

50376
1980
236

N° d'ordre : 873

50376
1980
236

THÈSE

présentée à

L'UNIVERSITÉ DES SCIENCES ET TECHNIQUES DE LILLE

pour l'obtention du titre de

DOCTEUR TROISIEME CYCLE

par

Khadir OURIACHI

**PROCESSUS DE RECONNAISSANCE
DES FORMES APPLICABLE A
UN ASSEMBLAGE AUTOMATIQUE**



Soutenu le 16 décembre 1980 devant la Commission d'Examen :

MM.	P. VIDAL	Président
	M. BOURTON	Rapporteur
	M. VERON	Examineur
	L. MICLET	Examineur
	M. BERTHOD	Examineur

036 244621

AVANT - PROPOS

Le travail présenté dans ce mémoire a été effectué au Centre d'Automatique de l'Université des Sciences et Techniques de Lille 1, dans le cadre d'un projet d'automatisation d'un poste d'assemblage de pièces mécaniques.

Notre plus grande reconnaissance va à Monsieur le Professeur Pierre VIDAL qui, par ses qualités humaines et son grand dynamisme, nous a apporté efficacement son soutien dans les périodes difficiles et su créer au laboratoire une atmosphère très amicale.

Nos vifs remerciements s'expriment ici à l'égard de Monsieur le Professeur Michel BOURTON pour le rôle initiateur qu'il a joué dans nos recherches en nous en fournissant le support et pour l'intérêt constant qu'il a bien voulu nous témoigner tout au long de ce travail. Qu'il trouve ici l'expression de notre attachement.

Monsieur le Professeur M. VERON nous honore grandement en acceptant de juger ce travail et de participer à notre jury de thèse.

Nous exprimons nos remerciements à Monsieur le Professeur L. MICLET pour l'intérêt qu'il nous manifeste en participant à notre jury. Qu'il trouve ici l'assurance de notre reconnaissance.

Que Monsieur M. BERTHOD, ingénieur INRIA, trouve ici l'expression de notre reconnaissance pour l'honneur qu'il nous fait de bien vouloir porter un jugement sur le contenu de nos travaux en faisant parti de notre jury.

Nous tenons à rendre hommage à Madame A. PIGNON, Madame M. LELONG, Monsieur G. VEINNANT et Monsieur HOUZE qui ont, à des titres divers, largement contribué à la réalisation matérielle de ce mémoire.

INTRODUCTION GENERALE

Les travaux présentés dans ce mémoire sont effectués dans le cadre d'un projet, en cours de réalisation, qui automatise un poste d'assemblage de pièces mécaniques. Ils traitent plus particulièrement de procédures de traitement d'images, de reconnaissance des formes, et d'assemblage.

Ce mémoire se décompose en trois parties principales :

Dans une première partie englobant la totalité du premier chapitre, est exposée une procédure de traitement d'images en temps réel.

Cette procédure se déroule en quatre étapes :

- l'image acquise est digitalisée, puis filtrée afin d'éliminer le bruit spatial pour assurer la modélisation des pièces dans des conditions satisfaisantes.
- le contour des pièces est détecté selon une méthode originale permettant d'améliorer les performances de la méthode gradient; les lignes du contour sont affinées et lissées, ensuite segmentées suivant une méthode fondée sur deux critères de segmentation : le critère d'Attneave et le critère de Freeman.
- le contour est, après une approximation polygonale, interprété par une phrase d'un langage spécifique L.A.P (Language describing the Assembly of Pixels). Cette phrase est un modèle de représentation de la pièce analysée.
- enfin, les pièces sont localisées dans la scène selon une méthode qui calcule leurs centres de gravité en contrastant au maximum l'image et en associant des poids aux pixels du contour . L'orientation angulaire peut être déterminée soit par une méthode de calcul de l'axe minimum d'inertie et du centre de l'aire, soit par une méthode qui calcule le minimum de la composante paire d'une signature polaire.

La deuxième partie à laquelle est consacré le troisième chapitre traite d'une procédure de reconnaissance des formes des pièces. A cette fin, on a été amené d'envisager une approche inspirée de la théorie des langages formels : l'approche structurale ou syntaxique, qui est particulièrement bien adaptée dès lors qu'il s'agit d'analyser des formes riches en informations structurales.

Dans cette procédure, on distingue deux phases essentielles :

. une phase d'apprentissage qui consiste à inférer, pour chacune des formes considérées, un modèle structural sous-forme de grammaire régulière (ou d'automate fini déterministe correspondant). La méthode d'inférence grammaticale employée est fondée sur le "Lemme de l'étoile", elle utilise comme heuristique la recherche, dans un échantillon de phrases, de récursions caractéristiques de la grammaire qui l'engendre, (ou des boucles de l'automate qui accepte les phrases de l'échantillon).

. une phase de reconnaissance qui, au terme d'une analyse syntaxique d'une phrase, représentant une forme inconnue, par rapport à un automate fini déterministe, permet de décider si l'automate accepte la phrase.

La troisième partie, constituée du quatrième chapitre, décrit une procédure d'assemblage automatique. La tâche d'assemblage se déroule comme suit :

soit un couple de pièces, ces pièces étant reconnues, un manipulateur saisit d'abord l'une d'entre elles et la place en un endroit précis sur le plan d'assemblage, ensuite saisit l'autre pièce, la fait venir à son tour sur le plan d'assemblage et la positionne de manière telle qu'elle puisse s'emboîter partiellement ou totalement dans la première.

L'algorithme procède en deux étapes :

- dans la première étape, il s'agit de chercher, parmi toutes les couples de sous-formes emboîtables détectées, en analysant deux pièces reconnues, celle qui permet d'assembler ces deux pièces avec le maximum

de vraisemblance. Le critère d'assemblage choisi est le degré d'emboîtement des sous-formes emboîtables.

- la deuxième étape permet d'une part la détermination de certains paramètres que nous appellerons paramètres d'assemblage; il s'agit d'une direction, d'un centre et d'un vecteur, choisis en émettant des critères sévères sur les informations fournies par ses paramètres : en effet, ces informations doivent être

- . non redondantes,

- . suffisantes pour réaliser l'assemblage,

et d'autre part de décrire l'exécution de la tâche d'assemblage par un manipulateur à quatre degrés de liberté.

Enfin, en conclusion, on énonce quelques perspectives d'amélioration de ces procédures en vue d'une extension de l'assemblage à des pièces tridimensionnelles.

CHAPITRE 1

LA RECONNAISSANCE DES FORMES EN ROBOTIQUE INDUSTRIELLE

CHAPITRE 1

I - RECONNAISSANCE DES FORMES ET EVOLUTION EN ROBOTIQUE INDUSTRIELLE

La reconnaissance des formes constitue l'étape la plus ambitieuse de l'automatisation des systèmes complexes. C'est bien là que la gageure de remplacer l'homme par le robot prend toute son acuité. Il s'agit en effet d'imiter l'homme dans un domaine - la perception visuelle, tactile, auditive, où il excelle et où son action fait appel à des niveaux d'intégrations très élevés.

Pourtant la reconnaissance des formes n'est pas encore une doctrine en soi. Elle est une science disparate qui participe à la fois de l'intelligence artificielle, de la théorie statistique de décision, de la théorie de l'information, et de la théorie des langages. Au demeurant les mathématiques n'y rentrent jusqu'à présent, presque uniquement, que comme outil et sur un plan algorithmique, non comme un procédé de compréhension et d'unification.

Le concept de forme n'est pas simple à préciser. Ainsi, est définie comme forme toute structuration minimale perceptible qui émerge d'un bruit aléatoire, tout groupe d'éléments qui serait perçu, d'emblée avant que chaque détail soit isolé, comme n'étant pas le produit d'un assemblage au hasard. La forme c'est aussi la manière dont les parties d'un tout sont arrangées entre elles, dont une chose est construite, disposée, agencée.

Le terme de forme revêt donc une double signification, l'une descriptive, l'autre normative. Il y a donc à la fois une idée de conception et une idée de modèle.

Loin de se limiter aux automatismes industriels, le domaine de la reconnaissance des formes s'étend jusqu'à l'étude des structures mentales, logiques, d'une façon générale jusqu'à la recherche de toute trace d'organisation quelle qu'elle soit dans un système quelconque. Mais pour l'heure les recherches se préoccupent davantage du premier problème, plus simple, dont les motivations ont une urgence économique immédiate. La robotique industrielle en particulier occupe dans le domaine de la reconnaissance

L'approche statistique est la plus ancienne et la mieux établie. Elle comporte traditionnellement trois étapes (figure 1) souvent imbriquées : la paramétrisation, l'apprentissage et la classification (décision, reconnaissance). La paramétrisation consiste à extraire un faible nombre de paramètres réputés pertinents de la forme au moyen de capteurs convenables. C'est sans nul doute l'étape qui fait le plus appel à la connaissance du phénomène physique étudié.

L'apprentissage statistique avec professeur est essentiellement une réduction cohérente d'information face à la quantité prohibitive de données que constituait la mémorisation des classes par leur occurrence. En effet, pour représenter une classe de formes dans l'espace des paramètres, on cherche à extraire sa spécificité par rapport aux autres classes, sous-forme de relations entre les paramètres des formes apprises dans cette classe.

La classification a pour but de trouver une décision d parmi un ensemble de décisions déterminant la classe d'un individu représenté dans l'espace paramétrique (assimilé à R^n), par un vecteur multidimensionnel v appelé vecteur-paramètres. Pour ce faire, on construit une fonction f , appelé fonction de décision, telle que

$$d = f (v)$$

La fonction de décision peut être une distance euclidienne (dans R^n) entre le vecteur-paramètres v et certains vecteurs prototypes (un vecteur par classe) qui mesure la ressemblance (ou non ressemblance) indiquant le degré d'appartenance à chaque classe. La classe choisie est la classe dont le vecteur-prototype est le moins distant de v .

Dans l'hypothèse où les représentants d'une même classe se concentrent dans une même région de l'espace des paramètres (R^n), la construction de la fonction de décision revient à partitionner l'espace en régions : dans chacune des régions, $f (v)$ est constante et indique la classe à laquelle v appartient.

des formes une place remarquable. Elle y voit un moyen de développer et mettre en oeuvre des manipulateurs à commandes programmables qui se distinguent des automatismes classiques car ils peuvent accomplir des tâches variées et complexes avec un comportement qualifié d'"intelligent". De son côté la reconnaissance des formes voit en la robotique un domaine d'étude particulièrement riche et motivant pour élaborer et expérimenter ses modèles, car elle s'intéresse à la construction de machines capables d'interagir avec leur environnement physique d'une façon relativement autonome, c'est-à-dire autrement qu'un exécutant des séquences d'actions explicitement programmées.

En effet, jusqu'à présent, les efforts en robotique industrielle ont surtout porté sur les aspects mécaniques des robots et leur motorisation plutôt que sur leur commande et l'utilisation de capteurs évolués. A de rares exceptions près, les robots actuellement opérationnels sont équipés de capteurs de modèles très classiques : contacts de fin de course, cellules photoélectriques etc ... Ils travaillent dans un environnement où les objets à manipuler sont préalablement positionnés et orientés à l'aide de dispositifs de guidage spécifiques. Le coût de ces guides peut être évalué à dix fois celui du robot /1/. D'où la nécessité de munir le robot de capteurs évolués : caméra, lasers, capteurs de forces ...

Cette évolution s'accompagne d'une augmentation considérable de la part de la reconnaissance des formes. Celle-ci représentera dès la première moitié des années 80, 50% des coûts de la robotique industrielle contre 20% seulement en 1977 /2/.

II - METHODES DE RECONNAISSANCE DES FORMES

Il est généralement admis de classer les méthodes de reconnaissance des formes en deux approches générales : l'une statistique fondée sur la théorie statistique de décision, l'autre structurale, fondée sur la théorie des langages et des automates /3/.

La méthode statistique de reconnaissance est à la base de la mise en oeuvre de quasiment toutes les techniques de reconnaissance automatique actuellement opérationnelles : classification des pièces mécaniques à la sortie d'une fonderie à Lorient (France) /4/, la reconnaissance de 200 pièces ayant chacune cinq positions stables à la Régie Renault /5/ ...

Cependant les limitations de la reconnaissance statistique sont rapidement apparues et ceci en particulier à l'occasion du développement considérable du traitement d'images. En effet, l'approche statistique s'est heurtée au caractère intrinsèque de la forme visuelle qui apparaît dans l'image comme une structure déterministe reliant des constituants dépourvus par eux-mêmes de spécificité vis à vis du problème. Les traits pertinents de la forme sont inscrits dans l'agencement caractéristique d'éléments appelées primitives.

Il est donc nécessaire pour décrire cette forme par un formalisme adapté échappant à la statistique de se tourner vers des outils mathématiques fournis par la théorie des langages. L'objet de ce formalisme est précisément de décrire les structures des formes par des phrases résultant de la concaténation de mots identifiant des primitives. Cette approche dite structurelle ou grammaticale de reconnaissance comporte trois phases principales : l'identification, l'inférence structurelle et la reconnaissance (figure 2).

L'identification consiste à décomposer la forme en signes élémentaires ou primitives convenablement choisis. Ces primitives sont identifiées par les mots d'un vocabulaire. Une phrase (résultant de la concaténation des mots des primitives) interprète la forme sous forme de juxtaposition de primitives.

L'inférence structurelle est essentiellement un problème d'apprentissage. Il s'agit de trouver une représentation condensée, sous forme de grammaire, à un langage dont on connaît le tout ou une partie. En pratique on se limitera à la connaissance d'une partie finie de ce langage.

Quand les distributions probabilistes des membres de chaque classe dans l'espace R^n sont connues (ou lorsqu'on peut les estimer), $P (v/k)$ représente une meilleure mesure de ressemblance que la distance de v à un vecteur-prototype de la classe k . $P (v/k)$ étant une probabilité conditionnelle de v étant donnée la classe k , la fonction de décision devient alors :

$$d = f (v) = \text{Max}_{k \in C} P (z/k)$$

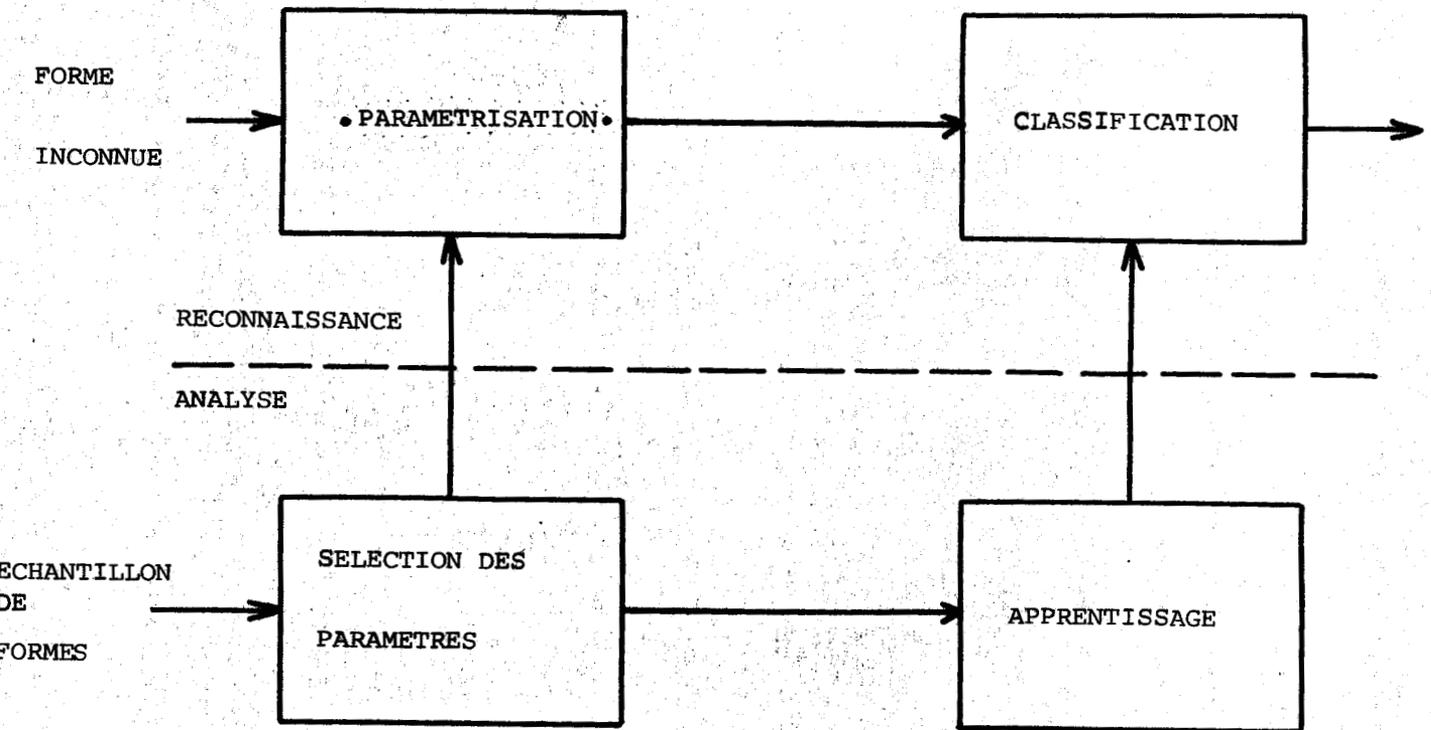


Figure 1 : Système de reconnaissance statistique

Le choix fondamental est le type de grammaire (context-sensitive, context-free ou réglère) que l'on veut inférer. La grammaire inférée à partir d'un langage de phrases qui interprètent une classe de formes constitue un modèle génératif de ce langage.

La reconnaissance est un problème d'analyse structurelle. Elle se fait en vérifiant l'appartenance d'une phrase inconnue au langage engendré par l'une des grammaires inférées au cours de la phrase d'apprentissage; c'est donc une inversion du modèle génératif.

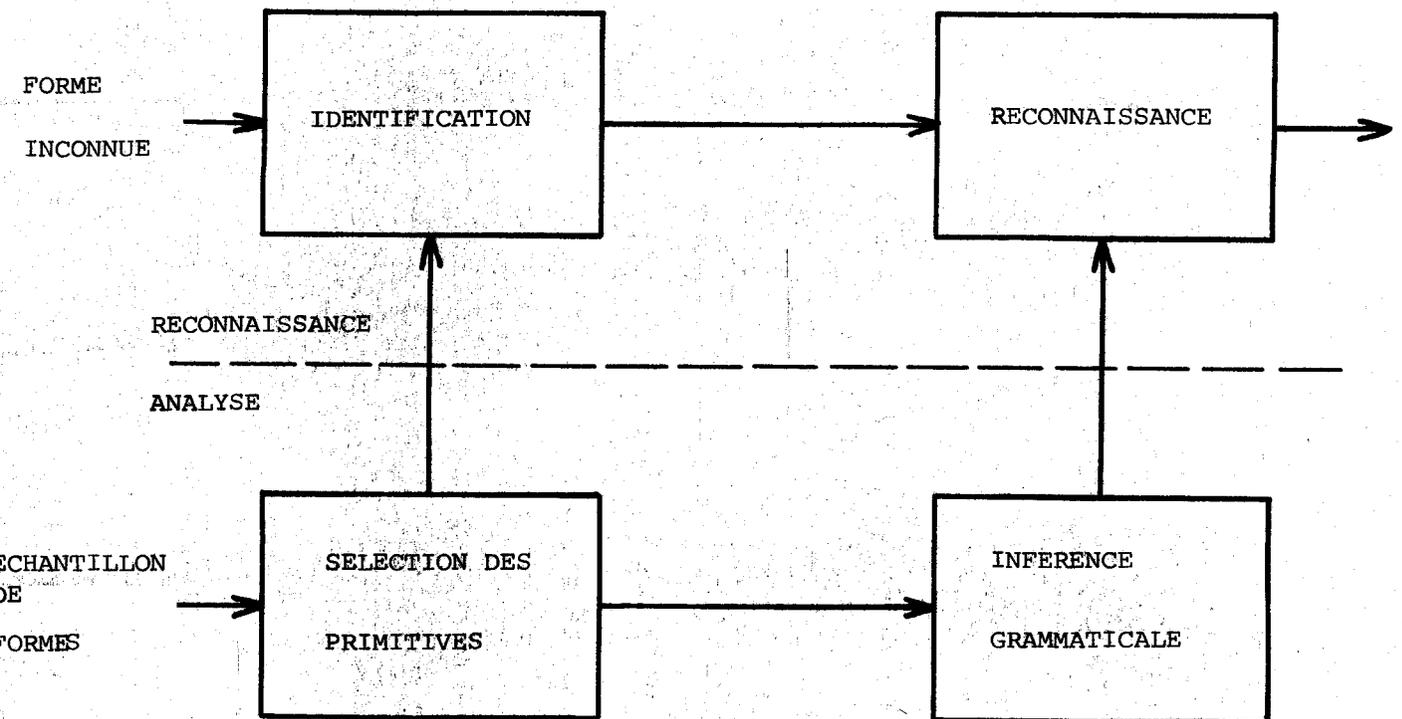


Figure 2 : Système de reconnaissance structurelle

Les travaux actuels tant fondamentaux qu'appliqués, effectués dans le domaine de la reconnaissance tendent sensiblement à privilégier le développement des techniques de reconnaissance structurale : classification automatique de chromosomes humains /6/, classification des fossiles appartenant à la famille Globotruncana, /7/ identification des empreintes digitales /8/

Mais cet intérêt croissant, apporté à la reconnaissance structurale, apparaît évident surtout à l'occasion de l'extension du domaine d'application du robot industriel à des tâches aussi complexes que le montage et l'assemblage des pièces mécaniques en petite et moyenne série, qui fait actuellement appel à une main-d'oeuvre importante.

Précisément, ce mémoire entreprend de mettre en oeuvre une procédure pour automatiser une tâche d'assemblage de pièces mécaniques en exploitant la commodité et la souplesse qu'offre l'approche structurale de reconnaissance des formes.

Cette procédure comporte deux étapes principales : la reconnaissance des pièces à manipuler, et l'assemblage de ces pièces.

La reconnaissance est basée sur l'interprétation de la forme de la pièce inconnue au moyen d'une phrase qui résulte de la concaténation de mots d'un langage spécifique L.A.P. (Langage décrivant l'Assemblage des Pixels), et la modélisation de cette forme grâce à un algorithme d'inférence de grammaires régulières.

L'assemblage est basé d'une part sur la recherche de couples de sous-formes pouvant s'emboîter exactement et la mesure d'un degré d'emboîtement et d'autre part sur la définition et le calcul de paramètres d'assemblage.

Cette technique d'assemblage présente par rapport aux techniques utilisant les forces de contact (Force - Feedback) développées tout récemment /9/, /10/ l'avantage d'une grande souplesse de fonctionnement.

Cependant elle nécessite de munir le robot d'une caméra et une importante étape de traitement des images qui risque cependant de ralentir la tâche d'assemblage.

L'effort doit donc porter autant sur le choix de capteur que sur le développement et la mise en pratique d'algorithmes de traitement et de reconnaissance rapides et efficaces.

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

- /1/ M. DIETRICH
" Application de petits et moyens manipulateurs "
Journées ' Automatisations par manipulateurs ', Adepta - Janvier 75
- /2/ YONEMOTO, K. SHIINO
" Present state and future outlook for industrial robots
in japan "
The Industrial Robot, December 77
- /3/ K.S. FU, A. ROSENFELD
" Pattern recognition and Image processing "
I.E.E.E. Trans. Computers, vol C.25, n°12, pp 1336-1346
- /4/ C. BIGUEREAU, B. ESPIAU, JJ BORRELY, C. LAURGEAU
" Recognition and sorting of mechanical pieces "
8th International Symposium on Industrial Robots
30.5.78 - 1.6.78 Stuttgart, West Germany
- /5/ M. DE PRACONTAL
" Robotique Industrielle "
Electronique application, n°7, Octobre 78
- /6/ K.S. FU
" Syntactic methods in pattern recognition "
Academic Press, New York
- /7/ Kh. OURIACHI, M. BOURTON
" Méthode de classification automatique des fossiles globotruncana "
Rapport D.E.A., Centre d'Automatique - Université de Lille 1
Juillet 78.

- /8/ B.MOAYER, K.S. FU
" A syntactic approach to fingerprint pattern recognition "
Pattern Recognition, vol.7 n°12/ - June 75
- /9/ S.H. DRAKE
" Using compliance in lieu of sensory Feedback for
automatic assembly "
Ph.D Thesis, MIT Departement of Mechanical Engineering,
September 77.
- /10/ D.E. WHITNEY, J.L NEVINS
" What is the remote center compliance (R.C.C.) and
what can it do ? "
Proceedings of 9 th International Symposium on Industrial
Robots, Washington D.C., March 13-15,79.

C H A P I T R E I I

ACQUISITION ET TRAITEMENT DE L'INFORMATION

CHAPITRE II

ACQUISITION ET TRAITEMENT DE L'INFORMATION

Les capteurs type caméra offrent au robot un moyen de percevoir son environnement avec l'avantage d'une grande souplesse de fonctionnement obtenu grâce à la suppression des dispositifs de guidage. Ils travaillent à distance et permettent d'obtenir une vue globale des situations. En contrepartie leurs utilisations présentent des difficultés, car ils fournissent une grande quantité de données mal structurées compte tenu de ce que l'on cherche à extraire. L'acquisition sous forme d'images; des données transmises par la caméra nécessitent donc une étape de pré-traitement au cours de laquelle seront développés des opérateurs de traitement d'images /1/ et des méthodes d'élaboration de modèles d'informations pour interpréter ces données.

Les travaux fondamentaux et appliqués effectués sur le thème du traitement d'images se distinguent davantage par le type de scènes visuelles à traiter que par des approches méthodologiques différentes /2/ /3/. Si le problème d'analyse des scènes composées d'éléments bidimensionnels a reçu des solutions satisfaisantes /4/, par contre pour les scènes à composants tridimensionnels les solutions se limitent presque exclusivement au cas simple des scènes composées de blocs polyédriques /5/.

