3

LE LANGAGE DE TRAITEMENT DES IMAGES

Le bas-niveau du système repose sur un langage de traitement d'images qui décrit les objets et les opérateurs les manipulant. Nous allons succesivement décrire les objets puis les opérateurs bien que la définition des premiers fasse nécessairement à quelques opérateurs pour illustrer l'exposé.

LES OBJETS T.I.

NOTION DE TYPE ABSTRAIT

Les objets sont représentés par des types abstraits c'est à dire des concepts définis formellement qui permettent en dehors de tout souci d'implémentation de préciser leur sens et la sémantique des opérateurs les manipulant <WUL 76>, <GUT 77>, <LIV 78>.

A chaque type on associe des attributs comme, dans le cas d'une REGION par exemple, sa surface ou sa couleur,...
Cette structure permet de distinguer deux classes parmi les attributs associés à un objet :

- ceux qui décrivent leur représentation interne
- ceux qui véhiculent la sémantique des opérations définies sur les objets.

Seuls ces derniers intéressent l'utilisateur et lui sont accessibles. L'avantage de cette formalisation est évident puisqu'il devient possible de manipuler des objets abstraits et d'écrire par exemple :

IMAGE = Somme (IMAGE,CONSTANTE)

sans se préoccuper des problèmes de mise en oeuvre.

SEMANTIQUE DES OPERATEURS ASSOCIES AUX TYPES ABSTRAITS

En soit l'exemple que nous avons cité n'a pas de sens; on peut tenter de lui en donner un en considérant que IMAGE désigne un tableau de nombres, que CONSTANTE dénote un scalaire et que Somme est un opérateur qui crée un nouveau tableau dont les cases sont obtenues en ajoutant au contenu des cases homologues du tableau initial le nombre dénoté par le second argument.
Ceci nous montre la nécessité de définir la sémantique associée aux types abstraits puisque la seule syntaxe ne peut y suffir.

Pour cela deux approches sont possibles, l'une axiomatique et l'autre opérationnelle.
La première précise explicitement pour tous les manipulateurs d'un type donné quels sont leurs profils (arité, type des arguments, type du résultat) et quels sont les résultats délivrés par ces opérateurs appliqués aux valeurs distinguées de l'objet. A l'évidence cet exercice, qui a été mené à son terme dans le cas de structures de base aussi simples que les piles, serait délicat s'agissant d'objets protéiformes comme ceux utilisé en T.I.. En outre la complétude de l'axiomatisation des types et sa cohérence seraient remise en cause à tout ajout d'un nouvel opérateur.
A l'inverse la spécification opérationnelle facilite l'implémentation puisqu'elle se réduit à l'écriture des procédures construisant et manipulant les objets. La mise en oeuvre de notions dont le niveau d'abstraction va croissant est de plus facilitée par la répartition des taches à des sous-programmes communiquant selon des protocoles rigoureux. Par exemple pour garantir que le type FENETRE est toujours défini par ses deux coins opposes, (en haut, à gauche; en bas, à droite) il suffit qu'un sous-programme unique soit chargé de toutes les manipulations de FENETRE qu'effectue le système. On peut alors manipuler le type FENETRE sans soucis d'implémentation.

EXEMPLE DE MISE EN OEUVRE DES TYPES ABSTRAITS

Dans un récent article BELAID précise la notion de type abstrait appliquée au traitement d'images, mettant en exergue les avantages de cette approche <BEL 84> :

- connaissance explicite de chaque type de donnée et des opérateurs associés
- facilité d'élaboration d'une structure liant les objets ou leurs parties entre eux.

Pour illustrer ce dernier point nous allons développer l'utilisation de la FENETRE dans notre langage :

Une FENETRE est un rectangle dont la fonction est de sélectionner une partie d'une IMAGE; reprenant l'idée de RADHAKRISHNAN, - <RAD 81> -, nous l'utilisons pour restreindre les traitements à une partie d'une IMAGE ce qui répond à un souci d'efficacité bien sur mais donne aussi un moyen de focaliser l'attention sur cette FENETRE au niveau du module de base exécutant le traitement.
La FENETRE peut etre rattachée à l'IMAGE de deux façons :

- soit explicitement par l'opérateur FENETRER (IMAGE,FENETRE) qui délivre une IMAGE
- soit etre un attribut hérité. Chaque opérateur rendant une IMAGE et dont l'un au moins des arguments est une IMAGE attache à l'image créée la FENETRE éventuellement attachée à l'IMAGE argument.

S'agissant de l'utilisation des FENETRE nous avons adopté la règle suivante :

toute opération utilisant une IMAGE comme paramètre ne considère que la partie de l'IMAGE contenue dans la FENETRE associée si elle existe.

Cette règle revient à l'utilisation implicite d'un argument supplémentaire pour tous les opérateurs manipulant des images. Dans un langage déclaratif l'utilisation de ces paramètres implicites est essentielle car elle allège beaucoup l'écriture des formules du langage. Il faut en revanche que les paramètres implicites aient une valeur par défaut lorsqu'il sont (implicitement) référencés avant d'avoir été (explicitement) instanciés dans la formule; ainsi toute IMAGE, lors de sa création, se voit attacher par défaut une FENETRE de taille égale à la sienne propre. Le revers du confort apporté par ces arguments implicites est le risque d'altérer le sens du traitement en créant des objets non explicitement décrits dans la formule; nous allons revenir plus loin sur ce problème.

Les attributs hérités, tels que la FENETRE, se propagent des feuilles vers la racine de la formule évitant d'avoir à recréeer des liens explicites. En d'autres termes la portée de la FENETRE s'étend de la feuille qui la crée à tous les ascendants en ligne directe qui sont du type IMAGE et sont "père" d'IMAGE.

Il faut également noter le point suivant relatif à la sémantique des FENETRE :

si deux ou plusieurs des arguments d'un créateur d'IMAGE sont du type IMAGE l'opérateur synthétisera la FENETRE résultant de l'intersection des FENETRE attachées à ses arguments. Ceci signifie que l'on a utilisé une nouvelle fenetre qui va elle meme se propager. Toutefois cet attribut qui n'a pas été créé explicitement dans la formule n'est pas accessible aux autres opérateurs en temps que tel. Ce n'est pas un objet du type FENETRE, mais un attribut herité des parents de l'IMAGE. Ainsi seuls les opérateurs manipulant cette IMAGE utiliseront la nouvelle FENETRE.

Remarquons enfin que le lien FENETRE-IMAGE est dynamique; il peut etre défait par la création d'un lien avec une autre FENETRE, couvrant eventuellement toute l'image.

Sur cet exemple détaillé nous avons montré comment la sémantique d'un objet pouvait évoluer : d'un rectangle appliqué sur une IMAGE, nous avons créé la notion de centre d'intéret en restreignant tous les futurs traitements de cette IMAGE à l'intérieur de ce rectangle. Parallèlement il est apparu qu'il ne fallait, pour garantir une sémantique fiable, autoriser les manipulations d'un type objet donné que par un seul module. A la fiabilité, cette approche associe la modularité permettant des modifications aisées de la sémantique.
Enfin nous avons abordé le problème des effets de bords, comme celui de la création par le système d'une FENETRE nouvelle à partir de l'intersection de deux FENETRE. Tous ces aspects rendent compte de la rigueur nécessaire pour une implémentation correcte de types prédéfinis.

LES TYPES DE OBJETS DU LANGAGE

La liste des types actuellement définis apparait sur la figure IV.4.

On remarquera que le type BOOLEEN n'appartient pas au traitement d'images. Il permet de considérer un prédicat comme un objet au meme titre qu'une REGION par exemple (cf remarque introductive concernant l'homogénéité des représentations dans les langages à base de logique).

Le type PATIENT est une notion reprise à partir du logiciel, fourni par les laboratoires ADAC, préexistant sur la machine que nous utilisons. Un PATIENT est une collection d'ETUDE effectuées chez un meme sujet.

Une ETUDE est une série d'IMAGE ayant un lien logique entre-elles : séquence dynamique d'IMAGE, ensemble d'IMAGE obtenues par le calcul à partir d'un meme examen initial,....

BOOLEEN : objet pouvant prendre deux valeurs distinctes.

PATIENT : collection d'ETUDE rassemblées sous un meme nom.

ETUDE : série d'IMAGE logiquement associées.

IMAGE : tableau carré de PIXEL.

FENETRE : rectangle déterminant deux REGIONS disjointes sur un IMAGE.

REGION : ensemble de points d'une IMAGE.

HISTOGRAMME : histogramme des valeurs de gris d'une partie d'une IMAGE.

PIXEL : triplet désignant une case d'une IMAGE et sa valeur de gris.

LIGNE : collection de points ayant chacun un précédent et un suivant à l'exception de deux d'entre-eux.

COURBE : graphe des variations du contenu d'une REGION successivement associée aux différentes IMAGES d'une ETUDE.

VALEUR : nombre entier, réel ou booléen.

LES TYPES DES OBJETS T.I. fig IV.4

Une IMAGE est un tableau carré de valeurs entières. Ces valeurs représentent un paramètre variable dépendant du mode d'obtention de l'IMAGE.

Le type FENETRE a été détaillé plus haut.

Une REGION est un ensemble de positions dans une IMAGE. Elle est donc une notion géométrique. Toutefois par un mécanisme d'association implicite déjà décrit les REGION sont toujours associées à des IMAGE. On verra plus loin que les créateurs de REGION utilisent des arguments eux-memes liés à des IMAGE.

Un HISTOGRAMME est une courbe histogramme des valeurs contenues dans une partie d'IMAGE. Aujourd'hui ce type n'est pas encore réellement mis en oeuvre.

Un PIXEL est un triplet désignant une case d'un tableau IMAGE et sa valeur.

Une LIGNE est une collection de positions dans une IMAGE, telle que chacune posséde un précédent et un suivant, à l'exception des deux d'entre-elles dont l'une a un suivant seulement et l'autre un précédent seulement. Ultérieurement on pourra grace aux LIGNE utiliser deux approches duales pour définir les REGION par leur contenu ou leurs contours.

COURBE : graphe des variations du contenu d'une REGION successivement calculé pour les différentes IMAGE d'une ETUDE.

VALEUR : nombre entier, réel ou booléen.

Nous ne détaillerons pas plus avant ces objets puisqu'il s'agit de type classiques; en revanche il faut noter que nous pensons rajouter les types suivants :

MAP : échelle de gris, utile pour la production de documents particuliers (extinction de certaines parties de l'image eg)

RAPPORT : texte alphanumérique; ce type pourrait contenir des informations telles que l'état civil du sujet ou etre le résultat fourni par un opérateur (cf partie V).

LES OPERATEURS T.I.

Pour chaque opération on définit une arité et un profil :
- l'arité est le nombre des arguments nécessaires à l'opérateur
- le profil spécifie le type des objets argument et le type de l'objet résultat.

Parmi les opérateurs il faut distinguer les constructeurs qui produisent un type objet différent de celui ou ceux de leurs arguments. Inversement les manipulateurs modifient les objets sans générer de résultat de type nouveau.
Enfin pour chaque type objet il existe un opérateur "constant" (d'arité zéro) qui crée l'objet. Ces opérateurs jouent un role important dans les mécanismes de correction d'erreur.

La liste des opérateurs apparait dans la figure IV.5

LES OPERATEURS CONSTANTS

Les créateurs de BOOLEEN, de PATIENT ou d'ETUDE utilisent la console de dialogue pour obtenir de l'opérateur la valeur de l'objet.

D'autres opérateurs créant une FENETRE ou une REGION utilisent un light-pen pour entrer la définition de l'objet.

Enfin l'opérateur IMAGE est pour l'heure fictif. Il pourrait dans l'avenir permettre l'acquisition d'une nouvelle IMAGE à partir des périphériques associés aux système.

LES OPERATEURS UNAIRES

Nous énumérerons seulement les possibilités offertes :

choix d'une ETUDE ou d'une IMAGE,
divers lissages d'une IMAGE,
mise à zéro d'une IMAGE ou d'un PIXEL,
transformation d'une FENETRE en REGION et inversement,
extraction du centre d'une REGION.