Précisément, pour développer la vision artificielle chez un robot exécutant une tâche d'assemblage de pièces mécaniques, nous nous sommes limités à des scènes dont les composants sont (ou peuvent être assimilés à) des formes bidimensionnelles.

Nous distinguons dans l'étape de traitement d'images trois phases conceptuelles: l'acquisition, la segmentation et l'interprétation.

I - ACQUISITION DE L'IMAGE DE LA SCENE

La phase d'acquisition transforme la scène visuelle en une image numérisée qui se présente sous la forme d'un tableau bidimensionnel dont chaque élément (pixel) porte l'information de luminosité d'une surface élémentaire de la scène originale. A la discrétisation des deux dimensions d'espace, s'ajoute une quantification de la luminosité dont les valeurs sont prises dans un ensemble fini de niveaux.

L'image digitale fournie par la caméra est souvent entachée d'un bruit spatial important. Il est nécessaire d'éliminer ce bruit afin d'assurer la modélisation des objets présents dans la scène, dans des conditions satisfaisantes.

Un filtrage simple peut être obtenu par convolution de l'image bruitée par le biais de la réponse percussive d'un filtre spatial $[2N + 1, 2N + 1]$

$$f^*(x,y) = \frac{1}{K} \sum_{u=-N}^{u=N} \sum_{v=-N}^{v=N} f(x+u,y+v) \cdot F(u,v)$$

où $f^*(x,y)$ est le point d'image calculée de coordonnées x,y

$f(x,y)$ le point d'image originale de coordonnées x,y

$F(u,v)$ un coefficient de la matrice (ou masque) correspondant au filtre

et K un facteur de normalisation

Pour $N = 1$, on a le cas simple d'un filtre $[3 \times 3]$. La matrice des coefficients peut être prise égale à la matrice $[3 \times 3]$:

$$\begin{bmatrix} \sqrt{2} & 1 & \sqrt{2} \\ 1 & 1/2 & 1 \\ \sqrt{2} & 1 & \sqrt{2} \end{bmatrix}$$

et le facteur de normalisation : $K = 10$

II - PHASE DE SEGMENTATION DE LA SCENE

La scène à analyser est une scène multi-objets. Donc l'une des premières difficultés rencontrée au cours de l'analyse de cette scène est de pouvoir distinguer les formes des objets du contexte.

La phase de segmentation consiste précisément à partitionner l'image en un ensemble de régions formées chacune par un ensemble de pixels de niveau de gris à peu près uniforme et en un ensemble d'entités géométriques élémentaires souvent des lignes déterminées par des variations plus ou moins nettes du niveau de gris.

On suppose que les objets présents dans la scène reposent sans recouvrement sur une surface plane.

II.1 - Recherche de contour des formes

La détection de contour d'une région de l'image ou d'une forme consiste à transformer l'image multi-niveaux en une image binaire, formée de "0" pour les points extérieurs à la forme; de "1" pour les points intérieurs. Le contour est défini par l'ensemble de points frontières entre forme et contexte.

La littérature scientifique est très riche en techniques de recherche de contour; toutefois, il est possible de dégager quelques principes généraux :

- *Méthode du gradient /6/*
=====

L'opération gradient est une approximation de différenciation spatiale de l'échelle de gris. La méthode du gradient donne une image de contour ou image gradient. Les régions de l'image dont le niveau de gris est à peu près uniforme sont caractérisées par un faible gradient alors que les zones frontalières de formes sont caractérisées par un gradient maximum.

- *Méthode de Hueckel /7/*
=====

Cette méthode procède de la façon suivante : on déplace sur l'image une fenêtre circulaire de 40 à 60 points; en chaque position on pose les questions suivantes :

- *quelle est la droite qui ~~est~~ sépare le mieux la fenêtre en deux plages uniformes ?*
- *la distance entre la fenêtre réelle et la fenêtre digitale autorise t-elle à remplacer la première par la seconde ?*

La méthode permet donc la détermination du contour sous forme de segments de droites. C'est une différence fondamentale avec la méthode du gradient qui, à partir d'une image (matrice de points) donne une image de contour .

- *La méthode à base statistique /8/*
=====

Les méthodes statistiques de détection de contour se basent sur l'analyse de la répartition des fréquences d'apparition des intensités lumineuses, obtenue par un histogramme. La forme dont on recherche le contour se caractérisent par des niveaux de gris différents de ceux du contexte, l'histogramme fait alors apparaître deux modes ou " pics "

caractérisant la forme et le contexte /9/. La vallée située entre les pics permet donc de séparer la forme de son contexte. Ainsi le contour est obtenu en seuillant l'image par la valeur σ du "fond" de la vallée.

II.1.1. - Utilisation de la méthode du gradient

La détermination des points frontières par l'exploitation de la technique du gradient présente un grand avantage en raison de la simplicité des algorithmes mis en oeuvre et par l'effet de l'opération gradient qui accentue les hautes fréquences spatiales riches en informations sur les contours.

L'image gradient peut être calculée à partir de l'image initiale en appliquant l'opérateur de Roberts sur chaque fenêtre (2x2).

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

le gradient est obtenue grâce à l'équation

$$G(a) = | a-d | + | c-b |$$

Le principal défaut de la méthode du gradient est, outre sa sensibilité au bruit, la multiplication, dans les zones à transitions dégradées, des points voisins du contour. Pour atténuer les effets de ces inconvénients, nous proposons de modifier cette méthode en appliquant l'opérateur de Roberts à une fenêtre plus large (4x4).

II.1.2. - Méthode du gradient améliorée

L'image gradient est ainsi obtenue en calculant pour chaque fenêtre (4x4) de l'image ,

$$\begin{vmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{vmatrix}$$

le gradient modifié donné par l'expression

$$G(a) = \sqrt[4]{A.B.C.D...}$$

$$\text{où } A = |f-k| + |j-g|$$

$$B = |a-p| + |m-d|$$

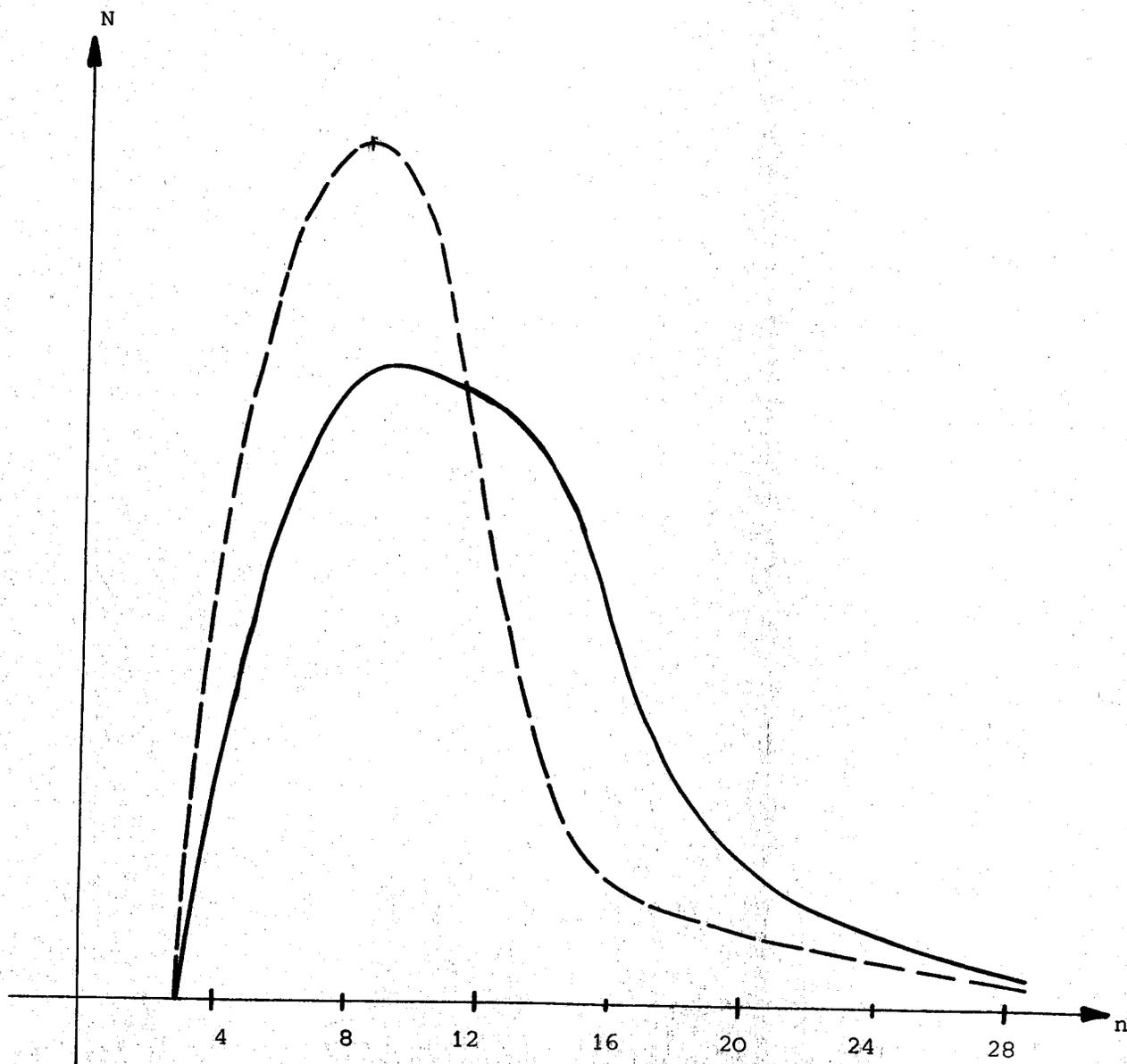
$$C = |b-o| + |i-h|$$

$$\text{et } D = |c-n| + |e-l|$$

Chaque point de la fenêtre est utilisé une et une seule fois. Le gradient modifié consiste en un produit de quatre gradients conventionnels dans trois directions différentes (A et B sont deux gradients de même direction car on suppose que les pixels sont uniformément espacés).

Ainsi l'image gradient n'est pas sensible à la rotation de l'image. De plus les points le long du contour sont nettement mis en évidence, car le produit de gradients de grandes valeurs est très grand. Au contraire l'ensemble des points isolés dûs au bruit et à la différenciation est considérablement réduit car l'un au moins des quatre gradients a,b,c et d a une valeur faible, le produit est donc également faible.

La figure 1 montre les histogrammes du gradient de Robert (1.a) et du gradient modifié (1.b) d'une même image. On remarque que l'opérateur gradient modifié redistribue les points de façon qu'un plus grand nombre de points le long des contours apparaissent au pic de l'histogramme.



— Courbe de distribution des pixels avec la méthode gradient classique

- - - Courbe de redistribution avec la méthode gradient améliorée

n : niveau de gris

N : nombre de pixels



II.2 - Lissage et affinage des lignes de contour

Les contours qui résultent de l'application de la méthode du gradient améliorée sont bruités et ont par endroit une certaine épaisseur. Cette épaisseur est due le plus souvent à la multiplication des points voisins de contours dans les zones à transitions dégradées. D'où une importante phase de filtrage et de lissage des contours qui ralentit l'algorithme précédent.

II.2.1. - Filtrage analytique

Un premier filtrage analytique permet par seuillage de réduire le bruit dans l'image gradient améliorée. Après avoir calculé une moyenne locale pondérée T du gradient sur une fenêtre $| 2L + 1 \times 2L + 1 |$ centrée au point courant (x,y) , on procède à l'élimination de ce point si la valeur de son gradient est inférieure à la moyenne de la fenêtre.

$$g^*(x,y) = \begin{cases} 0 & g(x,y) < T \\ 1 & \text{sinon} \end{cases}$$

avec

$$T = \sum_{u=-L}^{u=L} \sum_{v=-L}^{v=L} g(x+u,y+v) p(u,v)$$

où $g^*(x,y)$ est le point de l'image gradient calculé de coordonnées x,y

$g(x,y)$ le point de l'image gradient originale de coordonnées x,y
et $p(u,v)$ le poids correspondant au point de la fenêtre de coordonnées u,v .

Ce filtrage transforme l'image gradient en image binaire.

II.2.3. - Filtrage logique

C'est un filtrage destiné à faire disparaître des points peu significatifs : points isolés, points ayant un seul voisin (figures 3a et 3b).

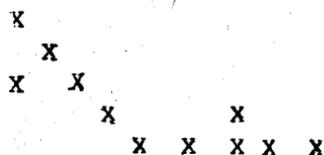


Figure 3a: portion de contour avant filtrage

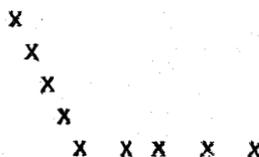


Figure 3b: portion de contour après filtrage

A l'issue de ces opérations successives de filtrage et de lissage, les contours sont transformés en courbes digitales définies par des séquences de points p_1, p_2, \dots, p_n où p_{i-1} et p_{i+1} sont les seuls voisins de p_i (l'indice i étant modulo n); ainsi si $p_i = (x_i, y_i)$

$$\text{on a : } \text{Max} [|x_i - x_{i+1}|, |y_i - y_{i+1}|] = 1 \text{ et } \text{Max} [|x_i - x_{i-1}|, |y_i - y_{i-1}|] = 1$$

11.3 - Segmentation des contours...

Le contour d'une forme est une courbe très complexe, qu'il est plus commode de décrire en termes de ses composantes. Ces composantes sont des entités géométriques élémentaires (segments de droites, arcs de cercles, portions d'ellipses).

Une variété de techniques a été proposée /10/ pour la segmentation d'une courbe complexe en parties significatives. L'une des plus ancienne, découlant des travaux d'ATTNEAVE /11/ considère que les points anguleux, c'est-à-dire les maximums locaux de courbure, sont des points intéressants pour segmenter une courbe. Une autre idée suggérée par

II.2.2. - Lissage des lignes de contour

Cette opération (figure 2) permet d'affiner les lignes de contour, en traitant les fenêtres binaires | 3x3 | centrées aux points dont la valeur est 1. Si ces fenêtres satisfont à l'une des conditions suivantes, la valeur du centre devient 0.

1 X O	X 1 1	X 1 X	X 1 X
1 1 O	O 1 X	1 1 X	X 1 1
X O O	O O O	X X O	O X X
O O O	O O X	X X O	O X X
X 1 O	O 1 1	1 1 X	X 1 1
1 1 X	O X 1	X 1 X	X 1 X

Exemples

```

X
 X
  X
   X X
    X X X

```

avant le lissage

```

X
 X
  X
   X
    X X X

```

après le lissage

```

X
X
X X
X X
X X
 X

```

avant le lissage

```

X
X
 X
 X
 X
 X

```

après le lissage

Figure 2 : opération de lissage des lignes de contour

FREEMAN /12/ consiste à prendre en compte également les points séparant les parties convexes des parties concaves de la courbe, c'est-à-dire les minimums locaux de courbure, pour améliorer la segmentation.

II.3.1. - Définition d'une fonction de courbure

Dans le plan réel euclidien la fonction de courbure d'un arc est définie comme la dérivée de la direction le long de cet arc par rapport à la longueur. Pour l'arc ayant pour équation :

$$y = f(x)$$

la fonction de courbure peut être exprimée en termes des dérivées première et seconde par l'expression /13/

$$\frac{d^2y / d^2x}{((1+(dy/dx)^2)^{3/2}} \quad (1)$$

Dans l'image, il s'agit d'une courbe digitale $| p_1, p_2 \dots p_n |$. Si on définit la courbure en un point $p_i = (x_i, y_i)$ en remplaçant les dérivées de l'expression (1) par les différences, la difficulté provient du fait que les pentes angulaires successives sur cette courbe digitale peuvent différer seulement d'un multiple de 45° , donc les petites variations de pente sont impossibles.

On peut pallier cette difficulté grâce à un lissage dans la mesure de la pente; ainsi la pente en p_i est définie par $(y_{i+k} - y_i) / (x_{i+k} - x_i)$ où $k > 1$ est appelé degré de lissage.

A partir de cette définition on peut décrire une procédure pour définir les maximums et les minimums significatifs de la fonction de courbure sur une courbe digitale grâce à une variation du degré de lissage. Cette procédure s'inspire d'une technique de détection des arêtes décrite par ROSENFELD /14/, /15/, qui consiste à rechercher les maximums significatifs du changement dans la moyenne du niveau de gris, en utilisant la variation du " degré de la moyenne ".

II.3.2. - Procédure de recherche des points significatifs

En chaque point $p_i = (x_i, y_i)$, on définit deux suites de vecteurs $\{ U_k, k = 1, 2, \dots \}$ et $\{ V_k, k = 1, 2, \dots \}$

$$U_k = (x_i - x_{i+k}, y_i - y_{i+k})$$

$$V_k = (x_i - x_{i-k}, y_i - y_{i-k})$$

Le cosinus des vecteurs U_k et V_k est donné par l'équation :

$$C_k(p_i) = U_k \cdot V_k / |U_k| |V_k|$$

On calcule $C_1(p_i), C_2(p_i), \dots, C_m(p_i)$ où m est un nombre choisi à priori tel que dans toute portion de m points de la courbe digitale, on est assuré de trouver au moins trois points significatifs.

Le point p_i tel que

$$C_m(p_i) < C_{m-1}(p_i) < \dots < C_1(p_i) < C_{1-1}(p_i)$$

est un maximum local de courbure si

$$|j - i| < 1/2 \text{ on ait } C_1(p_i) > C_q(p_j)$$

ou un minimum local de courbure si

$$|j - i| < 1/2 \text{ on ait } C_1(p_i) < C_q(p_j).$$

La figure 3 montre la segmentation d'une portion de courbe digitale grâce à la recherche des points significatifs. Les points marqués d'un cercle sont des points d'inflexion, ceux marqués par une flèche sont des sommets d'angles.

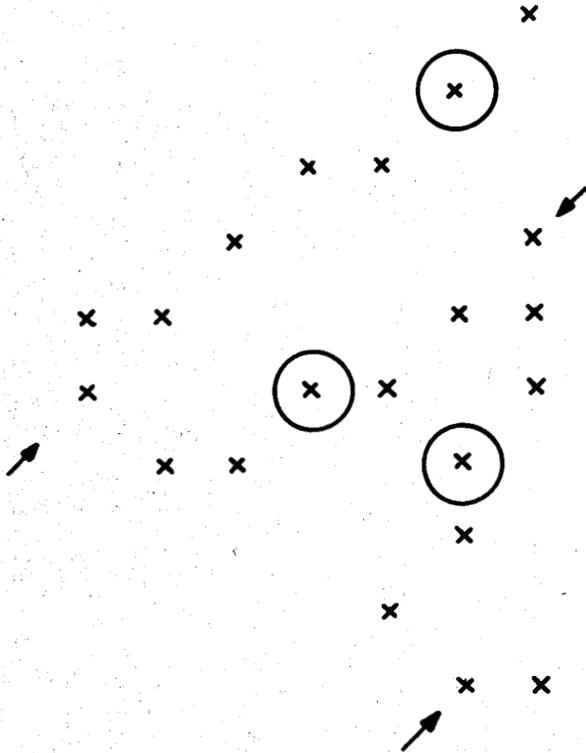


Figure 3 : Recherche des points significatifs

La segmentation d'un contour par cette procédure rend aisée sa description en termes de ces composantes qui sont des entités géométriques nettement moins complexes qu'on peut approcher par des entités plus élaborées : segments de droites, arcs de cercles, portions d'ellipse Cette segmentation et cette approximation facilitent beaucoup, nous allons le voir dans la prochaine phase, l'interprétation des contours de formes géométriquement complexes.

III - INTERPRETATION DE LA SCENE

Les contours de formes sont des entités géométriques complexes caractérisées par leurs propriétés intrinsèques, c'est-à-dire par la constance de certaines fonctions à travers diverses transformations. Leurs structures seront donc caractérisées des invariants structuraux qui peuvent être algébriques ou topologiques.

Dans l'approche statistique de reconnaissance des formes, la paramétrisation consiste à projeter une forme dans un espace de dimensionnalité réduite : l'espace de paramètres, ou " feature space ". Cependant, il arrive souvent qu'un nombre limité de paramètres ne suffit pas pour caractériser une forme et il est pratiquement impossible d'analyser tous les paramètres.

Dans l'approche structurale les traits pertinents d'une forme sont inscrits dans un agencement caractéristique d'éléments primitifs.

Le premier problème consiste d'abord à choisir convenablement les primitives ensuite et surtout d'être capable de les identifier. Pour ce fait, un langage spécifique décrivant la disposition des pixels dans l'image a été construit. Ce langage permet d'une part la mesure (identification) des primitives, d'autre part l'interprétation de la forme composée, par juxtaposition, de ces primitives.

III.1 - Approximation polygonale des contours des formes

Le processus de l'approximation polygonale est basée sur le calcul pour chaque point $p_i = (x_i, y_i)$ significatif (sommet d'un angle, ou point d'inflexion) de la différence absolue $| s_d(i) - s_g(i) |$

$$\text{où } s_d(i) = t_{an}^{-1} [(y_{i+k} - y_i) / (x_{i+k} - x_i)]$$

$$\text{et } s_g(i) = t_{an}^{-1} [(y_{i-k} - y_i) / (x_{i-k} - x_i)]$$

k est pris égal à 5.

L'ensemble des sommets du polygone approchant le contour est constitué de tous les points significatifs $p_j = (x_j, y_j)$ qui satisfont à la condition suivante :

$$| s_d(j) - s_g(j) | < \sigma$$

où σ est un seuil évalué expérimentalement à 0,6 rd.

L'approximation polygonale est caractérisée par deux critères :

. l'erreur d'approximation définie par le rapport S/L , où L est le périmètre du polygone et S la somme des distances entre les points du contour et ceux du polygone.

. le degré de convexité égal à la moyenne ϕ des angles entre les droites joignant chaque point du contour aux deux sommets du polygone extrémités de l'arc à qui il appartient.

Le choix de cette approximation polygone est guidée par deux raisons principales. D'abord la grande simplicité de l'algorithme mis en oeuvre ensuite et surtout la simplicité relative des formes géométriques des pièces mécaniques qui font l'objet de manipulations. Les primitives sont les segments de droites formant les côtés du polygone.

III.2 Interprétation des formes au moyen d'un langage : L.A.P.

(Langage décrivant l'Assemblage des Pixels)

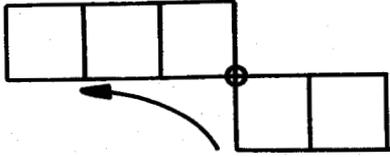
III.2.1. - Construction du langage L.A.P.

III.2.1.1. - Définitions

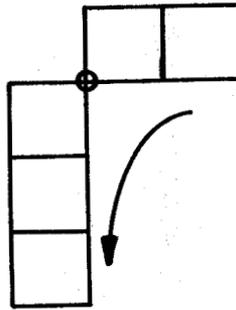
On appelle vocabulaire un ensemble fini dont les éléments sont appelés lettres.

Une phrase sur un vocabulaire est une suite ordonnée de lettres, représentée par simple juxtaposition de ces lettres.

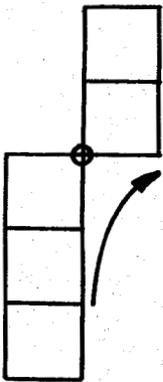
L'ensemble des phrases sur un vocabulaire V est noté V^* . Cet ensemble possède l'opération interne associative de concaténation. Il possède la structure du monoïde libre.



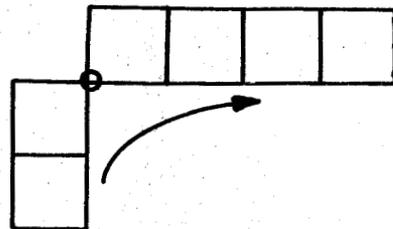
$$(-a)^2 \ a \ (-a)^3$$



$$(-a)^2 \ (-b) \ (-c)^3$$



$$c^3 \ b \ c^2$$



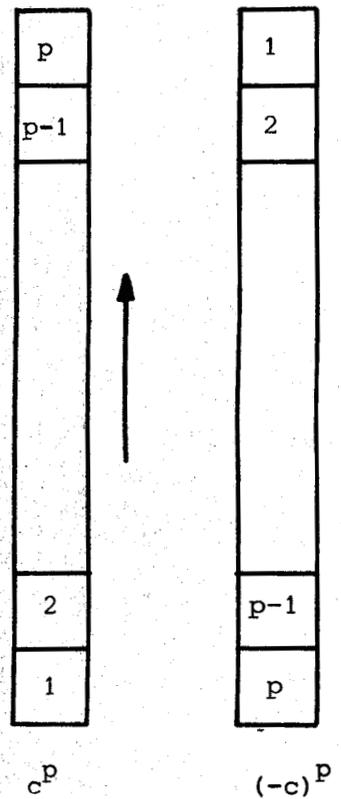
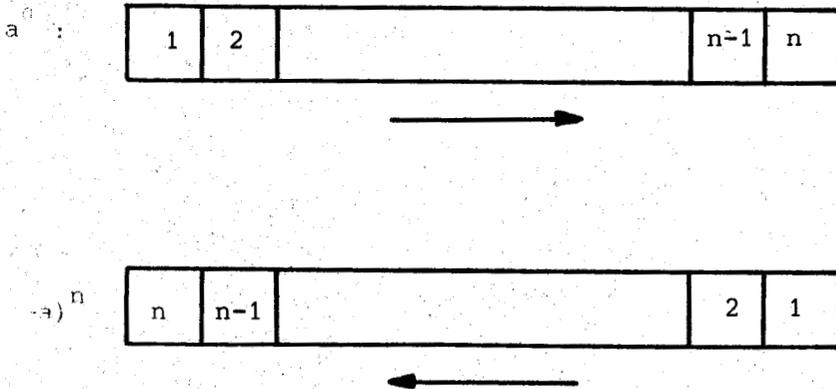
$$c^2 \ b \ a^4$$

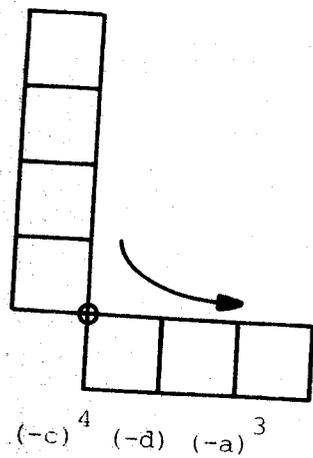
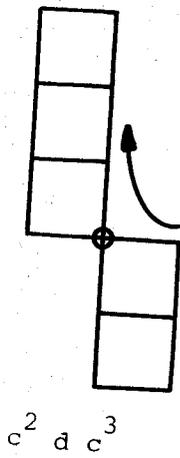
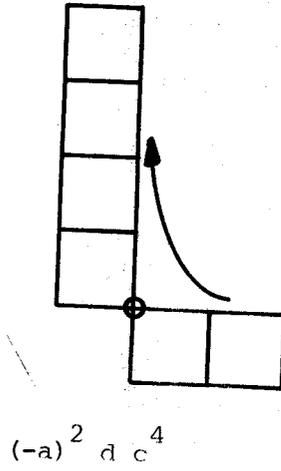
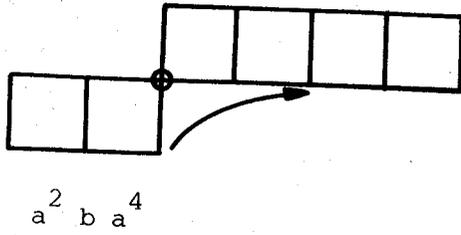


III.2.1.2. - Vocabulaire du langage L.A.P.