Identificateur	Arguments		Résultat
BOOLEENC			BOOLEEN
PATIENTC			PATIENT
ETUDEC			ETUDE
IMAGEC			IMAGE
FENETREC			FENETRE
REGIONC			REGION
HISTOGRAMMEC			HISTOGRAMME
PIXELC			PIXEL
LIGNEC			LIGNE
COURBEC			COURBE
VALEURC			VALEUR
ETUDE	PATIENT		ETUDE
ENCADRE	REGION		FENETRE
LECTURE	ETUDE		IMAGE
ZEROIM	IMAGE		IMAGE
LAPLACIEN	IMAGE		IMAGE
LISSAGE	IMAGE		IMAGE
PREWITT	IMAGE		IMAGE
SEGMENTE	IMAGE		IMAGE
SOBEL	IMAGE		IMAGE
ZEROPX	PIXEL		PIXEL
CENTRE	REGION		PIXEL
REMPLIT	FENETRE		REGION
FERMER	REGION		REGION
OUVRIR	REGION		REGION
FENETRER	IMAGE	FENETRE	IMAGE
DIVIM	IMAGE	IMAGE	IMAGE
MULIM	IMAGE	IMAGE	IMAGE
PLUSIM	IMAGE	IMAGE	IMAGE
MOINSIM	IMAGE	IMAGE	IMAGE
DISTANCE	PIXEL	PIXEL	VALEUR
MASQUE	IMAGE	REGION	IMAGE
INTERSECTION	REGION	REGION	REGION
REUNION	REGION	REGION	REGION
DIVVAL	IMAGE	IMAGE	IMAGE
MULVAL	IMAGE	IMAGE	IMAGE
PLUSVAL	IMAGE	IMAGE	IMAGE
MOINSVAL	IMAGE	IMAGE	IMAGE
ZOOMER	IMAGE	VALEUR	IMAGE
SEUILLAGE	IMAGE	VALEUR	REGION
DILATER	REGION	VALEUR	REGION
ERODER	REGION	VALEUR	REGION

LISTE DES OPERATEURS Fig IV.5

LES OPERATEURS BINAIRES

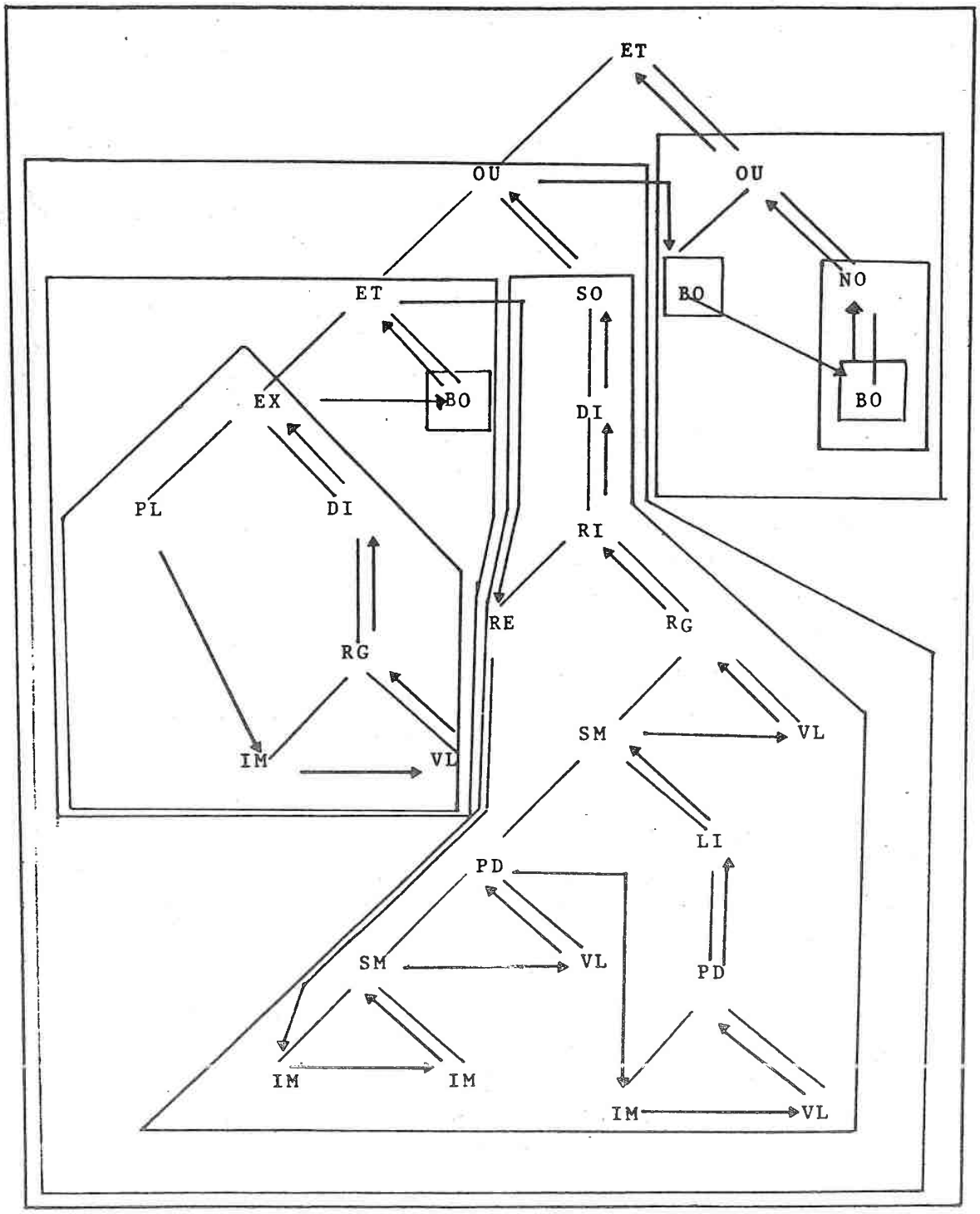
Ici on trouve :

arithmétique entre IMAGE et IMAGE, entre IMAGE et VALEUR,
seuillage et zoom d'une IMAGE,
érosion et dilatation d'une REGION,
opérations entre REGION,
mise à zéro d'une partie d'une IMAGE.

Nous n'avons pas à ce jour affiné les opérateurs. La littérature propose de nombreux algorithmes : <AYA 83>, <CLA 83>, <CUR 79>, <DUF 81>, <GRA 71>, <HOR 81>, <HOU 62>, <HUE 73>, <KES 83>, <MAR 76>, <MAR 72>, <PON 83>, <SCH 78>, <SER 82>. En outre des ouvrages entiers sont consacrés au T.I. : <AND 77>, <BAL 82>, <CAS 77>, <GON 77>, <HAN 78>, <HUA 75>, <NIE 81>, <PRA 78>, <PRE 81>, <ROS 76.1>, <ROS 76.2>.

Ainsi donc cet aspect est important mais banal et sans plus détailler nous illustrerons cette présentation de l'aspect T.I. de MEDECIM par un exemple (fig IV.6) qui montre l'articulation entre le niveau logique et le T.I.. En fait on développe ici la formule de la figure IV.3 en rajoutant au niveau inférieur la description de la procédure utilisée pour créer les arguments des prédicats SO et EX :

PL : opérateur constant définissant un PIXEL
IM : opérateur constant définissant une IMAGE
VL : opérateur constant définissant une VALEUR
RG : seuillage
DI : dilatation
EX : prédicat indiquant si un PIXEL est intérieur à une REGION
SM : opérateur sommant deux IMAGE
PD : opérateur faisant le produit d'une IMAGE par une VALEUR
LI : opérateur lissant une IMAGE
RE : opérateur constant définissant une REGION
RI : intersection de deux REGION
DI : dilatation d'une REGION
SO : prédicat indiquant si une REGION est sombre



Une formule du langage : exemple

fig IV.6

MISE EN OEUVRE

Cet aspect est, dans la version actuelle, rudimentaire. Il permettra toutefois au système de fonctionner afin de tester la faisabilité de l'approche choisie pour les divers modules de T.I..

LES REQUETES

Le fonctionnement de MEDECIM est en ce moment limité au seul mode immédiat; l'utilisateur choisit parmi les formules dont il dispose celle qu'il désire voir exécuter.
Il exprime donc une requete que le système satisfait en deux temps :

- interprétation de la formule d'abord,
- exécution ensuite, commençant par la partie T.I. suivie du calcul de la valeur de vérité de la partie logique de la commande.

Il est donc possible ainsi de dépouiller des examens, conformément aux protocoles employés jusqu'alors, simplement en écrivant la formule correspondante. On n'exploite alors que l'aspect T.I..

La version future comportera un moteur d'inférences qui sera capable d'activer les commandes en emettant lui-meme les requetes. Il y aura alors trois modes de fonctionnement :

- manuel, sur requete comme aujourd'hui, pour prendre en charge les besoins du T.I. en temps que tel.
- automatique, controlé par le niveau supérieur, pour permettre l'intégration des images à un système expert d'aide au diagnostic.
- en mode pas à pas pour la mise au point des commandes. A noter que la contrainte selon laquelle toute arborescence doit avoir une racine logique est factice volontairement imposée par l'analyseur syntaxique; il sera donc possible, bien sur, de tester des fragments de commandes purement T.I. pour aider au développement sur MEDECIM.

ECRITURE DES FORMULES

Ici aussi la réalisation présente devra etre améliorée.
Les formules sont contenues dans des variables initialisées lors de la compilation (via des instructions DATA de FORTRAN). La rédaction d'une formule passe donc actuellement par l'écriture de quelques lignes de FORTRAN et la compilation d'un module.
La syntaxe du langage n'utilise pas de séparateur et l'écriture est linéaire, utilisant des mnémoniques à deux lettres, peu intelligible.

Nous prévoyons de réaliser un module éditeur de commandes dont le role sera quadruple :

- permettre l'utilisation d'identificateurs de longueur quelconque par l'écrivain des commandes. Ainsi les mnémoniques seront plus faciles à mémoriser.
- compacter ces identificateurs pour une représentation interne plus efficace.
- lever, au niveau de l'écriture, l'obligation d'utiliser une syntaxe préfixée fort peu naturelle pour revenir à une notation infixée plus conforme à nos habitudes.
- gérer une base de commandes. Les fonctionnalités associées seront la lecture, l'édition et le stockage des commandes mais aussi l'optimisation des commandes (recherche des segments communs aux commandes eg).

DETAIL DU FONCTIONNEMENT

Ici nous allons donner quelques explications relatives à l'analyse syntaxique des formules et à leur exécution afin de démontrer quelques principes fondamentaux qui doivent permettent de faire de ce langage un outil opérationnel.

L'INTERPRETE

Le module interpréteur crée les informations nécessaires à l'execution des formules du langage.
Il accepte en entrée des commandes. Il produit en sortie d'une part un arbre de parcours de la formule et d'autre part une table des objets décrits par la formule. De plus il vérifie que la racine de l'arborescence est un objet logique et qu'il y a bien concordance entre les arguments et les profils des divers opérateurs de la commande. En cas d'erreur, il rejette purement et simplement la formule. Lorsque sera réalisé l'éditeur de commandes on envisagera une correction interactive de la syntaxe des formules, rendue alors d'autand plus utile que l'exécution ne se fera plus en mode immédiat.

L'ARBRE DE PARCOURS DE LA FORMULE

L'interprète gére deux arbres permettant de parcourir la formule selon la stratégie en bas et à gauche d'abord.
Le premier arbre permet le parcours de la commande T.I. et le second ne décrit que la partie logique de la formule. Ces deux arbres seront utilisés par deux modules différents du processus controlant l'exécution.

Les figures IV.3 et IV.6 explicitent ces arbres, montrant le sens de parcours utilisé.

LA TABLE DES OBJETS

Chaque opérateur, prédicat, connecteur ou quantificateur se voit donc attacher un objet, du type idoine, dans cette table par l'interprète. Ainsi, dès avant l'exécution, les objets existent mais ne sont pas instanciés.