On construit un vocabulaire $V = \{ a, b, c, d, - \}$ dont les lettres 'a', 'b', 'c', 'd', et '-' permettent de décrire les chaînes de pixels dans l'image à la manière suivante :

- les lettres 'a' et 'c' décrivent respectivement les blocs horizontaux et verticaux de pixels (figure 4)
- les lettres 'b' et 'd' décrivent les connexités entre blocs de pixels (figure 5)
- le signe '-' indique le sens de déplacement à l'intérieur des blocs de pixels où le sens des connexités à réaliser entre blocs.





BUS
LILLE

III.2.1.3. - Définition du langage L.A.P.

Le langage L.A.P. est un sous-ensemble de l'ensemble des phrases obtenues par concaténation des lettres du vocabulaire v

L A P C v^*

ce langage permet l'interprétation des formes grâce à la description de leurs primitives. En effet, toute forme F apparaît comme une juxtaposition de primitives f_1, f_2, \dots, f_n ; dont chacune d'elles est décrite par une phrase φ_i $1 < i < n$ du langage L.A.P. / Donc la forme peut être interprétée par une phrase Φ construite en concaténant les phrases $\varphi_1, \varphi_2, \dots, \varphi_n$:

$$\Phi = \varphi_1 \cdot \varphi_2 \cdot \dots \cdot \varphi_n$$

III.2.2. - Description des primitives au moyen du langage L.A.P.

Le choix des primitives sous forme de segments de droites réduit le problème de la description des primitives à celui de la digitalisation des segments de droites.

Plusieurs auteurs ont étudié les critères qu'un arc digital doit satisfaire pour être une digitalisation dans l'image d'un arc réel convexe (c'est-à-dire d'un segment de droite)/19//20//21/ .

Et il a été montré en particulier/22/ que la " propriété de la corde "/23/ est une condition nécessaire et suffisante pour qu'un arc digital soit la digitalisation d'un segment de droite.

III.2.2.1. - La propriété de la corde

Soit un arc digital S et deux points P et Q de cet arc, on dit que le segment de droite réel P Q approche l'arc S si, pour tout point réel (x,y) de P Q, il existe un point (i,j) de S tel que $\text{Max} [|i-x|, |j-y|] < 1$. (cette expression est essentiellement la fonction distance standard dans le 8 voisinage |24|).

L'arc S possède la propriété de la corde si, pour tout couple de points (P,Q) de S, la corde P Q approche l'arc S.

Le résultat principal obtenu grâce à cette propriété est que pour qu'un arc soit la digitalisation d'un segment de droite, il faut et il suffit qu'il possède la propriété de la corde.

III.2.2.2. - Conséquences de la propriété de la corde

A partir de la propriété de la corde, on peut établir certains critères que doit satisfaire un arc digitalisant un segment de droite.

- 1 - l'arc digital peut être composé d'un seul type de blocs de pixels : blocs type horizontal ou type vertical.
- 2 - les longueurs de ces blocs diffèrent au plus d'une unité.
- 3 - si on distingue deux blocs de longueurs différentes, soit L et L' ces longueurs ($|L - L'| = 1$) et si le nombre de blocs de longueur L par exemple est supérieur à celui des blocs de longueur L' alors tout bloc de longueur L' est adjacent à deux blocs de longueur L.
- 4 - les blocs de longueur L forment des chaînes de blocs qui peuvent avoir deux longueurs possibles. Ces longueurs diffèrent d'une unité.

L'arc digital qui possède la propriété de la corde est formé d'une suite de chaînes de pixels connexes et identiques. Ces chaînes, appelées périodes ou motifs, peuvent spécifier les attributs du segment de droite digitalisé par l'arc.

III.2.2.3. - Description des segments de droites au
moyen du langage L.A.P.

Quand la courbe à digitaliser est un segment de droite dont la pente angulaire α est strictement comprises entre -45° et 45° , il a été montré /22/ en utilisant la grille standard d'intersection /24/ que sa digitalisation dépend seulement de ses intersections avec les lignes verticales de la grille.

On observe un résultat analogue dans la digitalisation par chaînes de pixels. En effet, le segment de droite précédent est digitalisé par un arc formé seulement de blocs de pixels de type horizontal.

Pour décrire les segments de droites digitalisés, on se contentera d'abord de ne considérer que les segments de droites dont la pente $\tan \alpha = m/n$ est comprise dans l'intervalle $] 0 \ 1]$ ($m \neq 0, n > m$). Le grand avantage qu'offre la description des segments de droites est de ne s'attacher qu'aux motifs (ou périodes) correspondants.

La phrase décrivant les motifs de ces segments se présente sous la forme :

$$(a_1^{n_1} b) (a_2^{n_2} b) \dots (a_p^{n_p} b) / \quad (1)$$

. Le deuxième critère découlant de la propriété impose des contraintes sur les valeurs des exposants n_1, n_2, \dots, n_p :

$$\forall_{i,j} \ 1 < i,j < p \quad |n_i - n_j| \leq 1 \quad (2)$$

Les longueurs de blocs horizontaux sont données par

$$N_1 = E(n/m) \quad (E(.) \text{ désigne la fonction "partie entière"})$$

$$N_2 = E\left(\frac{N-M}{m}\right) + 1$$

. La phrase (1) devient en appliquant également le troisième critère :

$$(a^{N_1} b) (a^{N_2} b)^{k_1} (a^{N_1} b) (a^{N_2} b)^{k_2} \dots (a^{N_1} b) (a^{N_2} b)^{k_l} \quad (3)$$

si $n - m N_1 > m/2$

$$(a^{N_2} b) (a^{N_1} b)^{k_1} (a^{N_2} b) (a^{N_1} b)^{k_2} \dots (a^{N_2} b) (a^{N_1} b)^{k_l} \quad (4)$$

si $m.N_2 - n > m/2$

. Le quatrième critère impose des contraintes sur les valeurs des exposants k_1, k_2, \dots, k_l

$$\forall_i k_i \in \{K_1, K_1 + 1\} \quad (5)$$

Dans (3) le nombre K_1 et l'indice l sont donnés par :

$$l = m N_2 - n$$

$$K_1 = E (n - m N_1 / l)$$

Alors que dans (4) ils sont donnés par :

$$l = n - m N_1$$

$$\text{et } K_1 = E (m N_2 - n / l)$$

Cette procédure de description offre, outre la possibilité de décrire, un segment de droite digitalisé, par une phrase du langage L.A.P., l'avantage de tester si une phrase donnée décrit effectivement un arc digitalisant un segment de droite (critère découlant de la propriété de la corde) et de calculer éventuellement ses attributs (pente, longueur).

Exemples

1 . La phrase : $[(a^{n_1}b) (a^{n_2}b)^{m_1} (a^{n_1}b) (a^{n_2}b)^{m_2} \dots (a^{n_1}b) (a^{n_2}b)^{m_q}]^p$

où les exposants $n_1, n_2, m_1, m_2, \dots, m_q$ satisfont aux critères 1, 2, 3 et 4 est un segment de droite dont la pente $\tan \alpha$ et la longueur L sont données par :

$$\tan \alpha = (q + \sum_{i=1}^q m_i) / (n_1 \cdot q + n_2 \sum_{i=1}^q m_i)$$

$$L = p \cdot \sqrt{(q + \sum_{i=1}^q m_i)^2 + (n_1 \cdot q + n_2 \sum_{i=1}^q m_i)^2}$$

2 . La phrase : $(a^2b) (a^3b)^3 (a^2b) (a^3b)^4)^3$

décrit un segment de droite tel que :

$$\tan \alpha = 9/25 \quad 9/34$$

$$L = 3 \sqrt{706} \approx 79,7 \text{ unités de longueur}$$

3 . La phrase $(a^k b)^*$ décrit une droite dont la pente est :

$$\tan \alpha = \frac{1}{k}$$

A partir de la description des segments de droites dont les pentes sont comprises dans l'intervalle $|0 \ 1|$, il est facile de déduire, par symétrie soit par rapport aux axes du repère du plan de l'image, soit par rapport aux droites de directions $k \cdot \frac{\pi}{4}$ (où $k = 1, 3$) les phrases décrivant tous les autres segments de droites.

- symétrie par rapport aux axes ox et oy :

$$\begin{aligned} S(a, ox) &= a \\ S(b, ox) &= -d \\ S(c, ox) &= -c \\ S(d, ox) &= -b \\ S(a, oy) &= -a \\ S(b, oy) &= d \\ S(c, oy) &= c \\ S(d, oy) &= b \end{aligned}$$

- symétrie par rapport aux droites de directions $\frac{\pi}{4}$ et $\frac{3\pi}{4}$

$$\begin{aligned} S(a, \pi/4) &= c \\ S(b, \pi/4) &= b \\ S(c, 3\pi/4) &= -a \\ S(d, 3\pi/4) &= d \end{aligned}$$

Exemple

Les phrases $(c_1^{\ n_1} b) (c_2^{\ n_2} b)^{m_1} \dots (c_1^{\ n_1} b) (c_2^{\ n_2} b)^{m_q}$ et $(a_1^{\ n_1} b) (a_2^{\ n_2} b)^{m_1} \dots (a_1^{\ n_1} b) (a_2^{\ n_2} b)^{m_q}$ décrivent deux segments de droites symétriques par rapport à la première bissectrice .

IV - LOCALISATION ET ORIENTATION DES OBJETS DANS LA SCENE

Pour saisir un objet, un robot doit être capable de

- le localiser dans la scène
- d'identifier la face sur laquelle il repose
- et déterminer son orientation par rapport à une orientation de référence.

IV.1 - Localisation des objets

Pour localiser un objet dans la scène, il suffit de déterminer, indépendamment de toute rotation, la position d'un seul point lui appartenant. Si on choisit de considérer la rotation de l'objet par rapport à son centre de gravité G, sa position est déterminée en calculant les coordonnées x_G et y_G du point G au moyen des formules

$$x_G = \frac{1}{D} \sum_{j=1}^N \left(\sum_{i=1}^N i \cdot P_{ij} \right)$$

$$y_G = \frac{1}{D} \sum_{j=1}^N \left(\sum_{i=1}^N j \cdot P_{ij} \right)$$

où

$$D = \sum_{j=1}^N \left(\sum_{i=1}^N P_{ij} \right)$$

N est la dimension de la sous-image où apparaît l'objet à localiser.

et P_{ij} est un facteur qui dépend de la position du pixel de coordonnées i et j par rapport à l'objet.

- si le pixel (i, j) n'appartient pas à l'objet, il est mis au minimum de l'échelle de gris : $P_{ij} = 0$

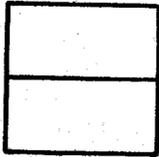
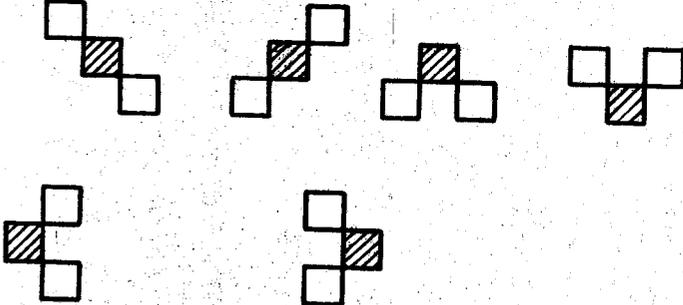
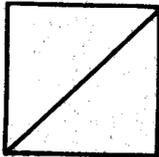
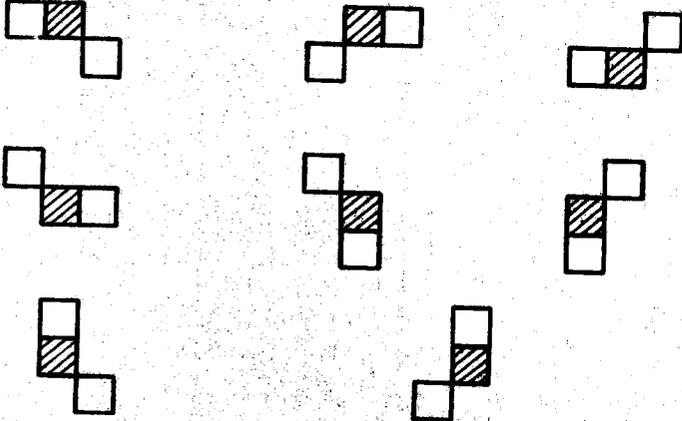
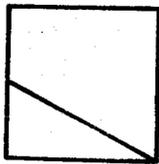
- s'il est situé à l'intérieur de l'objet, il est mis au maximum de l'échelle de gris; par exemple $P_{ij} = 31$

- enfin si c'est un pixel appartenant au contour, le facteur P_{ij} est donné par :

$$P_{ij} = m_{ij} \cdot g_{ij}$$

où m_{ij} est un poids représentant la contribution réelle du pixel dans la longueur du contour. La table suivante montre comment ce poids est fixé compte tenu de la topologie locale des pixels adjacents.

g_{ij} est le niveau de gris de ce pixel.

CONFIGURATIONS	Contributions en longueur	Poids
		1
		$\sqrt{2}$
		1.11

La précision dans le calcul des coordonnées du centre de gravité est nettement améliorée grâce à cette méthode. Elle est de l'ordre de 0,8 unité (ou pas de la caméra), alors que pour les méthodes classiques l'erreur peut s'étendre sur plusieurs pixels (figure 6) .

Le but recherché en augmentant le contraste de l'image a été surtout l'élimination des distorsions dues aux tâches et aux reflets.

En général la précision obtenue dépend de plusieurs facteurs :

- surface que la pièce occupe dans le champ de la caméra
- pas de la caméra
- pas de quantification de niveau de gris
- méthode d'extraction du contour
- et type de connexité (4 - voisinage ou 8 - voisinage) entre les pixels adjacents.

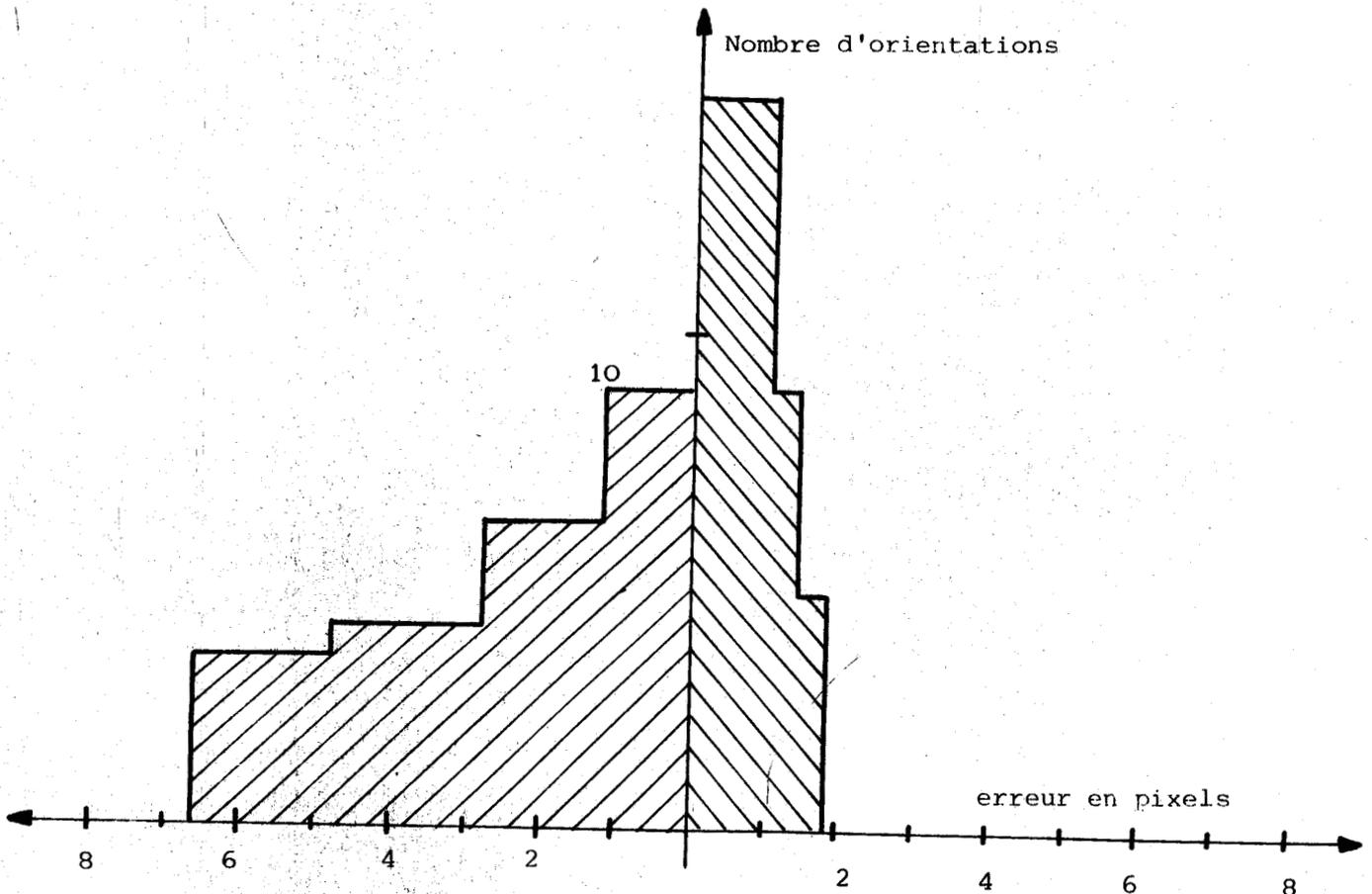


Figure 6 : à gauche histogramme d'erreurs relatif à la méthode générale
à droite histogramme d'erreurs relatif à la nouvelle méthode.

$$R(\theta_i) = r_i$$

où r_i et θ_i sont respectivement le module et l'argument du rayon vecteur \vec{r}_i

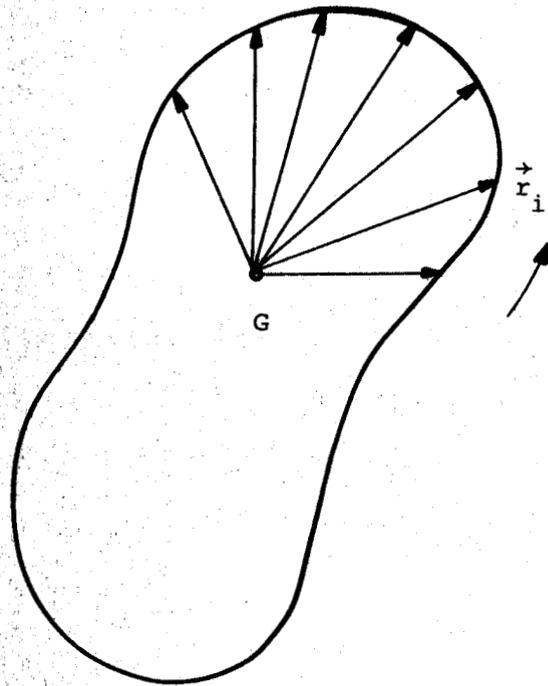


Figure 7

La fonction $\{ R(\theta_i) \mid i = 1, \dots, N \}$ est invariante par rapport à la position de l'objet dans le champ de vision. La détermination de l'orientation angulaire se réduit donc à une comparaison circulaire de la fonction $\{ R_a(\theta_i), i=1, \dots, N \}$ correspondant à l'orientation actuelle avec une fonction $\{ R_r(\theta_i), i=1, \dots, N \}$ correspondant à une orientation de référence .

IV.2 - Détermination de l'orientation des objets

La détermination de l'orientation angulaire d'un objet est un problème généralement délicat. Plusieurs méthodes ont été récemment proposées, pour tenter de les résoudre /16/, /17/, /18/. Parmi ces méthodes on distingue trois principales :

Méthode de corrélation

Elle est basée sur le principe de corrélation entre l'image actuelle et une image de référence. Un certain nombre de réalisations n'utilisent pas la totalité des informations fournies par l'image, mais se contentent de balayer l'objet étudié par un cercle centré au centre de gravité de cet objet. On obtient ainsi une fonction caractéristique $H(\alpha)$.

La fonction de corrélation $K(\alpha')$ est donnée par l'équation suivante :

$$K(\alpha') = \int_0^{2H} H(\alpha) \cdot H_r(\alpha + \alpha') d\alpha$$

où $H_r(\alpha)$ est la fonction caractéristique de référence.

La fonction $K(\alpha')$ est maximum pour la valeur $\alpha' = \sigma$ correspondant à la rotation de l'objet, par rapport à son centre de gravité, à partir de la position de référence.

Méthode du codage polaire

Le contour de l'objet est balayé radialement en N pas angulaires égaux à partir du centre de gravité. Une fonction discrète $\{ R(\theta_i), i = 1, \dots, N \}$ est définie en calculant à chaque pas le rayon entre le centre de gravité et le contour (figure 7)

Méthode du moment d'inertie de la forme

Cette méthode est basée sur la détermination du déplacement angulaire de l'objet à partir du calcul de moments d'inertie /17/. Mais cependant elle n'est pas générale, car elle donne des résultats erronés pour les objets qui possèdent plusieurs axes de symétrie.

IV.2.1. - Méthode de l'axe minimum d'inertie

A chaque forme dans l'image, on associe une fonction $I(x,y)$ telle que

$$I(x,y) = \begin{cases} 1 & \text{si le pixel de coordonnées } x \text{ et } y \text{ appartient à la forme} \\ 0 & \text{sinon} \end{cases}$$

Le moment M du second ordre par rapport à une droite Δ passant par le centre de gravité $G(x_G, y_G)$ d'une forme, avec l'angle θ , est obtenue par sommation des carrés des distances entre la droite Δ et les pixels de la forme (figure 8)

$$M = \sum_{x=i}^N \sum_{y=1}^N \frac{(y-y_G) - (x-x_G) \tan \theta}{1 + \tan^2 \theta} I(x,y)$$

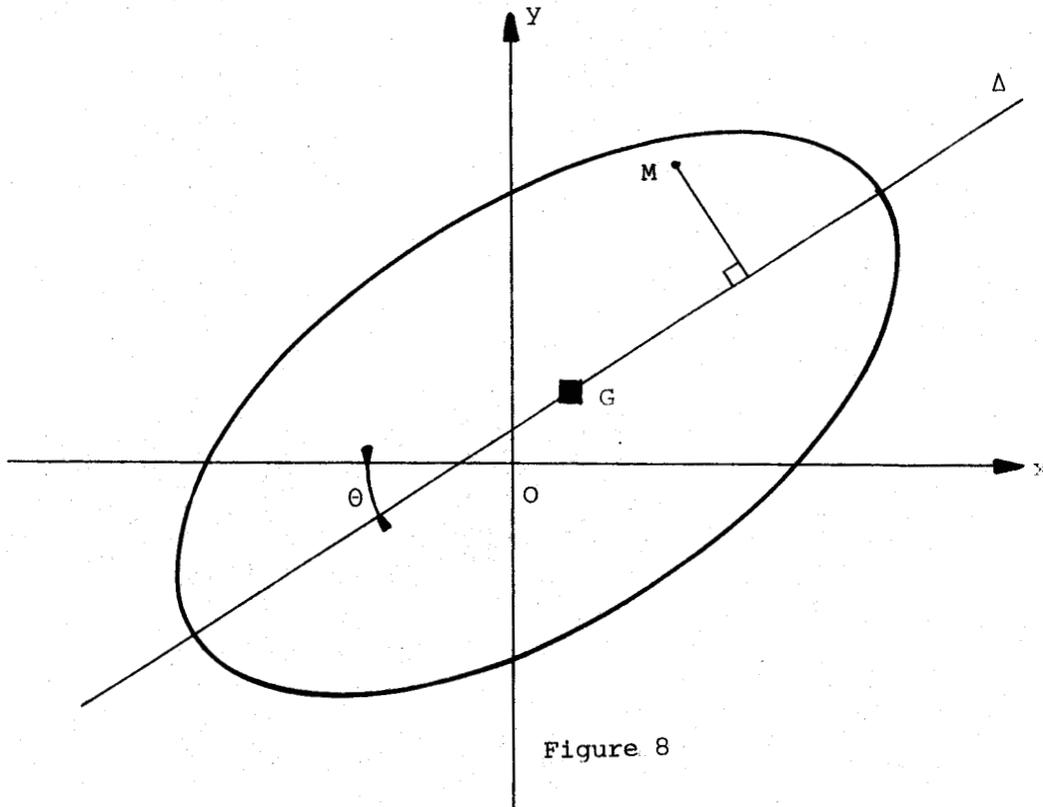


Figure 8

Le moment M varie uniquement en fonction de l'angle θ de rotation de la droite Δ . En variant, M passe par un minimum quand l'angle de rotation prend la valeur θ_m donnée par l'équation.

$$\theta_m = \frac{1}{2} \tan^{-1} \frac{1}{2} \frac{\bar{Y}^2 - \bar{X}^2}{\bar{XY}} + \text{sign}(\bar{XY}) \cdot \Pi/4 + n \Pi \quad n = 0,1$$

où

$$\bar{X} = \frac{1}{N} \sum_{x=1}^N \sum_{y=1}^N (x-x_G)^2 I(x,y)$$

$$\bar{Y} = \frac{1}{N} \sum_{x=1}^N \sum_{y=1}^N (y-y_G)^2 I(x,y)$$

et

$$\bar{XY} = \frac{1}{N} \sum_{x=1}^N \sum_{y=1}^N (x-x_G)(y-y_G) I(x,y)$$

La droite Δ_m dont la pente angulaire est θ_m est un axe principal d'inertie de la forme. Sa direction permet de déterminer l'orientation de cette forme avec une incertitude de Π radians. Cette ambiguïté peut être levée si on est capable de fixer la valeur de n ($n = 0,1$) dans l'équation (3).