Chaque objet posséde un nom. Celui-ci peut etre affecté de manière automatique par l'interprète qui, lors de la l'analyse syntaxique, assigne les noms suivant l'ordre alphabétique. Mais cette affectation automatique ne se fait que si le nom n'est pas imposé par la commande; on réalise ceci par simple apposition d'un identificateur (de deux lettres), différent des mots réservés du langage, à un symbole opératoire; l'objet que crée le symbole portera alors ce nom. Les noms peuvent etre utilisés ailleurs dans la commande pour invoquer les objets associés; ils permettent ainsi d'alléger l'écriture.
Sous réserve d'étendre la portée des noms, il sera possible dans le futur de créer une bibliothèque de commandes qui seront appelables par leur nom.

LE CONTROLE D'EXECUTION

Ce module supervise l'exécution grace aux arbres de parcours fournis par l'interprète. Actuellement il est directement activé après l'interprétation des formules puisque nous travaillons en mode immédiat. Il comporte un module balayant la partie logique de la formule, un module controlant la partie T.I., un module responsable du passage des arguments et de l'affectation des résultats aux objets.

LE PARCOURS DE L'ARBRE DES COMMANDES

Le balayage de l'arbre se fait de gauche à droite et en profondeur d'abord.

Il est optimisé de manière à ne parcourir que les branches nécessaires : ainsi si la première branche d'un OU est vraie, la seconde n'est pas évaluée, et de meme si la première branche d'un ET est fausse. Notons que dans le cas où une branche est fausse il n'est pas indispensable de détruire les objets produits pendant le parcours de ce segment; en effet les objets T.I. aussi bien que les prédicats associés gardent leur sens et le seul risque est un encombrement de la table des objets. Ceci est à mettre en balance avec le temps de calcul utilisé pour créer ces objets et peut justifier de les conserver.

D'autre part, si l'évaluation d'une partie de l'arbre est impossible parce que se référant à un objet non instancié, l'exécution passe à la branche suivante et un nouveau balayage de l'arbre est initié après le premier. Inversement avant l'exécution d'une quelconque procédure il existe un test qui détecte si l'objet à créer est déjà instancié; si c'est le cas, la procédure n'est pas activée et le balayage continue.

LES QUANTIFICATEURS

Le mécanisme permettant d'initier un nouveau cycle de parcours de l'arbre est le meme qui sert pour les quantificateurs; nous avons dit que la variable quantifiée pouvait etre remplacée par tous les objets de meme type apparaissant dans la formule. Ainsi, au niveau de l'exécution, les quantificateurs se traitent par un parcours itératif de l'arbre.
Dans ces divers cas, un mécanisme limite les itérations pour prévenir les rebouclages sans fin possibles si, par exemple, on référence un objet par son nom sans jamais l'instancier dans la formule
- l'analyseur syntaxique ne détecte pas ce type d'incohérences -.

Pour terminer notons que les connecteurs et les quantificateurs sont directement évalués et qu'on ne génére donc pas de code exécutable pour eux.

L'EXECUTION AU NIVEAU T.I.

Les prédicats et leur opérateurs T.I. sont mis en oeuvre sous la forme d'une bibliothèque de sous-programmes qui reçoivent tous leur arguments selon un protocole unique. Ainsi il n'ont pas accès à la table des objets mais seulement à des copies et l'affectation à la table des objets se fait par un module unique après vérification de la bonne fin de la procédure.

LES ATTRIBUTS

Lors de l'instanciation des objets, les attributs associés à l'objet peuvent ou non etre évalués. Certains concernant l'implémentation sont toujours déterminés à la création de l'objet; ainsi en est-il de l'attribut ADRESSE dans l'objet REGION qui permet de la localiser en mémoire.

```
REGION      ::=   <ID,LIEN_IMAGE,ADRESSE,MOYENNE,VARIANCE
                  ,CENTRE,PERIMETRE,SURFACE>
```

A l'inverse le calcul de la MOYENNE est subordonné à la création préalable d'un LIEN_IMAGE attachant la REGION à une IMAGE. De plus le calcul est long, et c'est pourquoi il ne sera fait que lorsqu'un opérateur (comme le prédicat EST_SOMBRE) fera référence à cet attribut.

LE TRAITEMENT DES ERREURS

Afin de rendre le système opérationnel et en contradiction avec notre ambition d'automatiser les traitements, nous l'avons doté d'une procédure de rattrapage des erreurs sous controle de l'opérateur.
Si, au niveau logique, il ne peut pas y avoir d'erreur mais seulement des assertions fausses, au contraire, au niveau T.I., le risque est grand de voir des procédures échouer (segmentation automatique par exemple). Afin de récupérer ces erreurs, le controle est capable de

les détecter; dans ce cas il remplace automatiquement toute l'arborescence sous-jacente au niveau de l'erreur, celle-ci incluse, par l'opérateur constant produisant l'objet du meme type que celui nomalement créé par la procédure défaillante.

Reprenant l'exemple de la figure IV.6 nous supposerons que l'opérateur "DI" qui apparait comme argument de l'opérateur "EX" fournisse un résultat en erreur. Dans ce cas automatiquement MEDECIM déroutera l'exécution insérant un opérateur "RE" qui demande à l'opérateur de tracer au light-pen une région d'intéret.

Ceci montre l'utilité des opérateurs d'arité nulle qui intrinséquement permettent la création d'objets ex-nihilo.

Voici donc schématiquement présentées les caractéristiques du langage. Nous allons maintenant envisager quelques extensions possibles.

DEVELOPPEMENTS ULTERIEURS

CONCLUSION

DEVELOPPEMENTS ULTERIEURS - CONCLUSION

Le système MEDECIM doit d'abord faire la preuve du bien fondé de l'approche utilisée pour décrire les images. Ceci se fera par une utilisation de type T.I. classique permettant de valider les primitives de traitement et les formules.

Ensuite il évoluera vers une version plus ergonomique en s'appuyant sur des outils tels que l'éditeur pour le développement et la mise au point d'une base de connaissances rédigée sous la forme de commandes. Ce sont ces aspects que nous avons évoqué dans le paragraphe précédent.