Pour ce faire, on a choisi de déterminer un point particulier de la forme appelé centre de l'aire et noté A dont les coordonnées x_A et y_A sont données par les formules

$$\sum_{x=1}^N \sum_{y=1}^N \text{signe}(x-x_A) I(x,y) = 0 \quad (7)$$

$$\sum_{x=1}^N \sum_{y=1}^N \text{signe}(y-y_A) I(x,y) = 0 \quad (8)$$

et d'orienter l'axe principal Δ_m grâce au vecteur \vec{GA} (figure 9)

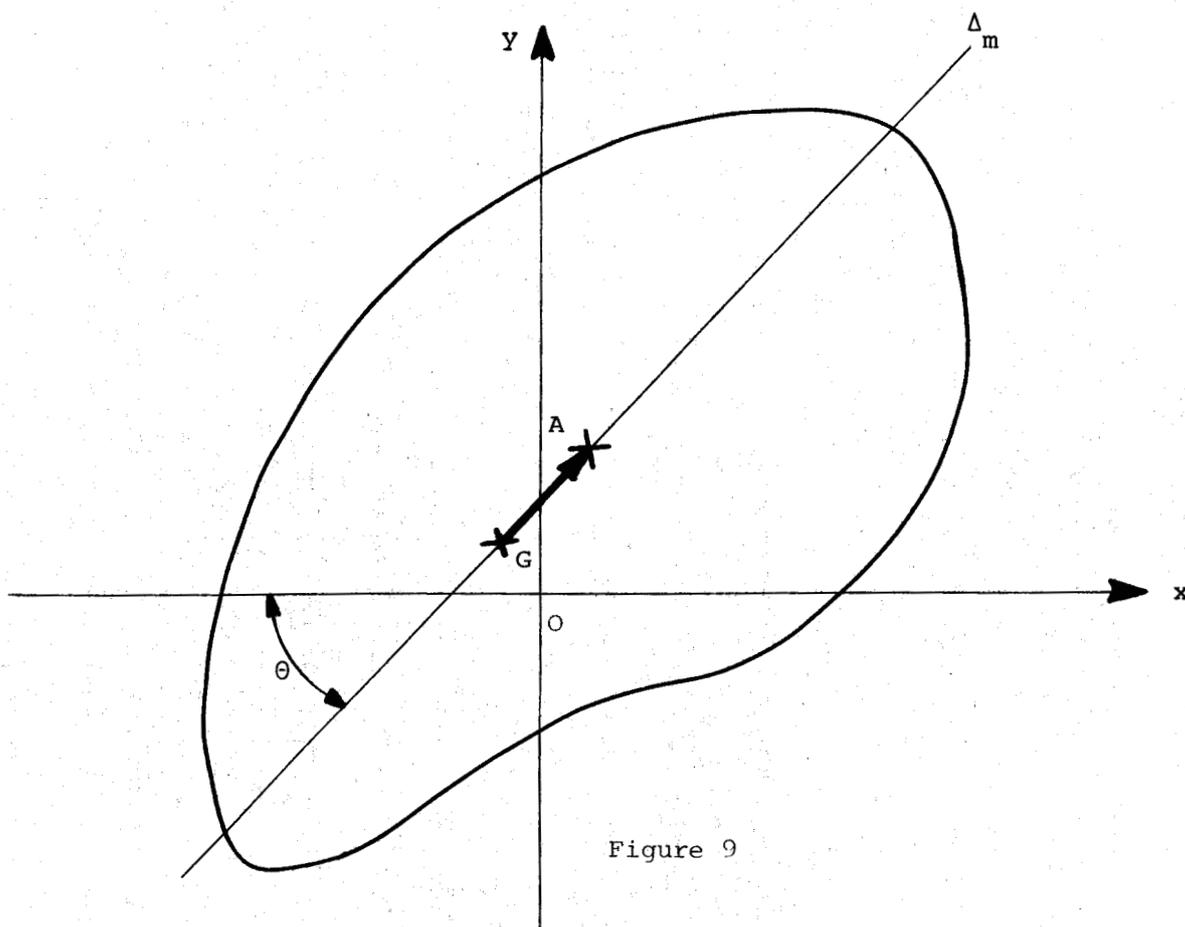


Figure 9

Cette méthode de détermination de l'orientation appelle deux principales remarques :

- . la première concerne le cas où la forme a deux axes de symétrie. Il convient alors de coïncider l'axe minimum d'inertie avec l'un de ces axes.
- . la seconde concerne le cas où la forme a plus de deux axes de symétrie. En effet l'orientation ne peut être déterminée par cette méthode, car dans l'équation (3) on a :

$$\bar{Y}^2 - \bar{X}^2 = 0$$

et $\bar{XY} = 0$

On envisagera pour cette raison une méthode plus générale basée sur le codage polaire.

IV.2.2. - Exploitation de la méthode du codage polaire

A chaque forme, on associe deux signatures polaires $\{ R_F(\theta_i) \ i=1, \dots, N \}$ et $\{ R_P(\theta_i) \ i=1, \dots, N \}$ correspondant à ses deux faces. Ces deux signatures sont telles que :

$$R_P(\theta_i) = R_F(-\theta_i) \ i=1, \dots, N$$

La procédure pour déterminer l'orientation et la face se déroule en deux étapes :

- Dans la première, on cherche à déterminer la direction d'un axe de symétrie particulier de la forme. Cette direction α_0 est choisie telle que la composante paire de la signature de la forme atteint le maximum pour la valeur α_0 , c'est à dire que :

$$P(\alpha) = \frac{1}{2} \sum_{i=1}^N R(\theta_i + \alpha) + R(-\theta_i - \alpha)$$

soit maximum quand la variable α prend la valeur α_0 .

Mais l'axe de symétrie de direction α_0 ne permet de déterminer l'orientation de la forme qu'avec une incertitude de Π radians.

Pour lever cette ambiguïté, on génère au cours de la seconde étape, quatre fonctions :

$$\{ R(\theta_i + \alpha_0) \quad i = 1, \dots, N \}$$

$$\{ R(\theta_i + \alpha_0 + \Pi) \quad i = 1, \dots, N \}$$

$$\{ R(-\theta_i - \alpha_0) \quad i = 1, \dots, N \}$$

$$\{ R(-\theta_i - \alpha_0 - \Pi) \quad i = 1, \dots, N \}$$

Ces fonctions correspondent aux quatre positions qu'on peut rencontrer : deux orientations possibles pour chacune des deux faces.

La détermination de l'orientation angulaire d'une forme et l'identification de sa face se ramène à une comparaison circulaire des quatre fonctions précédentes avec quatre fonctions de références.

La fonction de comparaison circulaire, entre deux signatures, qu'on a choisie à cette occasion est la corrélation.

$$C(\alpha) = \frac{1}{N} \sum R(\theta_i) R_r(\theta_i + \alpha)$$

Elle passe par un maximum quand α prend une valeur égale au déplacement angulaire de la forme.

La précision dans les calculs de l'orientation par cette méthode dépend essentiellement de deux facteurs :

- pas d'échantillonnage spatial de l'image
- et pas de balayage radial de la forme

Une résolution angulaire limite théorique est calculée en fonction du rayon maximum du contour par l'équation.

$$\alpha_{\min} = \tan^{-1} \frac{1}{r_{\max}}$$

où r_{\max} est le rayon maximum du contour.

Cette méthode plus générale que celle basée sur la recherche de l'axe minimum d'inertie permet la détermination de l'orientation et de la face d'une forme à une vitesse industriellement acceptable.

CONCLUSION

Les améliorations apportées aux différentes opérations de traitement d'images : extraction, filtrage, approximation du contour ... ont permis de limiter sensiblement le coût, l'espace mémoire et temps de calcul.

Ces opérations sont relativement complexes et leur mise en oeuvre apparaît lourde; néanmoins, au vu de l'importance particulière que revêt cette procédure de traitement d'images, à savoir la modélisation des formes des pièces à partir d'images digitalisées, cette complexité est à la mesure de tâche proposée.

Ainsi, par exemple, il apparaît clairement, en comparant la méthode du gradient améliorée à celle de Hueckel, ou la méthode appliquée pour la segmentation du contour à la méthode d'interpolation de Ramer, que notre souci principal est la recherche d'un compromis temps calcul-précision-espace mémoire qui rend les méthodes appliquées plus opérationnelles.

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

- /1/ A. ROSENFELD
" Picture Processing by computer "
Academie Press, New-York, 1969
- /2/ R.C. GONZALEZ, P. WINTZ
" Digital image Processing "
Addison - Wesley, Advanced Book Program Reading, Massachusetts, 1977
- /3/ J.W. Mc KEE, J.K. AGGARWAL
" Finding the Edges of the surface of three-dimensional curved
objects by computer "
Pattern Recognition, vol.7 n° 1/2 June 75
- /4/ H. MAITRE, B. FAUST
" Détection des contours d'image dans un but de reconnaissance
des formes "
Congrès AFCET - IRIA - " Reconnaissance des formes et traitement
d'images "
Chatenay - Malabry 21.22.23 Février 1978
- /5/ K.J. UDUPA, I.S.N. MURTHY
" Machine visualization of three-dimensional objects via
skeletal transformations "
I.E.E.E. SMC, vol.7 n°6 June 77
- /6/ J.M.S. PREWITT
" Object enhancement and extraction "
In Picture Processing and Psychopictorics, (E.S. Lipkin
and A. Rosenfeld, eds),
Academic Press, New-York, 70
- /7/ HUECKEL
J.of ASS. for Comp. Math - vol 20 n°4 - 73

- /8/ M. YACHIDA, S. TSUJI
" A versatile machine vision system for complex industrial parts "
I.E.E.E. Transactions on computers, vol C-26, n°9 Septembre 77
- /9/ A. ROSENFELD, Y.NAKAGAWA
" Some experiments on variable thresholding "
Pattern recognition vol.11, n°3, 79
- /10/ A. ROSENFELD
" Picture Processing by Computer "
NEW-YORK : Academic, 69
- /11/ F. ATTNEAVE
" Some informational aspects of visual perception "
Psychol. Rev. 61, pp 183.193, 1954
- /12/ H. FREEMAN
" On the classification of line-drawing data "
In model for the perception of speech and visual form. W Wathen
Dunn, Ed Cambridge Mass. : MIT, Press, 1967
- /13/ B. ROSENBERG
" The analysis of convex blobs "
In Computer graphics and Image Processing, vol 1, 1972
- /14/ A. ROSENFELD, M. THURSTON
" Edge and curbe detection for visual scene analysis "
I.E.E.E., Trans. Comp. vol .C-20, May 1971
- /15/ A. ROSENFELD, M. THURSTON, Y.H. LEE
" Edge and curbe detection : Further experiments "
I.E.E.E. Trans. Comp, vol C-21, July 1972

- /16/ J.D. DESSIMOZ
" Visual identification and location in a multi-object environment by contour tracking and curvature description "
8 th International Symposium on Industrial Robots
30.5.1978 - 1.6. 1978. Stuttgart, West Germany
- /17/ J.P. FOITH
" Lageerkennung von beliebig orientierten werkstücken ans der form ihrer silhouetten "
8 th International Symposium on Industrial Robots
30.5.78 - 1.6.78 Stuttgart, West Germany
- /18/ P. MARTINI, G. NEHR
" Recognition of angular orientation of objects with the help of optical sensors "
The Industrial Robot, June 79.
- /19/ G.U. MONTANARI
" A note on minimal length polygonal approximation to a digitized contour "
Commun. Ass. Comput. Mach., vol 13, pp 41-47, 1970
- /20/ J. SKLANSKY, R.L. CHAZIN, B.J. HANSEN
" Minimum - perimeter polygons of digitized silhouettes "
I.E.E.E. Trans. Comput., vol C-21, pp 260-268, Mar.72
- /21/ L. HODES
" Discrete approximation of continuous convex blobs "
SIAM J. Appl. Math., vol 19, pp 477-485, 72
- /22/ A. ROSENFELD
" Digital Straight Line Segments "
I.E.E.E. Trans. Comput., vol C-23, N°12 , pp 1264-1269, December 74.

/23/ A. ROSENFELD

" Arcs and Curves in digital pictures "

J. Ass. Comput. Mach., vol.20, pp 81-87, 73

/24/ A. ROSENFELD

" Distance functions on digital pictures "

Pattern Recognition, vol.1, pp 33-61, 68

C H A P I T R E I I I

RECONNAISSANCE AUTOMATIQUE DES ELEMENTS

D'ASSEMBLAGE

CHAPITRE III

RECONNAISSANCE AUTOMATIQUE DES ELEMENTS D'ASSEMBLAGE

La théorie des langages offre des outils très féconds , non seulement sur le plan de la potentialité théorique, mais également comme modèles de phénomènes réels, comme outils d'analyse et de synthèse de données dont l'organisation structurée doit être prise en compte.

C'est en particulier en robotique industrielle que plus récemment, il est apparu très fertile d'utiliser les outils grammaticaux pour modéliser des familles de formes dans un but de reconnaissance.

Incontestablement, l'utilisation des outils de la théorie des langages a donné de nouvelles possibilités à la reconnaissance des formes; mais en même temps elle a suscité des problèmes nouveaux, qui n'avaient pas été abordés auparavant.

Un des problèmes, a priori de peu d'intérêt dans le cadre de la théorie des langages, est devenu crucial si l'on pense à ses applications en reconnaissance des formes. C'est celui de l'apprentissage : c'est-à-dire, la possibilité d'abstraire automatiquement d'un ensemble de données le modèle le plus adéquat pour le représenter. L'apprentissage automatique est un sujet très général et d'un intérêt immense; à son intersection avec la théorie des langages, il prend le nom d' "Inférence grammaticale " ; c'est donc dans ce cadre que se place une part importante du travail qui va suivre.

I - PHASE D'APPRENTISSAGE

L'un des problèmes essentiels en reconnaissance des formes est celui de l'apprentissage : il s'agit de découvrir une organisation caractéristique d'un ensemble fini de paramètres ou primitives, cette organisation pouvant s'exprimer soit sous une forme statistique et métrique, soit sous une forme structurelle.

Dans le cas de la reconnaissance structurelle où l'on formalise la structure d'une forme par une grammaire, le problème de l'apprentissage repose sur celui de l'inférence puisqu'il s'agit de découvrir une grammaire modélisant une classe de formes, c'est-à-dire générant un langage qui décrit toutes les formes appartenant à cette classe.

I.1 - Définitions fondamentales

Les grammaires formelles sont des outils mathématiques permettant de décrire de façon condensée des langages appartenant à certaines familles; une grammaire G (ensemble fini de symboles) définira un langage noté $L(G)$.

I.1.1. - Définition de grammaire formelle

Une grammaire formelle est un quadruplet

$$G = (V_T, V_N, S, P)$$

où V_T est le vocabulaire terminal

V_N le vocabulaire non-terminal; V_T et V_N sont deux ensembles finis disjoints

S un élément de V_T ou axiome

P l'ensemble fini des règles de production

Un élément de P est de la forme

$$A \rightarrow B$$

$$\text{où } A \in (V_N \cup V_T)^* V_N (V_N \cup V_T)^*$$

$$B \in (V_N \cup V_T)^*$$

L'opération étoile '*' (ou "produit de kleene") est définie par :

$$L^* = \{ \Lambda \} \cup L \cup L^2 \dots \cup L^n \cup \dots$$

ou

$$L = \bigcup_{n=0}^{\infty} L^n$$

Les éléments de l'ensemble $(V_T \cup V_N)^*$ sont appelés phrases sur $V_T \cup V_N$.

Le langage engendré par une grammaire $G = (V_T, V_N, S, P)$ est l'ensemble des phrases sur V_T pouvant être engendrées par une suite de réécriture à partir de l'axiome S au moyen des règles de production :

$$L(G) = \{ x \in V_T^* : S \xrightarrow{*} x \}$$

I.1.2. - Grammaire régulière

Une grammaire est dite régulière ou de 'kleene' quand les règles de production sont de la forme

$$A \rightarrow a B \quad A, B \in V_N$$

ou

$$A \rightarrow a \quad a \in V_T$$

Le langage accepté par une grammaire régulière est dit " langage régulier ".

I.1.3. - Représentation des langages réguliers par des expressions régulières

Les expressions régulières constituent un formalisme permettant de définir les parties rationnelles d'un monoïde libre V_T^* , c'est-à-dire les langages rationnels sur V_T .

Le théorème de Kleene /1/ assure que la famille des langages rationnels est la même que celle des langages réguliers.

En conséquence, il peut donc paraître plus intéressant de chercher d'abord un modèle structural sous forme d'expressions régulières, pour faciliter ensuite l'inférence d'une grammaire régulière. En effet, les caractéristiques des expressions régulières apparaissent beaucoup plus clairement par répétition de chaînes de caractères.

I.2 - Inférence de grammaires régulières

I.2.1. - Position du problème

Le problème général de l'inférence grammaticale peut être formulé comme suit : soit un alphabet V_T , et deux parties I^+ et I^- de V_T^* ; est-il possible de trouver une grammaire G , telle que, si on note $L(G)$ le langage qu'elle engendre, on ait :

$$I^+ \subset L(G)$$

$$I^- \subset \bar{L}(G) \text{ (complémentation par rapport à } V_T^* \text{)}.$$

La grammaire G est alors dite " solution du problème d'inférence " et le couple $I = (I^+, I^-)$, appelé échantillon d'inférence, I^+ et I^- formant respectivement l'échantillon positif et l'échantillon négatif d'inférence.

De nombreux articles ont été consacrés à la résolution du problème d'inférence de grammaires régulières algébriques ou 'context sensitive'. Citons plus particulièrement FU /2/, /3/, qui passe en revue de nombreuses méthodes. La plupart des algorithmes possèdent le même défaut : le nombre considérable de grammaires étudiées et la difficulté de choisir parmi la multiplicité des solutions.

Pour contraindre fortement le problème d'inférence et lui assurer un cadre théorique solide, on ne s'intéressera par la suite qu'à la recherche d'une grammaire régulière.

Afin de simplifier le problème, on fixe trois contraintes sur les grammaires cherchées :

- l'échantillon négatif est vide : $I^- = \emptyset$
- l'alphabet de la grammaire est le même que celui de l'échantillon I^+ ; ceci afin d'éviter des règles arbitraires qu'il est toujours possible de rajouter.
- toute règle de la grammaire solution doit être utilisée au moins une fois pour la génération de I^+ ; ceci pour éliminer d'éventuelles règles improductives.

Sous ces trois conditions, une grammaire solution au problème de l'inférence est dite structurellement complète pour l'échantillon I^+ , le nombre de ces grammaires solutions est fini.

Le vrai problème de l'inférence se ramène donc à une question de choix d'une solution parmi un grand nombre.

I.2.2. - Rappel des principaux algorithmes d'inférence régulières

Ce n'est qu'à partir de la fin des années soixante que se sont développés les premiers algorithmes d'inférence, en particulier destinés à procurer des méthodes de synthèse des circuits séquentiels. Dès cette époque, on a vu apparaître diverses heuristiques de confusion d'états, explicites ou non, et la recherche dans l'échantillon des propriétés générales des langages auxquels celui-ci est supposé appartenir.

Parmi les publications traitant le problème de l'inférence, nous retenons trois articles bibliographiques très bien documentés /4/, /5/, /6/ présentant l'"état de l'art" au moment de leur parution: ils sont à la base de toute étude sur l'inférence grammaticale.

I.2.2.1. - Inférence régulière à partir de la grammaire canonique

On considère la grammaire régulière G_0 dite canonique, telle que :

$$L(G_0) = I^+$$

La grammaire canonique associée à l'échantillon de phrases $I^+ = \{ x_1, \dots, x_n \}$ est construite en associant à chaque phrase $x_i = a_{1i} a_{2i} \dots a_{ki}$ la suite des règles :

$$\begin{aligned} S &\rightarrow a_{1i} A_{1i} \\ A_{1i} &\rightarrow a_{2i} A_{2i} \\ &'' \\ &'' \\ A_{k-1i} &\rightarrow a_{ki} \end{aligned}$$

Le symbole S étant l'axiome, les vocabulaires terminal V_T et non-terminal V_N sont alors :

$$. V_T = \{ a_{ij} \}$$

$$. V_N = \{ A_{ij} \}$$

La grammaire canonique est une solution triviale qui conduit à un nombre de règles considérable pour de gros échantillons. D'où la construction d'un algorithme d'inférence qui procède par réduction de la grammaire canonique.

I.2.2.1.1. - Construction d'une grammaire dérivée

=====

En prenant une partition $V_{Np} = \{ B_S, B_1, B_2, \dots, B_l \}$ de l'ensemble des éléments non-terminaux de la grammaire canonique V_N , on peut construire une grammaire dérivée G_D définie sur la base suivante :

- . l'ensemble des éléments terminaux V_{T_D} est identique à celui de G_0
- . l'ensemble des éléments non-terminaux est une partition de V_N
- . B_S est l'axiome de G_D (contient au moins S)
- . R_D l'ensemble des règles, est défini de la façon suivante :

$$B_i \rightarrow a B_j \quad A, C \in V_N \text{ tel que } A \rightarrow a Z$$

$$\text{avec } A \in B_i \text{ et } C \in B_j$$

Quel que soit la partition de V_N il est évident que la grammaire dérivée G_D engendre l'échantillon I^+ . De plus I^+ étant structurellement complet, on montre qu'il existe une partition de V_N telle que G_D soit équivalente à la grammaire G recherchée.

Cette méthode n'est pas satisfaisante car le nombre de partitions distinctes de V_N croit rapidement avec le cardinal de V_N . Ainsi le nombre de grammaires solutions est très grand.

I.2.2.1. - Inférence par grammaire canonique dérivée

I.2.2.2.1. - Dérivée formelle d'un échantillon de phrases
 =====

Soit A un échantillon de phrases, la dérivée formelle de A par rapport à une lettre a est définie par :

$$D_a A = \{ x : a x \in A \}$$

I.2.2.2.1. - Grammaire canonique dérivée
 =====

On définit la grammaire canonique dérivée G_{CD} par :

$$G_{CD} = (U_T, U_N, R, \sigma)$$

- . le vocabulaire terminal V_T est formé par l'ensemble des symboles terminaux de I^+
- . le vocabulaire non terminal U_N est égal à l'ensemble des dérivées distinctes de I^+
- . R est l'ensemble des règles définies comme suit :

$$A_i \rightarrow a A_j \quad \forall \quad A_i, A_j \in U_N \quad D_a A_i = A_j$$

$$A_i \rightarrow a \quad \forall \quad A_i \in U_N \quad \lambda \in D_a A_i$$

où λ est l'élément neutre: $D_{\lambda} I^+ = I^+$

On constate, d'après la définition, que la grammaire canonique est une grammaire dérivée car l'ensemble V_N forme une partition des éléments non terminaux de la grammaire canonique. Sa recherche consiste donc à minimiser le nombre d'éléments non-terminaux de la grammaire canonique.

Cette méthode présente les mêmes inconvénients que la précédente dès que l'échantillon d'inférence ne possède pas trop de redondances, car alors la réduction de la grammaire devient illusoire.

I.2.2.3. - Inférence basée sur les " k-finales "

Cette méthode d'inférence régulière est à la base de nombreux travaux /11/, car elle utilise une heuristique simple de confusion d'états. Elle consiste à chercher dans les états de la grammaire canonique la relation de " k-équivalence ".

La partition sur les états de la grammaire canonique induite par la relation de k-équivalence produit une grammaire dérivée qui est la solution recherchée.

I.2.2.3.1. - " k-finale " d'une dérivée

=====

On appelle k-finale d'une dérivée $D_a A$ avec $a = a_1 a_2 \dots a_n \in V_T^*$ et $A \in L(G)$ l'ensemble $g(a, A, k) = \{ x : x \in D_a A \text{ et } |x| < k \}$

I.2.2.3.2. - Etats " k-équivalents "

=====

Considérons la grammaire canonique dérivée G_{CD} dont les états non-terminaux sont $\sigma, A_1, A_2, \dots, A_n$. Deux états A_i et A_j sont dits k-équivalents si et seulement si :

$$g(x_i, I^+, k) = g(x_j, I^+, k)$$

avec $D_{x_i} I^+ = A_i$ et $D_{x_j} I^+ = A_j$

La grammaire d'ordre k est la grammaire dérivée formée par la partition induite par la relation de " k-équivalence ". Une façon simple de réaliser ceci est d'utiliser un algorithme de minimisation des grammaires régulières, et de l'arrêter pour une valeur arbitraire de k.

D'une façon générale cette méthode donne rarement des solutions satisfaisantes, en particulier à cause de la grande sensibilité du paramètre k. En effet, d'une valeur à une autre de k, il peut se produire des modifications très importantes sur les grammaires inférées.

Elle est donc peu adaptée aux problèmes de la reconnaissance des formes, où la grammaire cherchée doit tendre à généraliser les propriétés globales de l'échantillon d'inférence.

I.2.2.4. - Inférence fondée sur le "Lemme de l'étoile"

Le rappel précédent de méthodes d'inférence régulière a permis de montrer que le choix d'un algorithme d'inférence relevait d'un arbitraire puisque la théorie même à la preuve de l'existence d'un très grand nombre de solutions.