S'agissant de MEDECIM, les points à étudier porteront aussi sur la possibilité d'exprimer des contraintes sur les attributs d'un objet. Ceci sera utile dans une approche utilisant un mécanisme de résolution pour trouver un objet vérifiant un ensemble de conditions. Bien sur nous aurons à résoudre ce type de problème lorsque sera adjoint à MEDECIM un moteur d'inférences capable de le contrôler.
Une possibilité pour résoudre ce problème serait d'utiliser une opérateur COMPARE capable de créer par exemple un objet du type RAPPORT décrivant les similitudes et les dissemblances entre ses deux arguments dont l'un serait un modèle et l'autre un objet testé.

CONCLUSION

Le langage MEDECIM est opérationnel pour tout ce qui concerne la partie logique (interprète et contrôle, mécanismes d'itération, de rattrapage d'erreur,...). Les primitives T.I. citées sont écrites mais n'ont pas encore fonctionné sous le contrôle du haut-niveau.

Dans ses annexes ce rapport présentera des exemples concrets d'utilisation du système.

Il reste à améliorer la partie traitement d'images afin d'en faire un outil robuste, et surtout il faut maintenant créer les outils capables de manipuler les assertions que MEDECIM permet d'extraire des images

ANNEXES - BIBLIOGRAPHIE

BIBLIOGRAPHIE

BIBLIOGRAPHIE

AND 77 : ANDREWS A.C. HUNT B.R.
Digital image restoration
Prentice Hall, Englewood Cliffs, New Jersey 1977

ASH 77 : ASHCROFT E.A. WADGE W.W.
LUCID, a non procedural language with iteration
Comm of the A.C.M.,1977, 20, 519-526

AYA 83 : AYACHE N.
Analyse d'images
Traitement et analyse d'images numériques
E.S.E. Session de perfectionnement L801, 1983

BAL 77 : BALLARD D.H., BROWN C.M., FELDMAN J.A.
An approach to knowledge-directed image analysis
I.J.C.A.I.-77, 1977, 2, 664-670

BAL 82 : BALLARD D.H., BROWN C.M.
Computer vision
Prentice Hall 1982

BAL 81 : BALSTON D.M.
A high-level language for constructing image processing commands
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for image processing", 1981, Academic Press, 99-116

BEL 83 : BELAID A.
SAPIN : système d'aide à la programmation du traitement d'images numérisées
Rapport interne n° 83-R-032, CRIN, Université de NANCY I, 06-83

BEL 84 : BELAID A.
Un langage de traitement d'images fondé sur des concepts de types abstraits
T.S.I., à paraitre

BOL 83 : BOLT R.A.
Les images interactives
La révolution des images
La Recherche, Mai 1983, n° spécial 144, 678-686

CAS 83 : CASSIGNAC D.
SAPIN-NI
Un système hautement interactif pour le traitement des images numérisées
Rapport de DEA, 83-R-061, CRIN Nancy, 1983

CAS 77 : CASTELMAN R.R.
Digital image processing
Prentice Hall, Englewood Cliffs, New Jersey 1977

CAS 84 : CASTIEL A.
Imagerie médicale : la nouvelle donne
Sciences et Techniques, 1984, 6, 36-45

CHA 81 : CHANG N.S., FU K.S.
A study on parallel parsing of tree languages and its application to syntactic pattern recognition
In : M. ONOE, K. PRESTON, A. ROSENFELD, "Real-time/parallel computing; Image analysis", 1981, Plenum Press, 107-129

CHE 83 : CHEVALIER P., GUYE O., ROUX P.
Revue sur le traitement des images industrielles
Traitement et analyse des images numériques
E.S.E. Session de perfectionnement L801, 1983

CLA 83 : CLARA F.
Approximation et implémentation de filtres bi-indiciels.
Restautration d'images
Traitement et analyse d'images numériques
E.S.E. Session de perfectionnement L801, 1983

CLA 83 : CLARA F.
Etude du seuil de visibilité en vision chromatique humaine
Traitement et analyse d'images numériques
E.S.E. Session de perfectionnement L801, 1983

CLA 83 : CLARA F.
Outils mathématiques d'amélioration d'images
Traitement et analyse d'images numériques
E.S.E. Session de perfectionnement L801, 1983

COL 83 : COMERAUER A., KANOUI H., VAN CANEGHEM M.
Prolog, bases théoriques et développements actuels
T.S.I., 1983, 2, 271-311

COL 84 : COLMERAUER A.
Prolog, langage de l'intelligence artificielle
La Recherche, 1984, 158, 1104-1114

COM 84 : COMET M., GODART J.
La cardiologie nucléaire
La Recherche, 1984, 159, 1216-1226

CRE 83 : CREHANGE M., AIT HADDOU A., BOUKAKIOU M., DAVID J.M., FOUCAULT O., MAROLDT J.
EXPRIM : an expert system to aid in progressive retrevial from a picturial and descriptive data base
Special workshop on new applications of databases : joined with ICOD-2 (Second international conference on database)
August-September 1983 - CAMBRIDGE - (England)

CRO 78 : CROCUS
Systèmes d'exploitation des ordinateurs
DUNOD, 1978

CUR 79 : CURIEN B.
Méthode d'analyse des images scintigraphiques dynamiques
Rapport de D.E.A, 1979, Université de NANCY

DAL 81 : DALLE P., DEBORD P., CASTAN S.
Système d'analyse et de comprehension de scènes par ordinateur : SASCO
3ème Congrès de reconnaissances des formes et d'intelligence artificielle, AFCET, NANCY 1981, 297-308

DEB 82 : DEBORD P.
Contribution à la définition du système d'analyse de scène SASCO et réalisation de l'interpréteur minimal
Thèse de 3ème cycle, Univ. P. Sabatier, Toulouse, Oct-82

DOU 81 : DOUGLASS R.J.
MAC : a programming language for asynchronous image processing
In M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for Image Processing", 1981, Academic Press

DUF 81 : DUFF M.J.B.
The elements of digital picture processing
In : M. ONOE, K. PRESTON, A. ROSENFELD, "Real-time/parallel computing; Image analysis", 1981, Plenum Press, 1-9

DUF 82 : DUFF M.J.B.
Parallel algorithms and their influence on the specification of application problems
In : K. PRESTON, L. UHR, "Multicomputers and image processing, Algorithms and programms", 1982, Academic Press, 261-274