Pour arriver à une solution pratique, on cherchera donc un algorithme s'appuyant sur une information supplémentaire. Or on sait qu'une façon d'acquérir de l'information est de faire intervenir un opérateur. Mais une autre est d'utiliser pour heuristique les propriétés des grammaires à inférer.

Ainsi le Lemme de l'étoile /7/ a permis de suggérer /8/ /9/ une méthode d'inférence d'expressions régulières dont l'heuristique s'apparente étroitement à celles de Chomsky et Solomonof /10/. Cette inférence est fondée sur la recherche dans les phrases de l'échantillon de sous-phrases v apparaissant dans les décompositions $u v^k w$ avec $k > 2$.

I.2.2.4.1. - Lemme de l'étoile

=====

Soit L , un langage régulier sur l'alphabet V_T

$\exists p \in \mathbb{N} : \text{si } x \in L, \text{ avec } |x| > p$

alors x peut s'écrire :

$x = u v w$ avec $|v| > 1$, et $\forall k > 0 : u v^k w \in L$

I.2.2.4.2. - Exposé de l'algorithme d'inférence

=====

A chaque étape de l'algorithme, il s'agit de découvrir une récursion caractéristique grâce à l'examen des sous-phrases répétées dans les phrases de l'échantillon. Pour cela on procède en trois phases :

- la première consiste à chercher dans les phrases les plus longues de l'échantillon toutes les sous-phrases v telles qu'au moins une phrase puisse se décomposer sous la forme :

$$u v^k w \text{ avec } k > 2$$

On ne retient que les phrases suffisamment longues où on doit a priori d'après le lemme de l'étoile, découvrir les répétitions de sous-phrases appelées hypothèses qui sont les traces de récursions cherchées.

- la seconde phase revient à sélectionner certaines hypothèses. En effet, à chaque hypothèse émise est associée une fonction numérique dite consistance qui évalue son importance.

Si la phrase x se décompose en $u v^k w$, on appelle consistance de v par rapport à x la valeur.

$$C_{(v,x)} = k |v|$$

Cette consistance tient compte à la fois du nombre de répétitions d'une hypothèse et de sa longueur. Elle constitue donc un critère intéressant pour sélectionner les hypothèses, car a priori les hypothèses de plus grande longueur sont les plus précises pour décrire la grammaire à inférer.

- la troisième phase consiste à "vérifier" sur tout l'échantillon, les hypothèses sélectionnées selon le critère de consistance, et choisir la meilleure.

Pour cela, on procède en deux pas :

- . Au premier, on élimine d'une part toutes les hypothèses dont la consistance est trop faible, comparativement à celle dont la consistance est la plus forte, d'autre part celles qui n'apparaissent que dans une seule phrase.

. Au second, il s'agit de "vérifier" les hypothèses restantes et choisir celle dont la consistance est la plus grande. Au cours de cette vérification on retiendra pour le calcul de la consistance toutes les décompositions de la forme $u v^k w$ avec $k \geq 1$.

$$C(a, I) = \sum_{x \in I} C(a, x)$$

où I est l'échantillon d'inférence.

Les répétitions de l'hypothèse choisie sont la trace d'une récursion. Pour introduire cette récursion, on réécrit l'échantillon en remplaçant les occurrences de v^k , où v est l'hypothèse retenue, par un symbole T représentant l'expression régulière.

$$T = v^+ \text{ avec } v^+ = v.v^*$$

Quand à une étape donnée de l'algorithme, les hypothèses émises n'ont plus assez de consistance, l'algorithme s'arrête. Cet arrêt est automatique, au vu de la consistance de la meilleure hypothèse.

I.3 - Modélisation par inférence des formes à reconnaître

On vient de passer en revue certaines méthodes d'inférence régulière qui montre bien que sur le plan théorique, le problème d'inférence tel qu'il est posé en I.2.1. est résolu pour les grammaires régulières. On a trouvé une multiplicité de solutions pour lesquelles l'échantillon d'inférence est structurellement complet.

Cependant, dans un cas réel d'inférence, précisément en vue d'une application à un problème de reconnaissance des formes, il n'est pas opératoire de disposer de l'ensemble des solutions. On cherche en effet une seule d'entre elles. Cette solution doit être connue explicitement, de façon à l'utiliser réellement dans les algorithmes où elle interviendra.

On se trouve donc devant un problème de choix parmi un grand nombre de grammaires solutions. Pour réaliser ce choix, on se donne un critère. Ce critère peut se présenter soit sous la forme explicite d'une fonction numérique que l'on pourra évaluer sur toutes les grammaires, soit comme le recours à une heuristique permettant de réduire l'ensemble de grammaires solutions et de choisir une parmi celles-ci.

En pratique, il est généralement plus avantageux d'avoir recours à des heuristiques car l'utilisation d'un critère explicite se heurte au grand nombre de grammaires sur lequel il faut évaluer ce critère.

I.3.1. - Application de l'heuristique " $u v^k w$ " pour la
modélisation des formes

Pour reconnaître les formes, il faut après leur description par des phrases du langage L.A.P., les modéliser. Cette modélisation est essentiellement un problème d'inférence : en effet, chaque forme est décrite par un ensemble de phrases; chacune de ces phrases fournit une description de la forme de la pièce correspondant à une orientation de cette pièce, donc modéliser une forme revient à inférer une grammaire à partir de l'échantillon des phrases qui la décrivent.

Au vu de la structure de ces phrases qui sont en fait composées en concaténant les phrases du langage L.A.P. qui décrivent les primitives de la forme, et dans le but de simplifier le problème d'inférence, on ne s'intéressera ici qu'à la recherche d'une grammaire régulière. On s'adresse donc à un modèle régulier qui est le plus simple des modèles syntaxiques.

La méthode appliquée pour inférer ce modèle est celle utilisant l'heuristique " u v^k w ". Ce choix est effectué surtout en raison de la grande facilité d'induire, dans le cas pratique d'inférence, des contraintes sur le modèle recherché.

I.3.1.1. - Echantillon de phrases associé à une forme

A chaque orientation de la pièce (on considère qu'en pratique, la pièce à analyser est contrainte à un nombre limité d'orientations) correspond une phrase qui décrit le contour de la pièce. Ce contour, approché par un polygone, est parcouru dans le sens positif à partir d'un sommet tel que la distance le séparant du centre de gravité soit minimale.

Les phrases correspondant aux différentes orientations de la pièce sont chacune le produit par concaténation des phrases du langage L.A.P. qui interprètent les primitives.

Or on a vu que ces phrases interprétant les primitives se présentent sous la forme :

$$f_1 = x^i$$

$$\text{ou } f_2 = (x^{i_1} y x^{i_2} y \dots x^{i_p} y)^m (x^{i_1} y x^{i_2} y \dots x^{i_k} y) \quad k < p$$

où $x \in \{ a, -a, c, -c \}$

$y \in \{ b, -b, d, -d \}$

x et $x^{i_1} y x^{i_2} y, \dots, x^{i_p} y$ représentent respectivement les motifs (ou périodes) des phrases f_1 et f_2

L'échantillon d'inférence correspondant à une forme F est constitué de phrases de la forme

$$F_i = f_{i_1} f_{i_2} \dots f_{i_m} \text{ avec } f_{i_j} \in \{f_1, f_2\} \text{ pour } j = 1, 2, \dots, m$$

La modélisation de cette forme se ramène donc à l'inférence d'une grammaire régulière à partir de l'échantillon

$$I = \{ F_1, F_2, \dots, F_q \}$$

où F_i est une interprétation de la forme correspond à une orientation donnée.

I.3.1.2. - Recherche des hypothèses dans l'échantillon

Il s'agit de rechercher dans les phrases F_i de l'échantillon I les sous-phrases qui se répètent pour faire ensuite l'hypothèse que celles-ci sont engendrées par plusieurs passages successifs de la même boucle d'un automate.

Dans notre cas particulier, il est évident que le choix des

motifs x et $x \overset{i_1}{y} x \overset{i_2}{y} \dots x \overset{i_p}{y}$ comme hypothèses est très commode.

Exemples

$$\begin{aligned} \text{En posant : } T &= a^+ \\ W &= (a \overset{j_1}{b} a \overset{j_2}{b} \dots a \overset{j_p}{b})^+ \end{aligned}$$

on déduit à partir de la phrase

$$F = a^n (a \overset{j_1}{b} a \overset{j_2}{b} \dots a \overset{j_p}{b})^m a \overset{j_1}{b} a \overset{j_2}{b} \dots a \overset{j_k}{b} \quad k < p$$

l'expression régulière

$$T W (x^{j_1} y x^{j_2} y \dots x^{j_k} y)$$

ou

$$x^+ (x^{j_1} y x^{j_2} y \dots x^{j_p} y)^+ (x^{j_1} y \cdot x^{j_2} y \dots x^{j_p} y)$$

En posant :

$$T = (c^3 d)^+$$

$$V = (c^2 d c^3 d c^2 d)^+$$

$$W = (a^3 b a^2 b a^3 b a^3 b)^+$$

On déduit à partir de la phrase

$$P = (c^3 d)^9 (c^2 d c^3 d c^2 d)^4 c^2 d (a^3 b a^2 b a^3 b a^3 b) a^3 b$$

l'expression régulière

$$T V c^2 d W a^3 b$$

Les caractéristiques de ces expressions régulières permettent de les représenter par d'autres formalismes que celui des grammaires régulières. En effet le formalisme fourni par les automates offre deux avantages:

- sa correspondance directe avec les grammaires régulières
- sa représentation graphique directement reliée à ces grammaires

I.3.1.3. - Correspondance grammaire régulière-automate fini

On rappelle qu'un automate fini est un quintuplet

$$A = (X, Y, f, p_0, F)$$

où x est l'alphabet

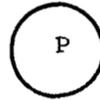
Y est l'ensemble fini des états

$p_0 \in Y$ est l'état initial

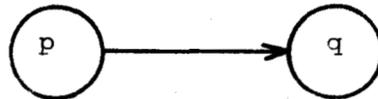
$F \subset Y$ est l'ensemble des états finaux

$f : Y \times X \rightarrow Y$ est la fonction de transition

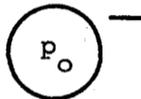
Graphiquement, on peut symboliser un état p par :



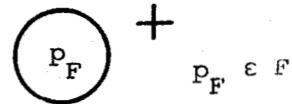
et la transition $q = f(p, a)$ par :



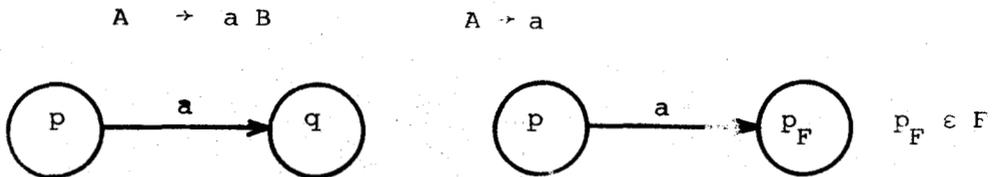
Pour la circonstance, l'état initial p_0 est représentée par :



et les états finaux par :



La correspondance entre grammaire régulière et automate fini est directe : l'état initial de l'automate correspond à l'axiome de la grammaire; les deux alphabets sont identiques; les règles de production correspondent à la fonction de transition de la façon suivante :



I.3.1.4. - Construction d'un automate fini à partir des expressions régulières

Dans un premier temps, on s'attachera à construire, à partir des expressions régulières déduites de l'échantillon, un automate fini non-déterministe.

On rappelle qu'un automate fini non-déterministe est un quintriplet :

$$A_N = (X_N, Y_N, g, P_N, F_N)$$

où X_N est l'alphabet

Y_N l'ensemble fini des états

g la fonction de transition : $Y_N \times X_N \rightarrow \mathcal{P}(Y_N)$

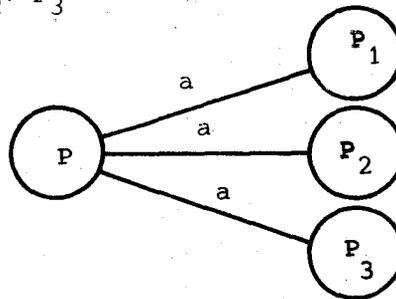
P_N l'ensemble des états initiaux

F_N l'ensemble des états finaux.

On constate par conséquent que dans un automate fini non-déterministe, à tout couple $(a, p) \in Y_N \times X$, on peut faire correspondre par la transition un ensemble fini d'états.

Alors, graphiquement on symbolise la relation par :

$$g (p, a) = \{ p_1, p_2, p_3 \}$$



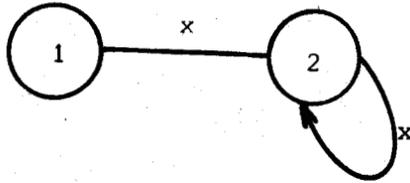
I.3.1.4.1. - Caractéristiques du graphe de l'automate

=====

à construire

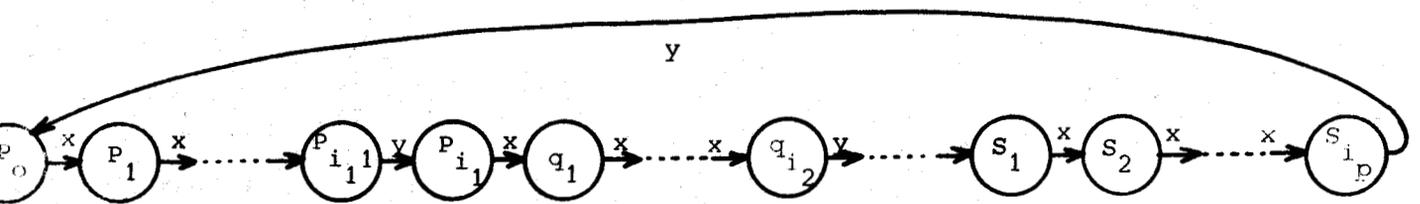
=====

Au vu de la structure des phrases, qui interprètent les formes, le graphe de l'automate à construire est composé exclusivement de sous graphes suivants :



pour engendrer les expressions régulières sous la forme :

$$f_1 = x^+$$



pour engendrer les expressions régulières sous la forme :

$$f_2 = (x^{i_1} y x^{i_2} y \dots x^{i_p} y)^+$$

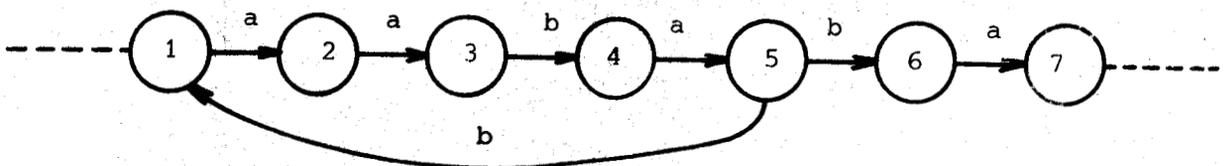
L'apparition d'états non-déterministes est provoquée par les sous-phrases sous la forme :

$$K = (x^{i_1} y x^{i_2} y \dots x^{i_p} y)^+ x^i$$

Donnons à titre d'exemple l'expression régulière :

$$K = (a^2 b a b)^+ a$$

graphiquement, on a :



à partir de l'état '5', il y a deux transitions possibles '1' et '6' symbolisées par deux arcs correspondant à la même lettre 'b'.

I.3.1.4.2. - Procédure de construction de l'automate
=====

à partir des expressions régulières
=====

Parmi toutes les méthodes de synthèse d'automates finis rencontrées dans la littérature /13/, /14/, /15/ la méthode de BRZOZOWSKI est celle qui, pour notre cas particulier pratique, présente le plus d'intérêt.

Le principe de cette méthode est simple : étant donnée une expression régulière, on va chercher quelle partie reste à faire pour que cette expression régulière soit réalisée. Si on trouve qu'il ne reste que la phrase de longueur nulle - notée λ - on aura réalisé l'expression régulière en question. Mais avant d'arriver au corps de la méthode, exposons quelques définitions.

DEFINITIONS

On appelle dérivée d'une expression régulière R par rapport à une lettre x, une expression régulière, notée $D_x(R)$ et possédant les propriétés suivantes :

- $D_x(x) = \lambda$
- $D_x(y) = \emptyset$ si $y \neq x$, $y = \lambda$ ou $y = \circ$,
- $D_x(R_1 + R_2) = D_x(R_1) + D_x(R_2)$
- $D_x(R_1 R_2) = D_x(R_1) R_2 + \delta(R_1) D_x(R_2)$

avec

$$\delta(R_1) = \lambda \text{ si } \lambda \in R_1$$

$$\delta(R_1) = \emptyset \text{ si } \lambda \notin R_1$$

$$- D_{xy}(R) = D_y [D_x(R)]$$

$$- D_\lambda(R) = R$$

Exemples

$$R = a^2 b + c d + a^+$$

$$D_a (R) = D_a (a^2 b) + D_a (cd) + D_a (a^+)$$

$$= D_a (a) a b + \emptyset + D_a (a) a^*$$

$$= a b + a^*$$

$$D_{aa} (R) = D_a (a b + a^*)$$

$$= D_a (a b) + D_a (a^*)$$

$$= b + a^*$$

$$D_{aaa} (R) = a^*$$

$$D_{aab} (R) = D_b (b + a^*)$$

$$= \lambda$$

ALGORITHME

Soit l'alphabet X

$$X = \{ x_1, x_2, \dots, x_n \}$$

On choisit de représenter l'automate à construire par sa table de transition T A B. La transition de l'état p_i à l'état p_j est représentée par T A B (i, j). Cette représentation convient parfaitement à notre type d'automate.

L'algorithme recherche les états de l'automate à partir de l'expression régulière correspondante - notée e - en procédant de la façon suivante :

- . il prend comme état initial l'état p_0 caractérisé par :

$$D_\lambda(e) = e$$

- . si $D_{x_i}(e) \neq D_\lambda(e)$, il crée un nouvel état p_i tel que :

$$p_i = f(p_0, x_i)$$

f étant la fonction de transition

. cette opération est répétée pour toutes les lettres de l'alphabet ($i = 1, 2, \dots, n$) si $D_{x_j}(e)$ est différente de toutes les dérivées $D_{x_k}(e)$ précédemment calculées, on introduit un nouvel état p_j

. Il calcul ensuite toutes les dérivées $D_{x_{ij}}(e)$ et de la même façon, associe à chacune un état p_{ij} .

. ce processus est arrêté lorsqu'il ne trouve plus de nouvelles dérivées.

Exemple

On va expliquer le procédé sur un exemple

Soit l'alphabet : $X = \{ a, b, c, d \}$

et l'expression régulière

$$e = c^+ + (a^2 b a b)^+ a b + (c^2 d)^+$$

On a : $D_\lambda (e) = e$

création de l'état p_0

$$\begin{aligned} D_a (e) &= D_a [(a^2 b a b)^* a b] \\ &= (a b)^2 (a^2 b a b)^* a b \end{aligned}$$

création de l'état p_1

$$\begin{aligned} D_c (e) &= D_c (c^+) + D_c [(c^2 d)^+] \\ &= c^* + c d (c^2 d)^* \end{aligned}$$

création de l'état p_2

On obtient une première table de transition

	p_0	p_1	p_2
p_0		a	c
p_1			
p_2			

$$\begin{aligned} D_{aa} (e) &= D_a [(a b)^2 (a^2 b a b)^* a b] \\ &= b a b (a^2 b a b)^* a b \end{aligned}$$

création de l'état p_3

$$\begin{aligned} D_{cc} (e) &= D_c (c^* + c d (c^2 d)^*) \\ &= c^* + d (c^2 d)^* \end{aligned}$$

création de l'état p_4

On obtient une seconde table de transition

	p_0	p_1	p_2	p_3	p_4
p_0		a	c		
p_1				a	
p_2					c
p_3					
p_4					

$$D_{aab}(e) = D_b [b a b (a^2 b a b)^* a b]$$

$$= a b (a^2 b a b)^* a b$$

création de l'état p_5

$$D_{ccc}(e) = D_c [c^* + d (c^2 d)^*]$$

$$= c^*$$

création de l'état p_6

$$D_{ccd}(e) = D_d [c^* + d (c^2 d)^*]$$

$$= (c^2 d)^*$$

création de l'état p_7



On obtient une troisième table de transition

	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7
p_0		a	c					
p_1				a				
p_2					c			
p_3						b		
p_4							c	d
p_5								
p_6								
p_7								

$$D_{aaba}(e) = D_a [a b (a^2 b a b)^* a b]$$

$$= b (a^2 b a b)^* a b$$

création de l'état p_8

$$D_{cccc}(e) = D_C (C^*)$$

$$= C^*$$

$$= D_{cc}(e)$$



le nouvel état est confondu avec p_6

$$D_{ccdc}(e) = D_c [(C^2 d)^*]$$

$$= c d (C^2 d)^*$$

création de l'état p_9

On obtient une quatrième table de transition

	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9
p_0		a	c							
p_1				a						
p_2					c					
p_3						b				
p_4							c	d		
p_5									a	
p_6							c			
p_7										c
p_8										
p_9										

$$\begin{aligned} D_{aabab}(e) &= D_b [b (a^2 b a b)^* a b] \\ &= (a^2 b a b)^* a b \end{aligned}$$

création de l'état p_{10}

$$\begin{aligned} D_{ccdcc}(e) &= D_c [c d (c^2 d)^*] \\ &= d (c^2 d)^* \end{aligned}$$

création de l'état p_{11}

$$\begin{aligned} D_{cedcc}(e) &= D_d [d (c^2 d)^*] \\ &= (c^2 d)^* \\ &= D_{ccd}(e) \end{aligned}$$

le nouvel état est confondu avec p_7

$$\begin{aligned} D_{aababa}(e) &= D_a [(a^2 b a b)^* a b] \\ &= (a b)^2 (a^2 b a b)^* + b \end{aligned}$$

création de l'état p_{12}

$$\begin{aligned} D_{aababaa}(e) &= b a b (a^2 b a b)^* \\ &= D_{aa}(e) \end{aligned}$$

le nouvel état est confondu avec p_3

D_{aababab} (e) = λ

arrêt du processus.

D'où l'on tire la table de transition finale donnée en figure 1.

Le graphe de l'automate construit est alors :

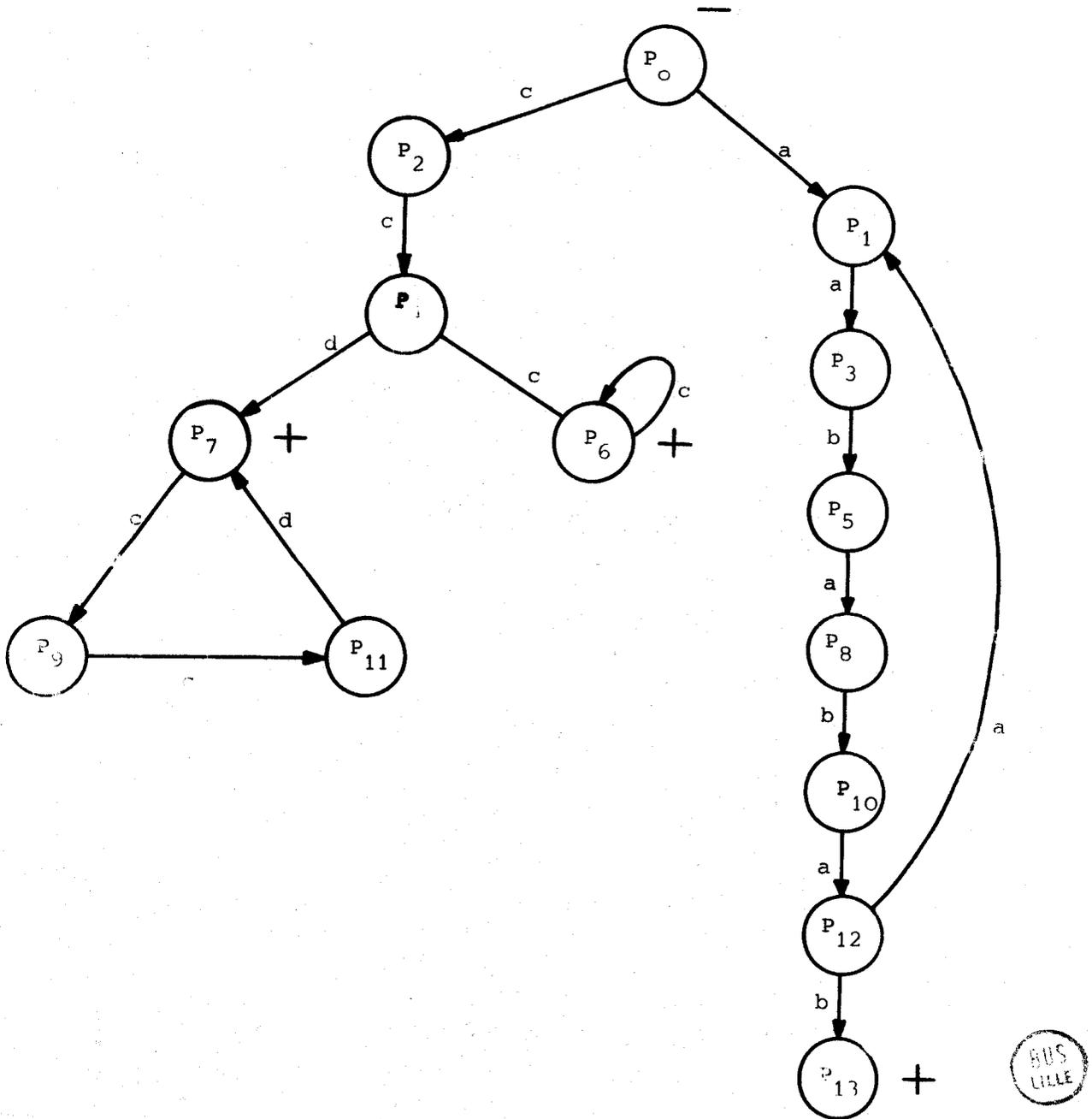


Figure 2 : Graphe de l'automate

	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	p_{13}
p_0		a	c											
p_1				a										
p_2					c									
p_3						b								
p_4							c	d						
p_5									a					
p_6							c							
p_7										c				
p_8											b			
p_9												c		
p_{10}													a	
p_{11}								d						
p_{12}		a												b
p_{13}														

Figure 1 : Table de transition de l'automate construit à partir de l'expression régulière e



II - PHASE DE RECONNAISSANCE

Durant la phase d'apprentissage, à chaque forme F_i ($i=1,2,\dots,k$) a été associée une grammaire régulière G_i ($i=1,2,\dots,k$) : son modèle structurel. Cette grammaire, inférée à partir de l'échantillon de phrases qui interprètent la forme, est représentée par l'automate correspondant.