ENO 81 : ENOMOTO H., KATAYAMA T., TONEZAKI N., MIYAMURA I.
Image data modeling and language for parallel processing
In : M. ONOE, K. PRESTON, A. ROSENFELD, "Real-time/parallel computing; Image analysis", 1981, Plenum Press, 95-105

FAU 76 : FAUGERAS O.
Digital color image processing and psychophysics within the framework of a human visual model
Ph. D. Thesis, June 76, University of Utah, USA

FER 84 : FERRETTI M.
Visionique : des yeux pour les robots
Sciences et Techniques, 1984, 1, 34-47

FRE 77 : FREI W. CHEN C.C.
Fast boundary detection : a generalisation and a new algorithm
IEEE Trans. on Computers, Vol C-26, N°10, October 1977, 988-998

FRE 77 : FREUDER E.C.
A computer system for visual recognition using active knowledge
IJCAI-77, 1977, 671-677

GON 77 : GONZALES R.C. WINTZ P.
Digital image processing
Addison-Wesley, Massachusetts 1977

GRA 81 : GRANLUND G.H.
GOP: a fast and flexible processor for image analysis
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for Image Processing", 1981, Academic Press, 179-187

GRA 71 : GRAY S.B.
Local properties of binary images in two dimensions
IEEE Trans. on Computers, 1971, C-20, 5, 551-561

GUT 77 : GUTTAG J.
Abstract data types and the development of data structures
Comm of the ACM, 1977, 20, 396-404

HAN 78 : HANSON A. RISEMAN E.
Computer vision systems
Academic Press, New York 1978

HOR 81 : HORAUD P.R., COULON P.Y.
Un algorithme pour détecter des objets à formes complexes dans une image
3ème Congrès de Reconnaissance des formes et Intelligence artificielle, AFCET, NANCY, 1981, 213-224

HOU 62 : HOUGH P.V.C.
Methods and means for recognizing complex patterns
US Pat. 3069654, Washington, DC, Dec (1962)

HUA 75 : HUANG T.S.
Picture processing and digital filtering
Springer Verlag, Berlin 1975

HUE 73 : HUECKEL M.H.
A local visual operator which recognizes edges and lines
J ACM 20, 1973, 4, 634-647

IMB 83 : IMBERT M.
La neurobiologie de l'image
La révolution des images, La Recherche, 1983, n° 144, 600-613

JOH 81 : JOHNSTON E.G.
PAX II picture processing system
In : M.J.B. DUFF, S. LEVIALDI, "Picture processing and psychopictorics", 1981, Academic Press

KEN 80 : KENNETH E.B.
Architecture of a massively parallel processor
CH1494-4/80/0000-0168 $00.75 c 1980 I.E.E.E., 168-173

KES 83 : KESKES N.
Detection de contour dans les images naturelles; Textures
Traitement et analyse d'images numériques
E.S.E. Session de perfectionnement L801, 1983

KLI 77 : KLINGER A. FU K.S. KUNII T.L.
Data structures, computer graphics and pattern recognition
Academic Press, New York 1977

KNU 68 : KNUTH D.
Semantics of context-free languages
Math. Systems Theory 2 ,1968

KOS 74 : KOSTER C.H.A.
Two-level grammars
In : F.L. BAUER, J. EICKEIL, " Compiler construction, an advanced course", 1974, Springer Verlag, LNCS, 21, 146-156

KOW 83 : KOWALSKI R.
Logic for expert systems
BCS Conference on Expert Systems, December 1983

KUL 81 : KULPA Z.
Picasso, Picasso-show and Pal
A Development of a High-level Software System for Image Processing
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for Image Processing", 1981, Academic Press, 13-24

LAU 84 : LAURENT J.P.
La structure de controle dans les systèmes experts
T.S.I., 1984, 3, 161-177

LAU 82 : LAURIERE J.L.
Représentation et utilisation des connaissances : les systèmes experts
T.S.I., 1982, 1, 25-42

LAU 82 : LAURIERE J.L.
Représentation et utilisation des connaissances : Représentation des connaisances
T.S.I., 1982, 1, 109-133

LEG 59 : LEGRAND Y.
Les yeux et la vision
DUNOD, 1959

LEV 81 : LEVIALDI S., MAGGIOLO A., NAPOLI M., TORTORA G., UCELLA G.
On the design and implementation of PIXAL, a language for image processing
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for images processing", Academic Press, 1981, 89-98

LEV 81 : LEVIALDI S., MAGGIOLO-SCHETTINI A., NAPOLI M., UCCELLA G.
PIXAL : a high-level language for image processing
In : M. ONOE, K. PRESTON, A. ROSENFELD, "Real-time/parallel computing; image analysis", 1981, Plenum Press, 10-14

LEV 82 : LEVINE M.D., NAZIF A.
An experimental rule-based system for testing low-level segmentation strategies
In : K. PRESTON, L. UHR, "Multicomputers nad image processing, Algorithms and programms", 1982, Academic Press, 149-160

LIV 78 : LIVERCY C.
Sémantique d'un langage de programmation
Théorie des programmes, Dunod, Paris, 1978, 241-314

LOU 80 : LOUGHEED R.M., McCUBBREY D.L.
The cytocomputer: a practical pipelined image processor
CH1494-4/00/0000-0271 $00.75 c 1980 I.E.E.E., 271-277

LUX 84 : LUX A.
Sur les problèmes d'intelligence artificielle en vision par ordinateur
Reconnaissance des formes et intelligence artificielle
4ème Congrès, AFCET, Roquencourt, 23-27 Jan 84

MAG 81 : MAGGIOLO-SCHETTINI A.
Comparing some high-level languages for image processing
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for image processing", 1981, Academic Press, 159-164

MAR 82 : MARR P.
Vision. A computational investigation into the human representation and procesing of visual information
FREEMAN, San Francisco CA, 1982

MAR 72 : MARTELLI A.
Edge detection using heuristic search methods
Computer Graphics and Image Processing, 1972, 1, 169-182

MAR 76 : MARTELLI A.
An application of heuristic search methods to edge and contour detection
Comm. of the A.C.M., 1976, 19, 73-83