La phase suivante est essentiellement une phase de décision qui se propose de résoudre le problème suivant : soit x une phrase interprétant une forme F , laquelle des grammaires G_i ($i=1,2,\dots,k$), inférées durant la phase d'apprentissage, génère x ou encore lequel des automates A_i ($i=1,2,\dots,k$) correspondant à ces grammaires accepte la phrase x .

On a choisi de résoudre ce problème en passant par les étapes suivantes :

- . *Déterminisation de l'automate de reconnaissance*
- . *Analyse syntaxique de la phrase inconnue par rapport à l'automate de reconnaissance déterminisé.*

II.1 - Déterminisation de l'automate de reconnaissance

Théoriquement, il est toujours possible de construire à partir d'un automate fini non-déterministe un automate fini déterministe équivalent (c'est-à-dire qui accepte le même langage) /12/.

Soit A_N un automate fini non-déterministe.

$$A_N = (X_N, Y_N, g, P_N, F_N)$$

L'automate fini déterministe équivalent A_D est défini par :

$$A_D = (X, Y, f, p_0, F)$$

$$\text{où } X = X_N$$

$$Y = P (Y_N)$$

la fonction de transition f est définie pour $R \in P (Y_N)$: $f (R, a) =$

$$= U g (q, a)$$

$$q \in R$$

$$p_0 \in P (Y_N)$$

l'ensemble F des états finaux est l'ensemble des parties S de Y telles que :

$$S \cap F_N \neq \emptyset$$

La littérature fournit divers algorithmes pratiques /11/ pour reconnaître une phrase à partir de la table des transitions d'un automate non-déterministe. Mais, ces méthodes, si elles rendent inutile la détermination, présentent l'inconvénient d'explorer un arbre avec de multiples possibilités de retour en arrière.

En général, la détermination apparaît volumineuse et compliquée puisqu'elle introduit 2^n états (n étant le nombre d'états de l'automate).

La mise en mémoire d'une table de transition à 2^n états rendrait prohibitif les différentes méthodes de détermination si la simplicité des expressions régulières obtenues ne limitait pas le nombre des nouveaux états (ce fait constaté par expérience ne se justifie pas rigoureusement).

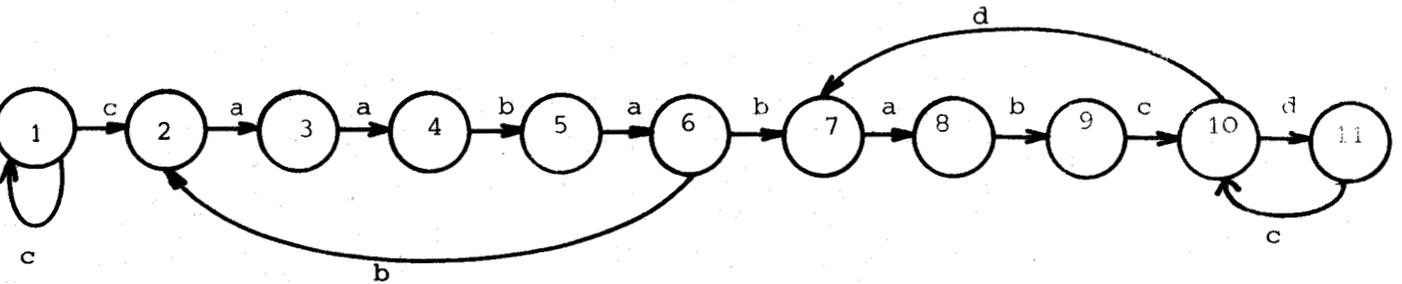
II.1.1. - Procédure de détermination de l'automate

On va exposer au moyen d'un exemple une méthode de détermination d'un automate. Le but fixé par cette méthode est la construction d'un

automate fini déterministe. L'algorithme de déterminisation se déroule comme suit : soit l'expression régulière :

$$c^+ (a^2 b a b)^+ (a b (c d)^+)^+$$

On part de l'automate correspondant

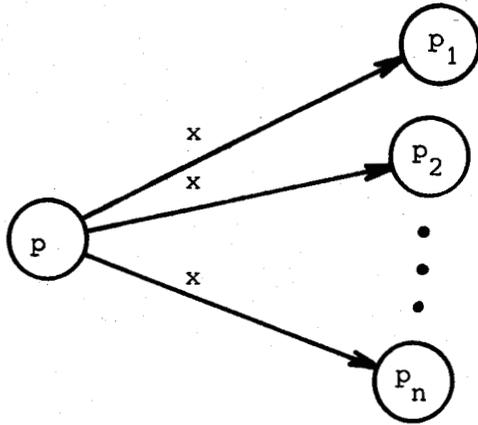


dont la table de transition est

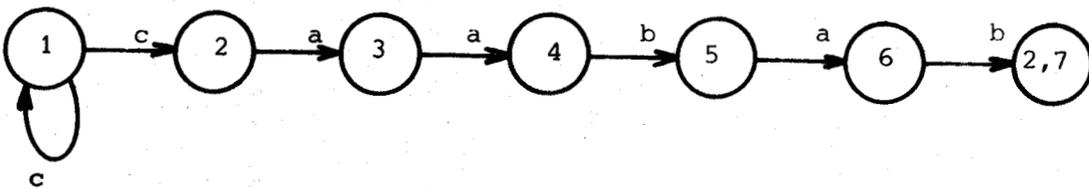
	1	2	3	4	5	6	7	8	9	10	11
1	c	c									
2			a								
3				a							
4					b						
5						a					
6		b					b				
7								a			
8									b		
9										c	
10							d				d
11										c	



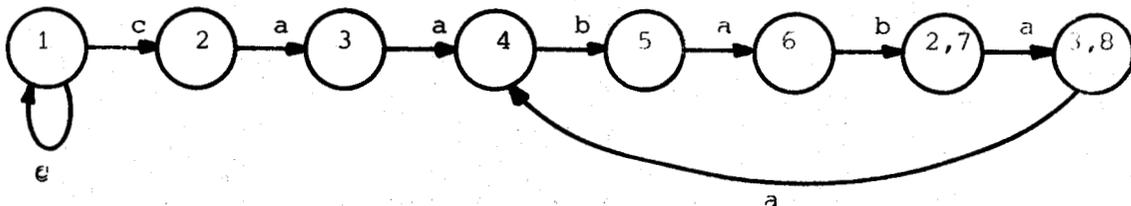
En partant de l'état 1, on regroupe en un seul état tous les états p_1, p_2, \dots, p_n tels que

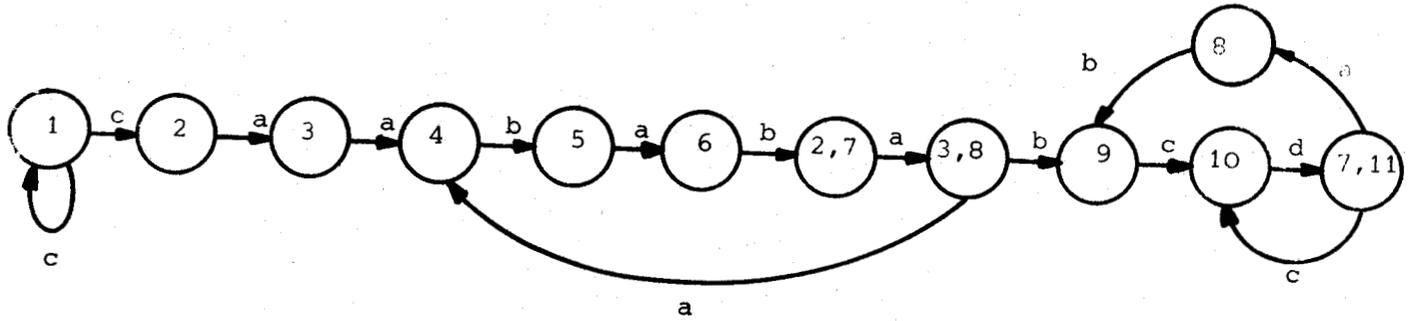


On introduit ici l'état $\{ 2,7 \}$ et on obtient :



En appliquant progressivement la règle précédente, on a successivement





La table des transitions de l'automate fini déterministe équivalent est :

	1	2	3	4	5	6	7'	8'	9	10	11	12
1	c	c										
2			a									
3				a								
4					b							
5						a						
6							b					
7'								a				
8'				a					b			
9										c		
10											d	
11										c		a
12									b			



Un automate non-déterministe accepte un langage de façon analogue à un automate déterministe. Cependant l'intérêt de la détermination est que l'analyse syntaxique d'une phrase par rapport à un automate non-déterministe est plus complexe que si l'automate est déterministe, car il faut en effet conserver en mémoire les alternatives caractéristiques des transitions multiples.

II.2 - Analyse syntaxique de phrases interprétant les formes à reconnaître

L'analyse syntaxique d'une phrase x par rapport à un automate A , est un procédé algorithmique permettant de décider en un temps fini si $x \in L(A)$ ou si $x \notin L(A)$, où $L(A)$ est le langage accepté par l'automate A .

Soit $x = a_1 a_2 \dots a_n$ une phrase interprétant une forme inconnue et $A = (X, Y, f, p, F)$ un automate de reconnaissance; x est dite acceptée ou reconnue par A lorsque partant de l'état initial p_0 , on peut trouver dans le graphe de l'automate une suite d'arcs portant successivement les lettres a_1, a_2, \dots, a_n et menant à un état final.

Sur le chemin menant à l'état final, on rencontre la suite d'états p_0, p_1, \dots, p_n tels que :

$$p_1 = f(p_0, a_1)$$

$$p_2 = f(p_1, a_2)$$

"

"

$$p_n = f(p_{n-1}, a_n)$$

si $p_n \in F$ alors l'automate accepte x et on écrit :

$$x \in L(A)$$

si $p_n \notin F$ alors x n'est pas acceptée par l'automate :

$$x \notin L(A)$$

II.2.1. - *Algorithme d'analyse syntaxique des phrases inconnues*

On se trouve dans un cas où les automates de reconnaissance ne sont pas complètement spécifiés, c'est-à-dire que la fonction de transition f n'est pas définie partout $Y \times X$.

L'algorithme d'analyse syntaxique fonctionne en autant d'étapes qu'il y a de lettres dans la phrase inconnue x , et permet de décider dans trois éventualités :

- . quand on rencontre sur le chemin du graphe de l'automate un état p_i tel que $f(p_i, a_{i+1})$ n'est pas définie.
- . $p_n \in F$
- . $p_n \notin F$

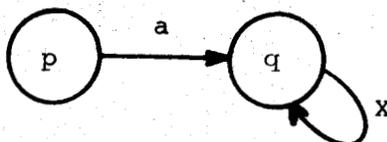
Dans la première éventualité, on considère implicitement que pour toute transition non définie $f(p, a)$ on impose :

$$f(p, a) = q$$

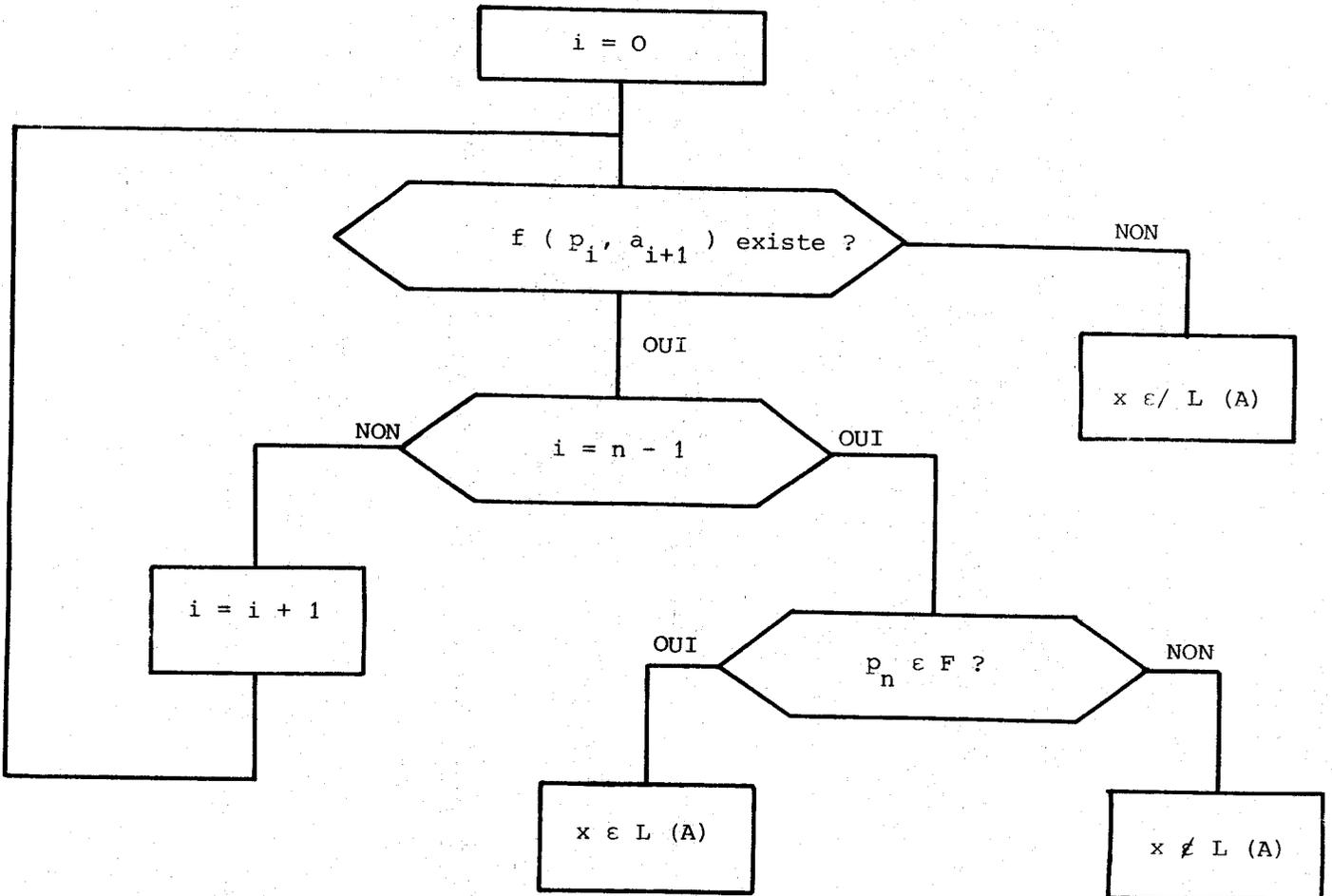
avec $\forall x \in X$ $f(p, x) = q$

et $q \notin F$

L'état q rajouté est un "état-puits" qui est représenté graphiquement par :



Pour la phrase $x = a_1 a_2 \dots a_n$, le déroulement de l'algorithme est illustré par l'organigramme suivant :



III - RESULTATS

Les figures 3, 4 et 5 montrent le contour d'une même pièce, soit p, mais dans chacune de ces trois figures, le contour est orienté distinctement par rapport au plan de l'image o x y.

La pièce p est alors interprétée par trois phases du langage L.A.P. F_1 , F_2 et F_3 :

$$F_1 = [(-c)^3 - b [(-c)^4 - b]^2]^4 (a-d)^{12} (a^4 b a^3 b)^8 (-ad)^{16} (ab)^{16} (-ad)^{88} \\ (-a-b)^{32} (a-d)^{20}$$

qui interprète le contour de la figure 3

$$F_2 = (a^2 - d a^3 - d a^2 - d)^4 a^2 d [(a^2 b)^2 ab]^4 c^{16} (a^2 b)^4 ab [(c^2 d)^2 cd]^4 \\ [(-a^2 - b)^2 - a - b]^{20} [[(-c)^2 - d]^2 (-c-d)]^8 [(a^2 b)^2 ab a^2 b]^4 ab$$

qui interprète le contour de la figure 4

$$F_3 = (c d c^2 d c d)^8 (-a)^{32} (-c^2 - b)^{28} (-a)^{20} (-c)^{20} a^{108} c^{44} (-a)^{28}$$

qui interprète le contour de la figure 5.

L'automate de reconnaissance (figure 5) est l'automate fini déterministe construit à partir de l'expression régulière e :

$$e = e_1 + e_2 + e_3$$

où

$$e_1 = [(-c)^3 - b [(-c)^4 - b]^2]^+ (a-d)^+ (a^4 b a^3 b)^+ (-ad)^+ (ab)^+ (-ad)^+ (-a-b)^+ (a-d)^+$$

$$e_2 = (a^2-d a^3-d a^3-d)^+ a^2 d \left[(a^2 b)^2 ab \right]^+ c^+ (a^2 b)^+ ab \left[(c^2 d)^2 cd \right]^+ \\ \left[(-a^2-b)^2 -a-b \right]^+ \left[[(-c)^2-d]^2 (-c-d) \right]^+ (a^2 b a b a^2 b)^+ ab$$

$$e_3 = (c d c^2 d c d)^+ (-a)^+ (-c^2-d)^+ (-a)^+ (-c)^+ a^+ c^+ (-a)^+$$

CONCLUSION

Nous avons vu que, pour automatiser la reconnaissance de pièces, le problème le plus délicat à résoudre est celui de la modélisation de ces pièces. Dans la phase d'apprentissage, on s'est attaché à inférer un modèle structurel régulier, sous forme de grammaire régulière, à partir des phrases interprétant la forme de la pièce à reconnaître.

La méthode de reconnaissance exposée bien qu'elle ne soit pas générale, car on a été amené à induire de fortes contraintes aussi bien sur la structure des phrases (on a recherché des primitives sous forme de segments de droites) que sur la forme du modèle régulier inféré (recours à l "heuristique u v^k w ") offre par rapport aux méthodes classiques de reconnaissance fondées sur l'approche statistique deux avantages :

- efficacité : car elle s'appuie sur une analyse structurelle de l'objet en termes de ses composants (primitives, sous-formes), de leurs propriétés, et de leurs interrelations, alors qu'il est clair que l'utilisation de l'approche statistique a pour principal inconvénient la négligence de l'information structurelle qui la rend inutile ou peu efficace pour des formes riches en informations structurelles.

- souplesse : en effet, l'introduction d'un nouvel échantillon de pièces à reconnaître entraînera généralement dans le cas d'utilisation de l'approche statistique l'introduction de nouveaux paramètres structuraux donc une modification des étapes de paramétrisation et d'apprentissage, alors qu'il suffit, dans le cadre de la nouvelle méthode de reconnaissance, d'inférer un modèle structurel (grammaire, automate) pour chacune des pièces nouvellement introduite.