MEG 82 : MEGED A., BELAID A., DRAMAN C., KEITH B., WENDEL P.L.
Projet ICOTECH
Système pour la conception assistée par ordinateur et pour le traitement numérique d'images
LRD/ENSPS 7 rue de l'Université 67000 Strasbourg, 1982

MOR 84 : MORIN E.
L'extraordinaire problème: le cerveau et l'esprit
Interfaces, 1984, 25, 3-9

NAG 81 : NAGIN P.A., HANSON A.R., RISEMAN M.
Region relaxation in parallel hierarchical architecture
In : M. ONOE, K. PRESTON, A. ROSENFELD, "Real-time/parallel computing; image analysis", 1981, Plenum Press, 37-61

NIE 81 : NIEMAN H.
Pattern analysis
Springer Verlag 1981

NIL 71 : NILSSON N.J.
Problems solving methods in artificial intelligence
Mc Graw Hill, New York, 1971

NOR 81 : NORTON-WAYNE L.
High-level image languages for automatic inspection - the filestructure problem
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for image processing", 1981, Academic Press, 139-146

PAB 74 : PABION J.F.
Logique mathématique
Hermannn, Paris, 1974

PAI 78 : PAIR C.
La programmation : de l'énoncé au programme
Rapport 78-P-061 CRIN

PAI : PAIR C.
Informatique non numérique : structures de données et algorithmes fondamentaux
Institut National Polytechnique de Nancy

PER 81 : PERSOON E.
Processing capabilities needed in learning systems for picture recognition
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for image processing", 1981, Academic Press, 301-333

PIN 81 : PINSON S.
Representation des connaissances dans les systèmes experts
RAIRO informatique, 1981, 15 , 4, 343-367

PON 83 : PONCIN J.
Transmission codage et compression de l'information
Traitement et analyse d'images numériques
E.S.E. Session de perfectionnement L801, 83

POT 82 : POTTER J.L.
MPP architecture and programming
In : K. PRESTON, L. UHR, "Multicomputers and image processing, Algorithms and programms", 1982, Academic Press, 275-289

PRA 78 : PRATT W.K.
Digital image processing
John Wiley & Sons, New York 1978

PRE 81 : PRESTON K., ONOE M.
Digital processing of biomedical images
Plenum Press, New York 1981

RAD 81 : RADHAKRISHNAN T., BARRERA R., GUZMAN A., JINICH A.
Design of a high-level language (L) for image processing
In : M.J.B. DUFF, S. LEVIALDI, "Languages and Architectures for Image Processing", 1981, Academic Press, 25-40

Rev 1 : Imageries
Rev. Prat., 1984, 34, 173-240

Rev 2 : Images de synthèse
Sciences et Technique, Mai 1984, n° spécial hors série

Rev 3 : La révolution des images
La Recherche, Mai 1983, n° spécial 144

ROS 76 : ROSENFELD A. KAK A.C.
Digital picture processing
Academic Press, New York 1976

ROS 76 : ROSENFELD A.
Digital picture analysis
Springer Verlag, Topics in applied physics, 11, Berlin 1976

RUC 79 : RUCH T.C.
The eye as an optical instrument
Vision and the retina
Visual fields and central visual pathways
In : T.C. RUCH, H.D. PATTON, "Physiology and biophysics, the brain and neural function", 1979, W.B. Saunders compagny

SCH 78 : SCHACHTER B.J. ROSENFELD A.
Some new methods of detecting step edges in digital pictures
Comm of the A.C.M., 1978, 21, 172-176

SER 82 : SERRA J.
Image analysis and mathematical morphology
Academic Press, 1982

SHN 77 : SHNEIER M.O.
Recognition using semantic constraints
IJCAI-77, 1977, 2, 585-589

STE 82 : STERNBERG S.R.
Pipeline architectures for image processing
In : PRESTON K., UHR L., "Multicomputers and image processing, Algorithms and Programs", 1982, Academic Press, 291-305

TIM 78 : TIMSIT C.
The PROPAL II computer
International conference on parallel processing, 1978

UHR 81 : UHR L.
A language for parallel processing of arrays, embedded in PASCAL
In : M.J.B. DUFF, S. LEVIALDI , "Languages and Architectures for Image Processing", 1981, Academic Press

UHR 82 : UHR L., SCHMITT L., HANRAHAN P.
Cone/pyramid perception programs for arrays and networks
In : PRESTON K., UHR L., "Multicomputers and image processing, Algorithms and Programs", 1982, Academic Press, 179-191

VAN 69 : VAN WIJNGAARDEN A.
Report on the algorithmic language ALGOL 68
Numerische Mathematik, 14, 1969

VRO 81 : VROLIJK J. PEARSON P.L. PLOEM J.S.
TAL: an interpretive language for the leyden television analysis system
In : M.J.B. DUFF, S. LEVIALDI , "Languages and Architectures for Image Processing", 1981, Academic Press, 125-137

WIN 75 : WINSTON P.H.
The psychology of computer vision
Mc Graw Hill, New York 1975

WOO 81 : WOOD A.
The interaction between hardware, software and algorithms
In : M.J.B. DUFF, S. LEVIALDI , "Languages and Architectures for Image Processing", 1981, Academic Press, 1-11

WUL 76 : WULF W.A., LONDON R.L., SHAW M.
An introduction to the construction and verification of ALPHARD programs
I.E.E.E. trans on software engineering, 1976, SE-2, 253-264

YAC 79 : YACHIDA M., IKEDA M., TSUJII S.
Plan-guided analysis of noisy dynamic images
IJCAI-79, 1979, 978-983

EXEMPLES DE TRAITEMENTS

EXEMPLES DE TRAITEMENTS

Cette annexe comportera des exemples illustrés de traitements éffectués par le système MEDECIM.

Il ne nous est pas possible pour l'heure de présenter ces images; en effet la bibliothèque des logiciels de service fournis par les laboratoires ADAC a récemment été modifiée et il nous faut obtenir la documentation nécessaire à la modification des primitives T.I. les référençant.
En outre afin de permettre un chainage des programmes, plutot qu'une d'utiliser une méthode d'overlay fort lourde, il nous faut également modifier certaines de ces routines afin de gérer une zone mémoire servant de boite aux lettres.