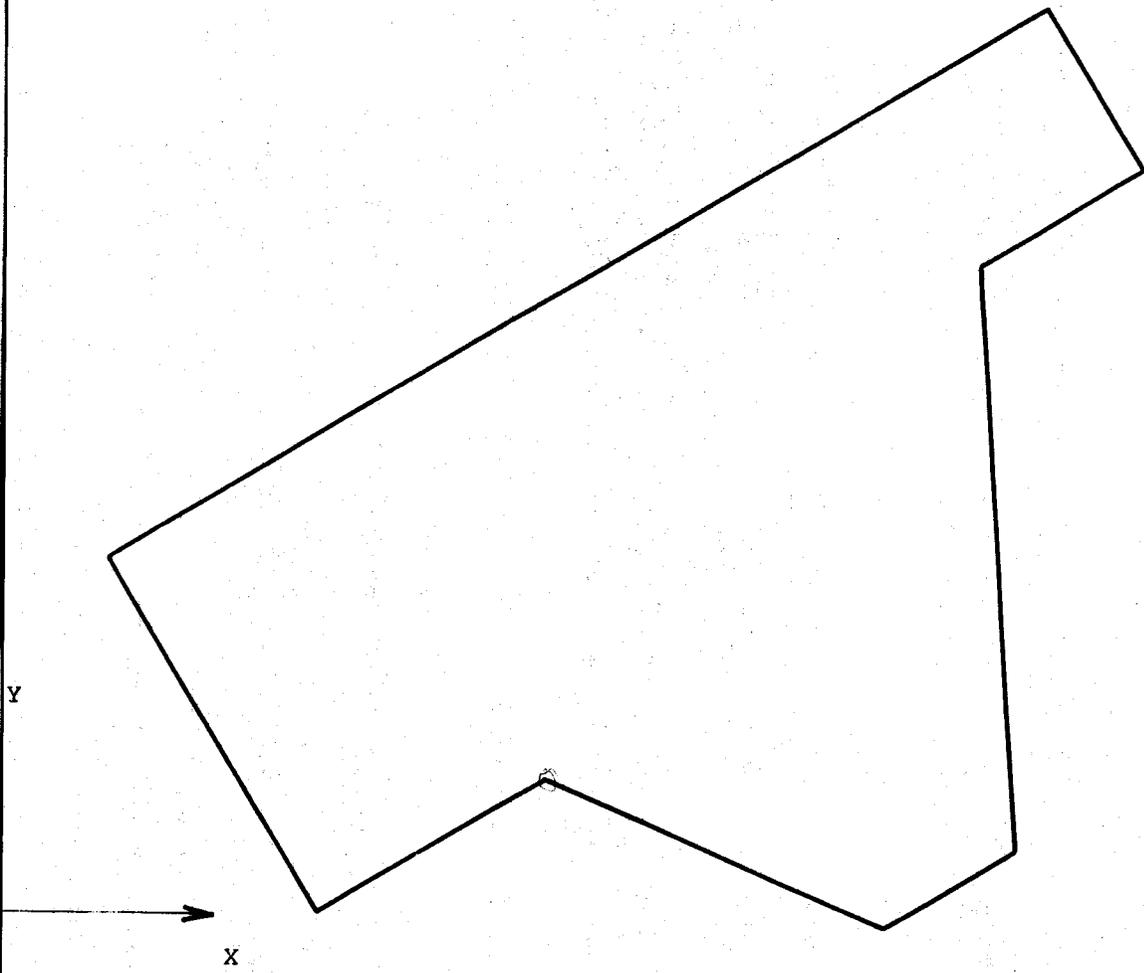


Figure 3



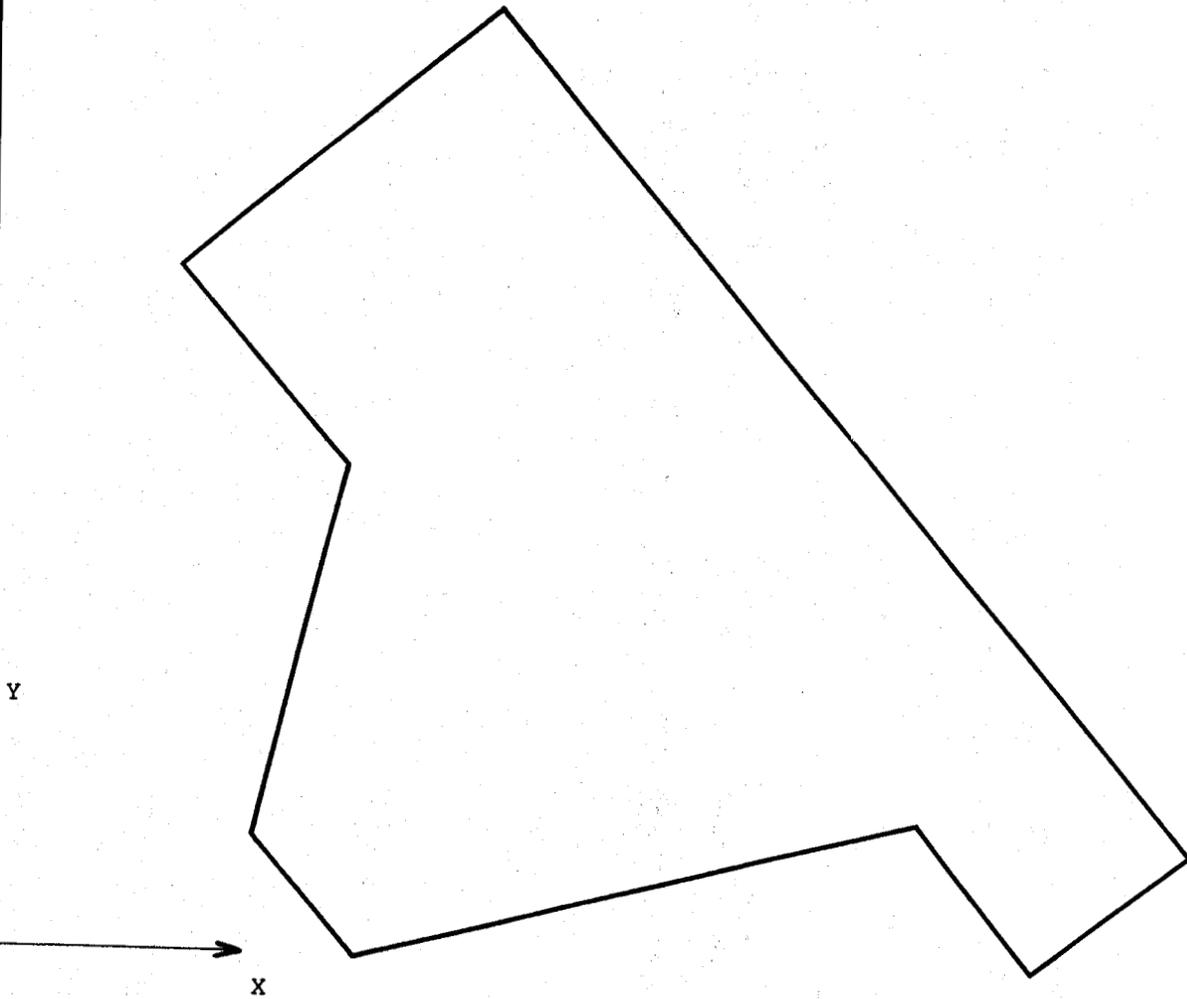


Figure 4

BUS
LILLE

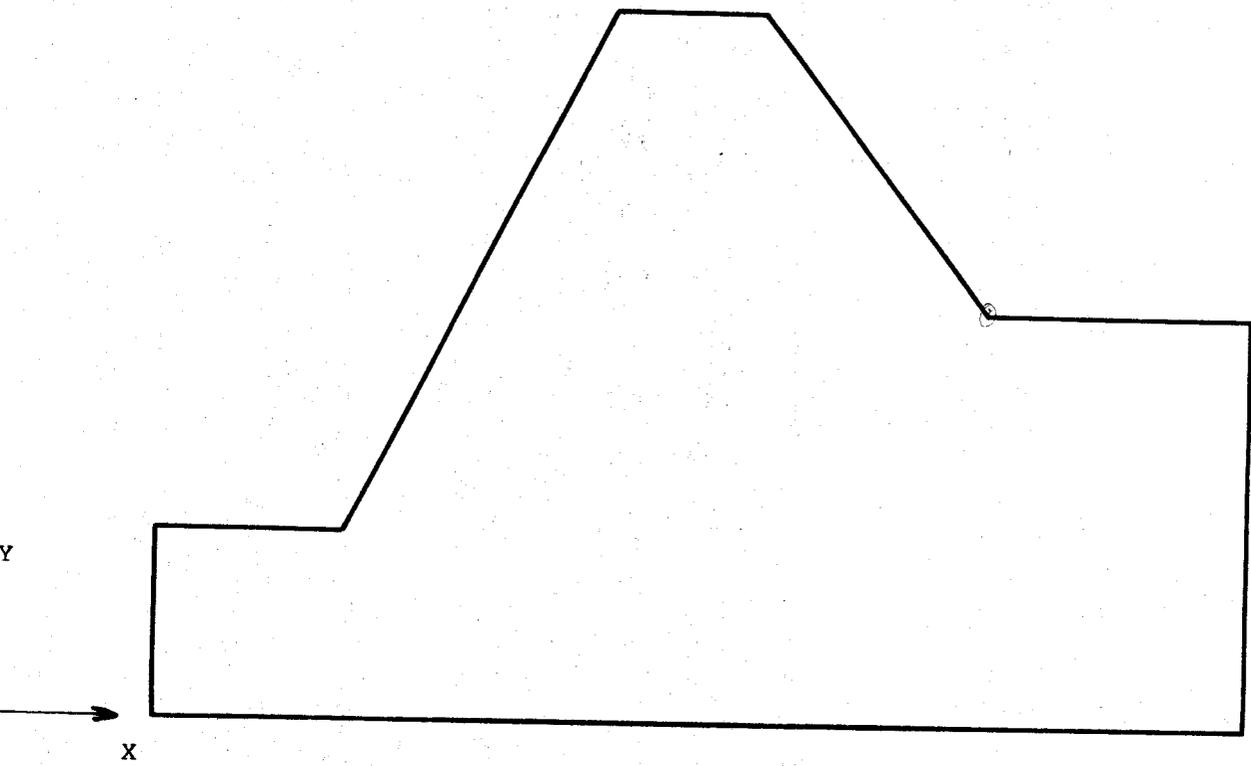


Figure 5



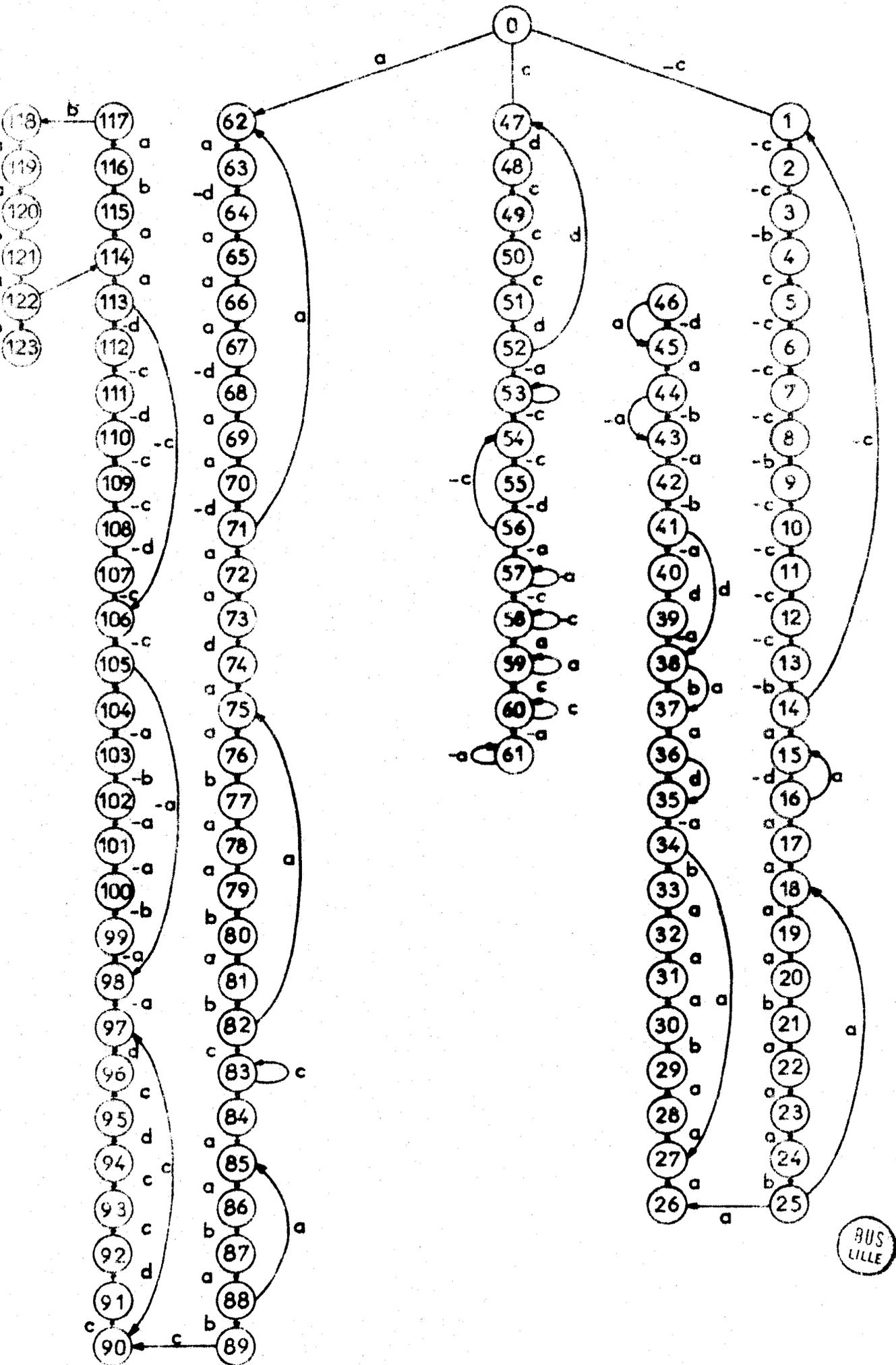


Figure 5 : Automate de reconnaissance

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

- /1/ BOHEC, BOASSON
" Langages et Automates "
tome 3, E.N.S.T., 1975.
- /2/ K. S. FU
" Syntactic methods in pattern recognition "
Academic Press, 1974.
- /3/ K. S. FU, TL BOOTH
" Grammatical inference : introduction and survey part I and II "
I.E.E.E., T.S.M.C., January and july 75.
- /4/ A.W. BIERMANN, J.A. FELDMAN
" A survey of results in grammatical inference "
Int. Conf. Frontiers of P.R.
Honolulu - Hawaii, january 71.
- /5/ S. GRESPI - REGHIZZI
" An effective model for grammar inference "
IFIP Congr.Yugoslavia, August 71.
- /6/ T. G. EVANS
" Grammatical inference techniques in pattern analysis "
Software Engineering (TOU ed.)
vol.2, Ac. Press, 71
- /7/ J.E. HOPCROFT, J.D. ULLMAN
" Forma languages an their relation to automata "
Addition - Wesley, 69.
- /8/ L. MICLET
" Inférence de grammaires régulières "
Thèse Docteur-Ingénieur, E.N.S.T., Décembre 79.

- /9/ H. DECKER, G. JOCHUM
" Unsupervised inference of regular expressions "
Laboratoire de Théorie des Systèmes, E.N.S.T., 76
- /10/ R.J. SOLOMONOFF
" A new method for discovering the grammars of phrase
structure language "
Information processing (Int.Conf. Proc.)
UNESCO Publications, 1959.
- /11/ AHO, HOPCROFT, ULLMAN
" The design and analysis of Computer algorithms "
Addition - wesley publishing compagny, 75.
- /12/ M. GROS, A. LENTIN
" Notions sur les grammaires formelles "
Editions Gauthier - Villars - Paris 67
- /13/ J.A. BRZOZOWSKI
" A survey of regular expressions and their applications "
I.R.E. Transactions on Electronic Computers, vol, E.C 9, pp 39-47 Mars 60
- /14/ J.A. BRZOZOWKI
" Canonical regular expressions and minimal state graphs define
events "
Proceedings of the symposium on mathematical theory of automata,
Polytechnic press of the polytechnic institute of Brooklyn, New-York 63
- /15/ Y.M. GLOUSHKOV
" Problems in the synthesis of digital automata "
Translated proceedings, relay systems and finite automata,
Burroughs corporation, 64.

CHAPITRE IV

AUTOMATISATION D'UN TYPE D'ASSEMBLAGE

Considéré, sans conteste, comme le domaine d'application le plus prometteur en robotique industrielle, l'assemblage automatique fait l'objet, depuis quelques années d'un intérêt sans cesse croissant. Cet intérêt est motivé par deux principaux facteurs :

- l'un économique : il s'agit d'automatiser des tâches de montage d'assemblages mécaniques en petites et moyennes séries qui font appel à une main d'oeuvre importante.
- l'autre humain qui verra le robot remplacer l'homme pour l'exécution de tâches très pénibles ou qui se déroulent dans un environnement très hostile : manutention de produits radioactifs /1/, ou de pièces à la sortie d'une fonderie /2/...

Les travaux touchant à l'assemblage automatique deviennent de plus en plus importants, au point que tout récemment, s'est tenu une conférence internationale /3/, la première du genre, consacrée exclusivement au thème de l'assemblage automatique. Elle est venue confronter de nombreux projets et témoigner, à travers les récentes réalisations, du progrès indéniable constaté en ce domaine.

Le projet auquel vient apporter sa contribution le présent mémoire, entreprend d'automatiser un type particulier d'assemblage à savoir l'emboîtement de pièces mécaniques; la machine appelée par la suite télémanipulateur qui réalise l'assemblage peut être décrite à l'aide de quatre ensembles interconnectés (figure 1).

- . un manipulateur qu'on désigne généralement comme le télémanipulateur lui-même pour la manutention des pièces
- . un calculateur chargé de générer les signaux de commandes du télémanipulateur
- . l'environnement qui se réduit à l'espace atteignable par l'organe terminal du manipulateur (pièces, plan de reconnaissance, plan d'assemblage)
- . la tâche d'assemblage définie comme la modification apportée à l'environnement et au télémanipulateur entre son début et sa fin d'exécution.

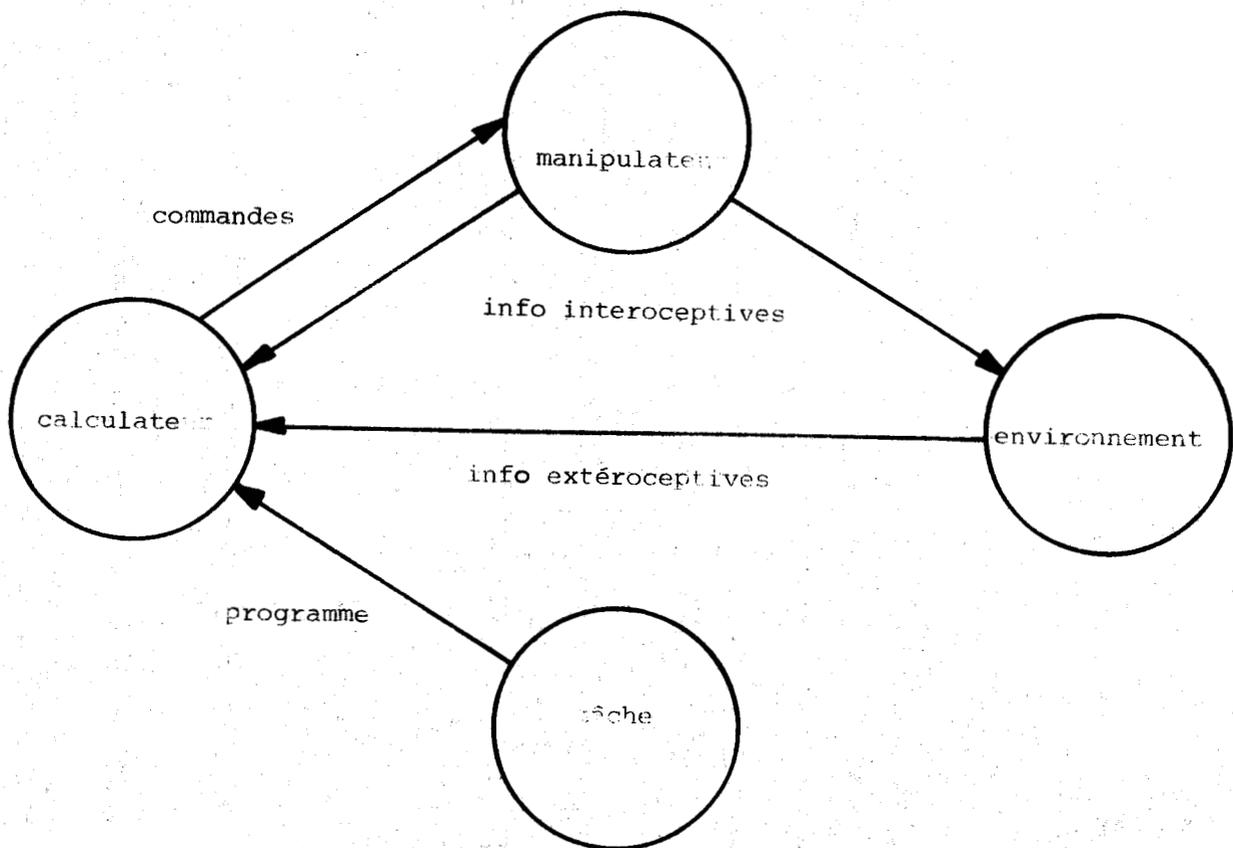


Figure 1 : Représentation du télémanipulateur

Pour élaborer les commandes du télémanipulateur qui font progresser celui-ci vers l'exécution correcte de la tâche d'assemblage, le calculateur possède :

- . un modèle de l'environnement et les informations extéroceptives concernant l'état actuel de cet environnement.
- . un modèle du manipulateur et les informations proprioceptives sur l'état actuel du manipulateur.
- . le programme relatif à la tâche à exécuter.

L'originalité de la procédure d'assemblage que nous allons écrire consiste à utiliser deux types d'informations extéroceptives :

- des informations dites primaires indispensables à l'exécution des tâches de positionnement et de saisie des pièces
- et des données plus élaborées relatives à l'emboîtement des pièces.

I - DEFINITION DE LA TACHE D'ASSEMBLAGE

Soient deux pièces P_m et P_n , reposant sur un plan dit plan de reconnaissance, la tâche proposée consiste à :

- * saisir l'une de ces deux pièces, soit P_m et la poser précisément de manière correcte en un endroit prévu à l'avance sur un plan dit plan d'assemblage.
- * saisir l'autre pièce P_n , la transférer sur le plan d'assemblage, en la positionnant et l'orientant de manière telle qu'elle puisse venir s'emboîter partiellement ou totalement dans la pièce P_m (figure 2).

Pour décrire la procédure d'assemblage, on procède en trois étapes :

- recherche des sous-formes emboîtables
- détermination des paramètres d'assemblage
- et élaboration du programme de commandes du manipulateur qui exécute la tâche d'assemblage.

II - RECHERCHE DES SOUS-FORMES EMBOITABLES

Soient F_m et F_n les phrases qui interprètent respectivement les formes des pièces P_m et P_n . Elles se présentent sous les formes suivantes :

$$F_m = f_{m_1} f_{m_2} \dots f_{m_p}$$

$$F_n = f_{n_1} f_{n_2} \dots f_{n_q}$$

où f_{m_i} et f_{n_j} sont de la forme :

$$(x^{i_1} y^{i_2} \dots x^{i_k} y)^r x^{i_1} y \dots x^{i_1} y \quad l < k$$

avec $x \in \{ a, -a, c, -c \}$

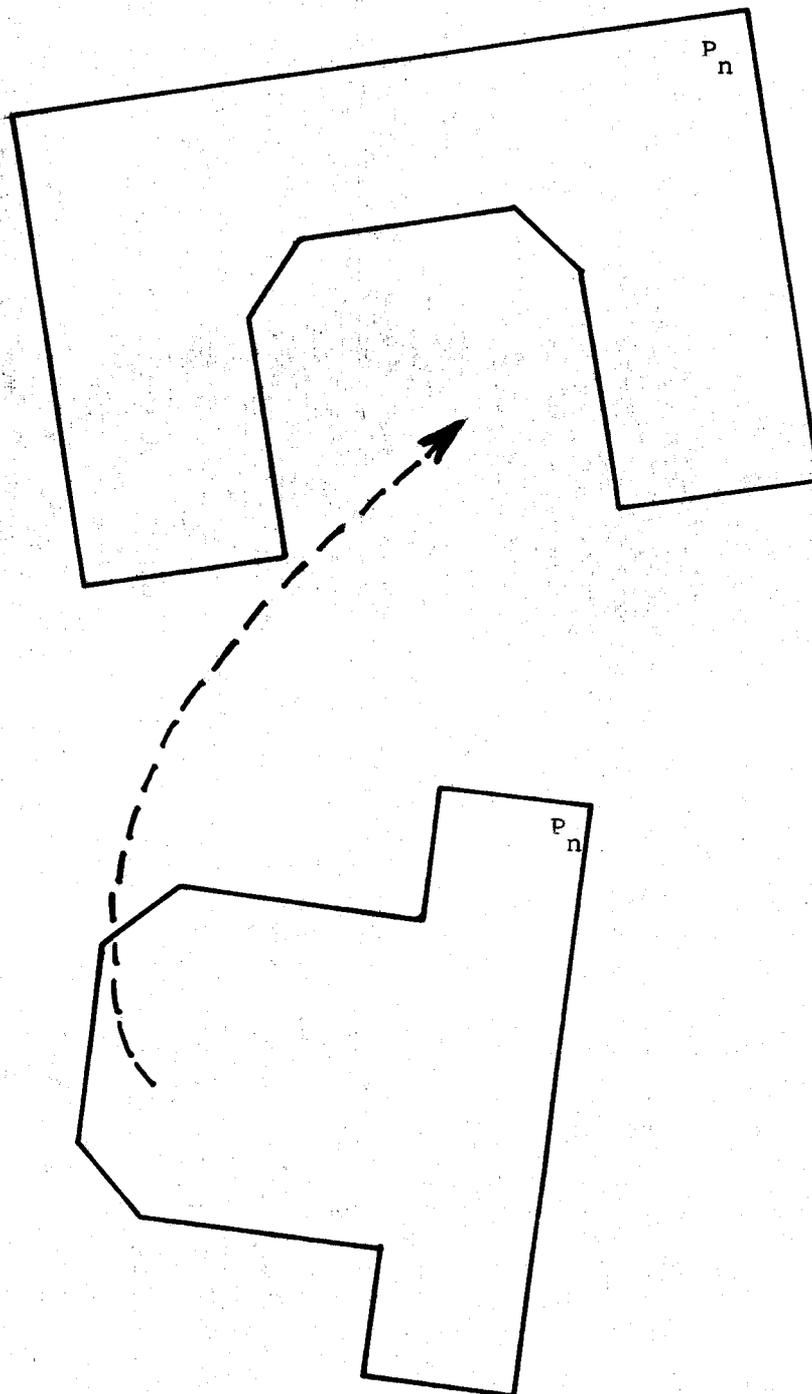


Figure 2 : Principe d'emboîtement des pièces



et $y \in \{ b, -b, d, -d \}$

La méthode de recherche des sous-formes emboîtables est fondée sur l'établissement de conditions nécessaires et suffisantes que doivent satisfaire les sous-phrases interprétant ces sous-formes. Mais avant d'arriver au corps de la méthode, exposons quelques définitions.

II.1 - Définitions

II.1.1. - Opération d'inversion de phrases du langage L.A.P.

Soit F_i une phrase du langage L.A.P.,

$$F_i = f_{i_1} f_{i_2} \dots f_{i_v}$$

où f_{i_j} ($j=1, \dots, v$) se met sous la forme :

$$(x^{j_1}_y x^{j_2}_y \dots x^{j_p}_y) x^{j_1}_y \dots x^{j_k}_y \quad k < p$$

où $(x)^{n_j}$

L'opération inversion, notée \sim , est définie comme suit :

$$(\sim F_i) = (\sim f_{i_v}) (\sim f_{i_{v-1}}) \dots (\sim f_{i_1})$$

$$\text{où } (\sim f_{i_j}) = (-x)^{j_k} (-y) \dots (-x)^{j_1} (-y) \left[(-x)^{j_p} (-y) \dots (-x)^{j_{p-1}} \right]$$

$$(-y) \dots (-x)^{j_1} (-y) \left]^{n_j}$$

$$\text{ou } (\sim f_{i_j}) = (-x)^{n_j}$$

selon les cas.

II.1.2. - Argument d'une primitive

Soit la primitive interprétée par la phrase :

$$(x^{i_1} y^{i_2} \dots x^{i_q} y) x^{i_1} y^{i_2} \dots x^{i_1} y \quad 1 < q$$

On appelle argument de cette primitive, la quantité A_r fonction de x, y et des exposants i_1, i_2, \dots, i_q et définie au moyen de la table suivante :

$x \backslash y$	b	$-b$	d	$-d$	Λ
a	R			$-R$	O
$-a$		$R-\pi$	$-R+\pi$		π
c	P		$-P+\pi$		$\pi/2$
$-c$		$P-\pi$		$-P$	$-\pi/2$

$$\text{où } P = \tan^{-1} \left(\frac{q}{\sum_{j=1}^q i_j} \right)$$

$$R = \tan^{-1} \left(\sum_{j=1}^q i_j / q \right)$$

Λ désigne le mot vide.



II.2 - Conditions nécessaires et suffisantes d'emboîtement de deux sous-formes :

Soient deux sous-formes S_m et S_n interprétées respectivement par les phrases f_m et f_n :

$$f_m = f_{m_{i_1}} \quad f_{m_{i_2}} \quad \dots \quad f_{m_{i_p}}$$

$$f_n = f_{n_{j_1}} \quad f_{n_{j_2}} \quad \dots \quad f_{n_{j_p}}$$

A tout couple de phrases $(f_{m_{i_k}}, f_{n_{j_k}})$ $k = 1, 2, \dots, p$ on calcule une fonction notée $B(i_k, j_k)$ et définie telle que :

$$B(i_k, j_k) = \begin{cases} 1 & \text{si } A_r(f_{m_{i_k}}) - A_r(f_{n_{j_k}}) = c \\ & \text{et } |L_{i_k} - L_{j_k}| = 0 \\ 0 & \text{sinon} \end{cases}$$

$$\text{où } c = A_r(f_{m_{i_1}}) - A_r(f_{m_{j_1}})$$

L_{i_k} et L_{j_k} étant les longueurs des primitives interprétées par $f_{m_{i_k}}$ et $f_{m_{j_k}}$ respectivement.

On dira que les sous-formes S_m et S_n sont emboîtables si seulement si :

$\forall k, k = 1, 2, \dots, p$ on ait $B(i_k, j_k) = 1$

II.3 - Algorithme de recherche de sous-formes emboîtables

Soient deux pièces P_m, P_n , la recherche des sous-formes emboîtables significatives (formées d'au moins cinq primitives) procède comme suit :

- inversion de l'une des deux phrases F_n, F_m qui interprètent respectivement les formes de P_m et P_n . Soit F'_n cette phrase, on a :

$$\begin{aligned} (\sim F'_n) &= (\sim f_{n_q}) (\sim f_{n_{q-1}}) \dots (\sim f_{n_1}) \\ &= f'_{n_q} f'_{n_{q-1}} \dots f'_{n_1} \end{aligned}$$

- recherche dans les phrases F'_m et F'_n telles que :

$$\begin{aligned} F'_m &= f_{m_1} f_{m_2} \dots f_{m_q} f_{m_1} f_{m_2} \dots f_{m_{p-1}} \\ F'_n &= f'_{n_q} f'_{n_{q-1}} \dots f'_{n_1} f'_{n_2} \dots f'_{n_2} \end{aligned}$$

de sous-phrases f_m et f_n

$$f_m = f_{m_{i_1}} f_{m_{i_2}} \dots f_{m_{i_1}}$$

$$f_n = f'_{n_{j_1}} f'_{n_{j_2}} \dots f'_{n_{j_1}}$$

vérifiant les conditions suivantes :

$$\forall k \ 1 < k < l \quad B(i_k, j_k) = 1$$

Exemple :

=====

Les pièces P_m et P_n montrées en figure 2 sont représentées par les phrases F_m et F_n respectivement :

$$F_m = a^{10} c^{42} (-a)^{70} c^{30} a^{16} [(-c)^2 -d]^{14} a^{28} [c^2 b (cb)^2]^4$$

$$F_n = a^{28} (a^2 b)^{14} c^{16} (-a)^{42} (-c)^{70} a^{34} c^{10} | (-a) d (-a)^2 d (-a) d |^4$$

on a :

$$F'_m = a^{10} c^{42} (-a)^{70} c^{30} a^{16} [(-c)^2 d]^{14} a^{28} [c^2 b (cb)^2]^4$$

$$a^{10} c^{42} (-a)^{70} c^{30} a^{16} [(-c)^2 -d]^{14} a^{28}$$

$$F_n' = (a d a^2 d a d)^4 (-c)^{10} (-a)^{34} c^{70} a^{46} (-c)^{16} [(-a)^2 b]^{14}$$

$$(-c)^{28} [a d a^2 d a d]^4 (-c)^{10} (-a)^{34} c^{70} a^{46} (-c)^{16} [(-a)^2 b]^{14}$$

Les sous-formes emboîtables S_m et S_n sont interprétées respectivement par les sous-phrases f_m et f_n données par :

$$f_m = a^{16} [(-c)^2 -d]^{14} a^{28} [c^2 b (cb)^2]^4 a^{10}$$

$$f_n = (-c)^{16} [(-a)^2 -b]^{14} (-c)^{28} [-a - b (-a)^2 - b - a - b]^4 (-c)^{10}$$

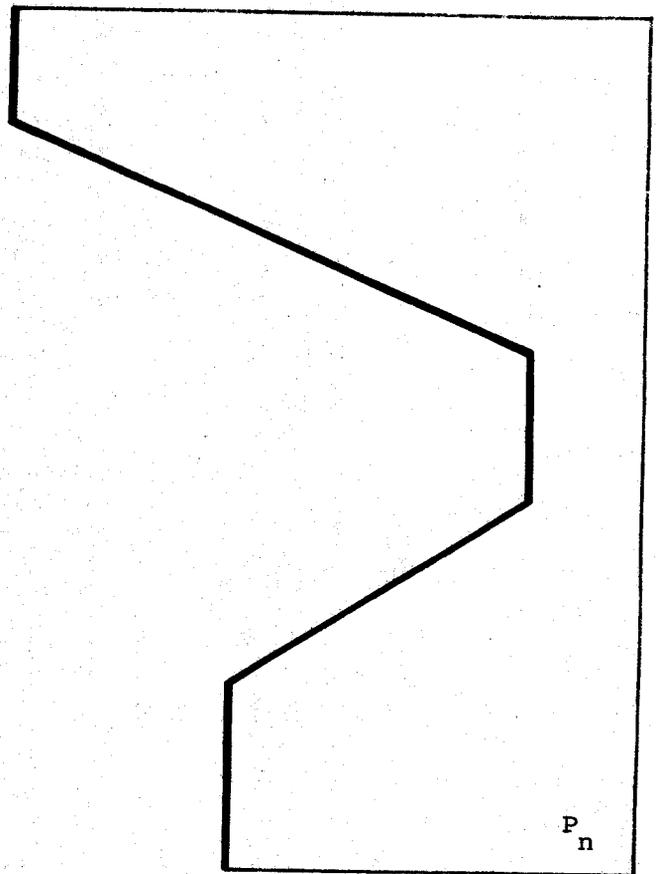
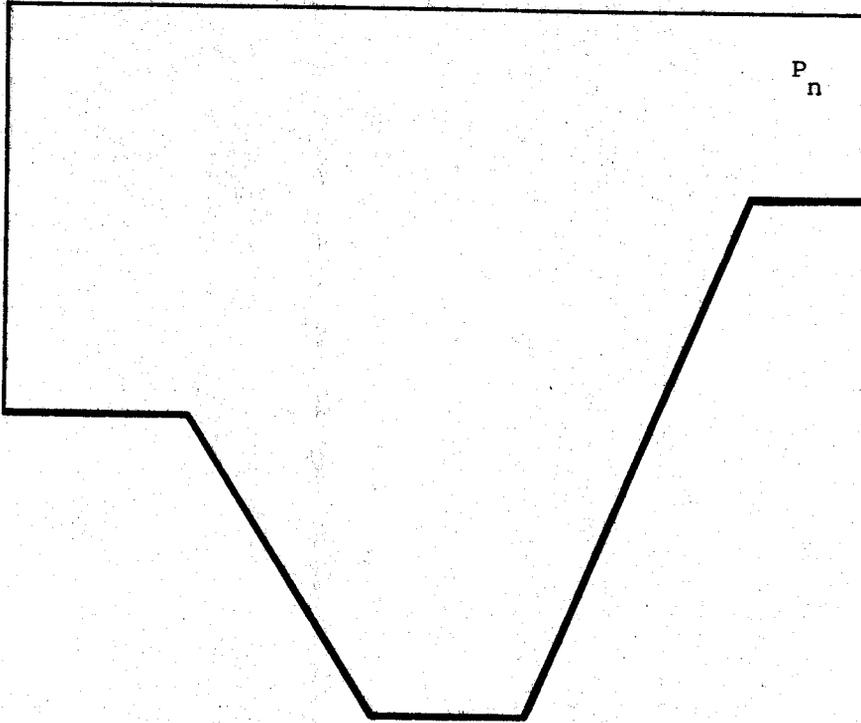


Figure 3



En effet, on vérifie aisément :

$$\text{Ar } (a^{16}) - \text{Ar } ((-c)^{16}) = \pi/2$$

$$\text{Ar } [[(-c)^2 (-d)]^{14}] - \text{Ar } [[(-a)^2 - b]^{14}] = \pi/2$$

$$\text{Ar } [[c^2 b (cb)^2]^4] - \text{Ar } [(-a -b (-a)^2 -b -a -b)^4] = \pi/2$$

$$\text{Ar } (a^{28}) - \text{Ar } [(-c)^{28}] = \pi/2$$

$$\text{Ar } (a^{10}) - \text{Ar } [(-c)^{10}] = \pi/2$$

$$\text{Lg } (a^{16}) = \text{Lg } ((-c)^{16}) = 16 \text{ (unités)}$$

$$\text{Lg } [[(-c)^2 -d]^{14}] = \text{Lg } [[(-a)^2 -b]^{14}] = 14 \sqrt{5}$$

$$\text{Lg } (a^{28}) = \text{Lg } [(-c)^{28}] = 28$$

$$\text{Lg } [[c^2 b (cb)^2]^4] = \text{Lg } [(-a -b (-a)^2 -b -a -b)^4] = 20$$

$$\text{Lg } (a^{10}) = \text{Lg } [(-c)^{10}] = 10$$

II.4 - Recherche de l'emboîtement le plus vraisemblable

Pour une couple de pièces (P_m, P_n) , la méthode précédente permet de détecter, le plus souvent, plusieurs couples de sous-formes emboîtables, soient $(S_{m_1}, S_{n_1}), (S_{m_2}, S_{n_2}), \dots, (S_{m_p}, S_{n_p})$ ces couples. Comme à chaque couple (S_{m_i}, S_{n_i}) correspond une possibilité d'assemblage des pièces P_m et P_n , on conclut alors théoriquement à l'existence d'une multiplicité d'assemblages solutions.

Malheureusement en pratique, il n'est pas opératoire de disposer ainsi potentiellement d'un éventail de possibilités d'assemblage de deux pièces : on cherche en effet une seule d'entre elles.

On se trouve donc devant un problème de choix d'une solution parmi plusieurs. Pour effectuer ce choix, on s'appuie sur une information supplémentaire introduite sous la forme d'une fonction explicite calculable sur chacune des couples $(S_{m_1}, S_{n_1}), (S_{m_2}, S_{n_2}), \dots, (S_{m_p}, S_{n_p})$. Cette fonction, appelée degré d'emboîtement, n'a pas une forme générale, car elle est spécifique au lot de pièces sur lequel opère l'assemblage.

Ce degré d'emboîtement est défini tel que : soient deux couples de sous-formes emboîtables (S_{m_i}, S_{n_i}) et (S_{m_j}, S_{n_j}) et Em_{i,n_i}, Em_{j,n_j} leurs degrés d'emboîtement respectifs, on dira que l'emboîtement de (S_{m_i}, S_{n_i}) est plus vraisemblable que celui de (S_{m_j}, S_{n_j}) si et seulement si :

$$Em_{i,n_i} > Em_{j,n_j}$$

Donc le degré d'emboîtement pourrait être un critère permettant d'évaluer la vraisemblance d'un assemblage réalisé en emboîtant deux sous-formes.

Soient $Em_{1,n_1}, Em_{2,n_2}, \dots, Em_{p,n_p}$ les degrés d'emboîtement associés aux couples $(S_{m_1}, S_{n_1}), (S_{m_2}, S_{n_2}), \dots, (S_{m_p}, S_{n_p})$ de sous-formes emboîtables détectées chez la couple de pièce (P_m, P_n) , l'assemblage qui serait réalisé correspond à l'emboîtement des sous-formes S_{m_i} et S_{n_i} telles que :

$$Em_{i,n_i} = \text{Max} (Em_{k,n_k})$$
$$0 < k \leq p$$

III - DEFINITION DES PARAMETRES D'ASSEMBLAGE

Après détection chez les deux pièces présentes dans la scène, pièces P_m et P_n , de deux sous-formes emboîtables S_m et S_n , la phase suivante de la procédure consiste à élaborer des informations portant spécifiquement sur la position et l'orientation de l'une de ces sous-formes par rapport à l'autre. A cet effet, on a été conduit à définir certains paramètres appelés 'paramètres d'assemblage'.

Ces paramètres sont au nombre de trois : il s'agit d'une direction, d'un point, et d'un vecteur qu'on désignera par la suite respectivement ' Direction d'assemblage ', ' Centre d'assemblage ' et ' Vecteur associé à l'assemblage '.

La donnée de ces paramètres permet l'acquisition d'informations

- non redondantes
- et suffisantes pour réaliser l'emboîtement (S_m, S_n).

III.1 - Définition de la Direction d'assemblage

On désigne par Δ_m et Δ_n les bissectrices intérieures respectives des angles formés respectivement par les primitives extrémités de S_m et S_n (figure 4).

Supposons que la pièce P_m soit la première saisie par le manipulateur, on définirait alors la Direction d'assemblage comme la bissectrice Δ_m associée à la sous-forme S_m .

En désignant par $\tan\theta_m$ et $\tan\alpha_m$, les pentes respectives des primitives extrémités de S_n et par $A_m(x_{m1}, y_{m1})$ et $B_m(x_{m2}, y_{m2})$ les sommets extrémités (figure 1) la Direction d'assemblage, noté Δ , admet pour équation :

$$\Delta \equiv y \cos \gamma_m - x \sin \gamma_m - c_m = 0$$

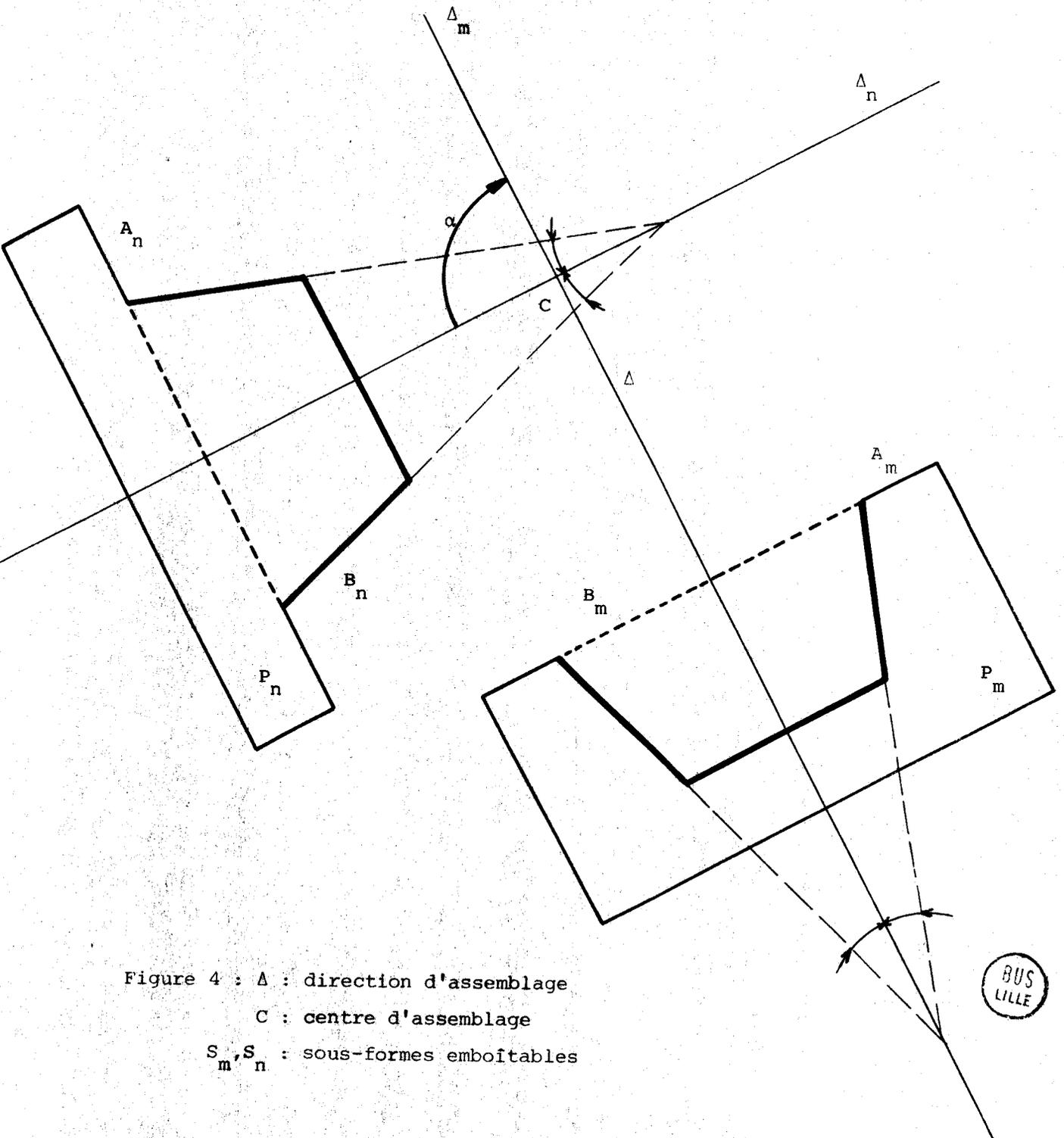


Figure 4 : Δ : direction d'assemblage
 C : centre d'assemblage
 S_m, S_n : sous-formes emboîtables

BUS
LILLE

$$\gamma_m = (\theta_m + \alpha_m)/2$$

$$c_m = \left[(ym_2 - ym_1) + (xm_1 \tan \alpha_m - xm_2 \tan \theta_m) \right]$$

$$\frac{\cos \gamma_m \tan \alpha_m - \sin \gamma_m}{\tan \alpha_m - \tan \theta_m} +$$

$$+ \cos \gamma_m (ym_1 - xm_1 \tan \alpha_m)$$

Remarque: Dans le cas particulier, où les primitives extrémités de S_m seraient parallèles, on choisirait comme Direction d'assemblage, la médiatrice du segment de droite $A_{m'}$, $B_{m'}$.

Son équation serait alors :

$$\Delta \equiv y \cos \theta_m - x \sin \theta_m - c_m = 0$$

où

$$c_m = \left[(ym_1 - ym_2) \cos \theta_m - (xm_1 - xm_2) \sin \theta_m \right] / 2$$

III.2 - Définition du Centre d'assemblage

Le Centre d'assemblage, noté c , est défini comme le point d'intersection des bissectrices (ou éventuellement médiatrices) Δ_m et Δ_n associées respectivement à S_m et S_n (figure 4).

L'équation de Δ_n étant :

$$\Delta_n \equiv y \cos \gamma_n - x \sin \gamma_n - c_n = 0$$

$$\gamma_n = (\theta_n + \alpha_n) / 2$$

$$c_n = \left[(y_{n_2} - y_{n_4}) + (x_{n_1} \tan \alpha_n - x_{n_2} \tan \theta_n) \right]$$

$$\frac{\cos \gamma_n \tan \alpha_n - \sin \gamma_n}{\tan \alpha_n - \tan \theta_n} +$$

$$+ \cos \gamma_n (y_{n_1} - x_{n_1} \tan \alpha_n)$$

les coordonnées (x_c, y_c) du Centre d'assemblage sont données par :

$$x_c = (c_n \cos \gamma_m - c_m \cos \gamma_n) / \sin (\gamma_m - \gamma_n)$$

$$y_c = -(x_c \sin \gamma_m + c_m) / \cos \gamma_m$$

Remarque : Quand les droites Δ_m et Δ_n sont parallèles ($\tan \gamma_m = \tan \gamma_n$)
le Centre d'assemblage est rejeté à l'infini.

III.3 - Définition du vecteur associé à l'assemblage

Soient $M = (x_M, y_M)$ et $N = (x_N, y_N)$ les points où les droites Δ_m et Δ_n coupent respectivement les segments de droites $A_m B_m$ et $A_n B_n$ (figure 4), le vecteur associé à l'assemblage, noté \vec{T} est défini comme suit (figure 5 - a, b, et c) :

$$\cdot \text{ Si } (\vec{CM}, \vec{CN}) + (A_n \vec{B}_n, A_m \vec{B}_m) = \pm \pi$$

$$| \vec{T} | = r_m + r_n$$

$$(\vec{T}, \vec{CM}) = 0$$

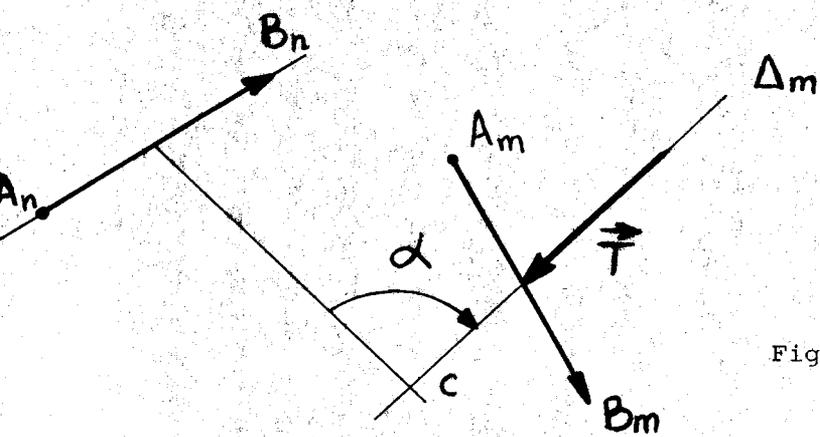


Figure 5.a : $|\vec{T}| = r_n - r_m$
 $(\vec{T}, \vec{CM}) = \pi$

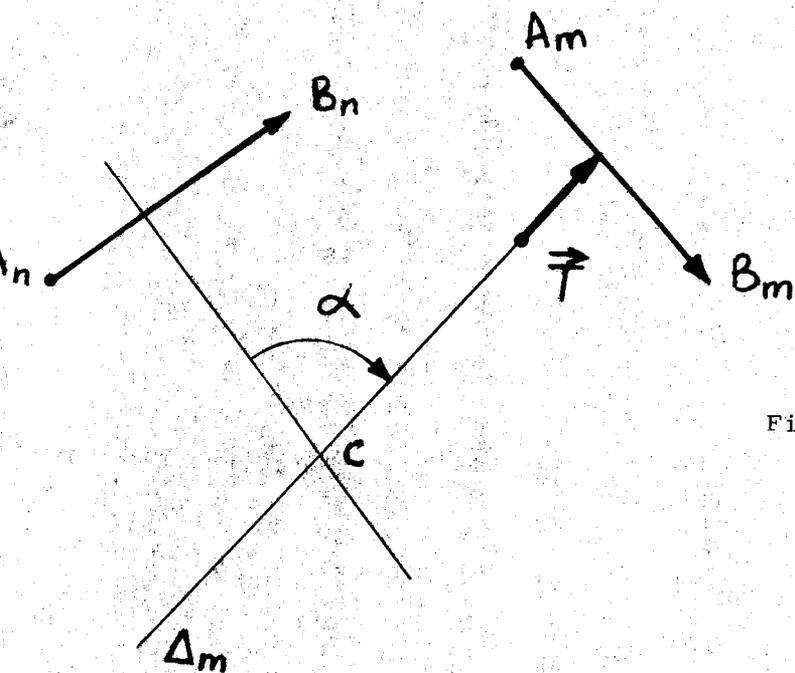


Figure 5.b : $|\vec{T}| = r_m - r_n$
 $(\vec{T}, \vec{CM}) = 0$

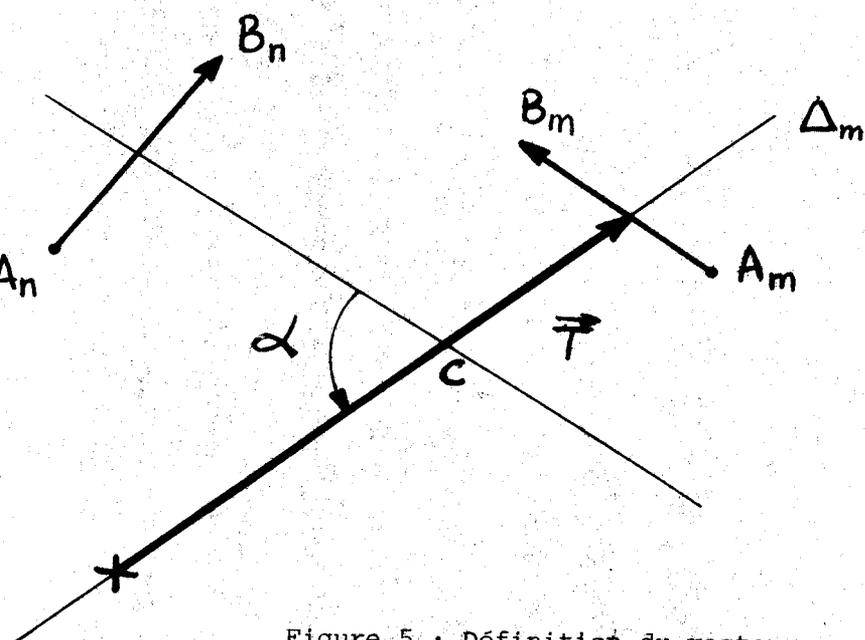


Figure 5.c : $|\vec{T}| = r_m - r_n$
 $(\vec{T}, \vec{CM}) = 0$



Figure 5 : Définition du vecteur associé à l'assemblage.

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

- /1/ U.RAMER
" An iterative procedure for polygonal approximation of plane curves "
J. Comput. Graphics and image processing, vol.1, pp 244-256,1972
- /2/ G.BIRKHOFF, C.R. DEBOOR
" Piecewise polynomial interpolation and approximation "
In Approximation of Functions, H.L Garabedian, Ed Amsterdam, The Netherlands : Elsevier, 1965, pp. 164-190.
- /3/ T.PAVLIDIS, S.L HOROWITZ
" Segmentation of Plane Curves "
I.E.E.E. Trans. on Comput, vol C-23, n°8, August 74
- /4/ H. VAN BRUSSEL, J. SIMONS
" The adaptable compliance concept and its rise for automatic assembly by active force feedback accommodations "
Proceedings of the 5 th International Symposium on Industrial Robots, Washington D.C., March 79
- /5/ M.K. HU
" Visual Pattern Recognition by Moment Invariants "
I.R.E. Trans. Info. Theory, vol. IT-8, pp.179-187..1962



$$\cdot \text{ Si } (\vec{CM}, \vec{CN}) + (A_n \vec{B}_n, A_m \vec{B}_m) = 0$$

$$| \vec{T} | = | r_m - r_n |$$

$$(\vec{T}, \vec{CM}) = 0 \text{ si } r_m > r_n$$

$$(\vec{T}, \vec{CM}) = \pi \text{ si } r_m < r_n$$

où $r_m = | \vec{CM} |$

$$r_n = | \vec{CN} |$$

IV - PROGRAMME RELATIF A LA TACHE D'ASSEMBLAGE

IV.1 - Description sommaire du manipulateur exécutant l'assemblage

La tâche d'assemblage sera exécutée par un manipulateur, dont la réalisation, actuellement en voie d'achèvement, concerne la seconde partie du projet qui possède quatre degrés de liberté (figure 6).

- trois : 1 rotation et 2 translations (θ, r, z) pour amener l'organe de préhension en un lieu déterminé de l'espace (réduit à l'environnement)
- et un : 1 rotation φ pour orienter convenablement cet organe par rapport à la pièce (bidimensionnelle) à saisir.

Le modèle de ce manipulateur à origine fixe est :

$$(\square, \theta, r, z, \varphi)$$

$$- 38 \leq \theta \leq 90$$

$$1 \leq r \leq 80$$

$$0 \leq \psi \leq 180$$

$$1 \leq z \leq 15$$

Les mesures s'expriment en degrés et en centième de mètre.

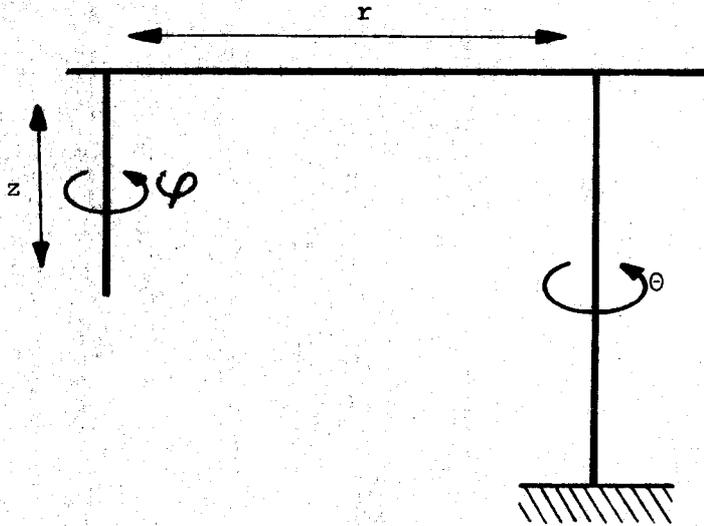


Figure 6 : Représentation du manipulateur à quatre degrés de liberté



IV.2 - Passage des coordonnées images aux coordonnées manipulateur

On désigne par P_r et P_a respectivement le plan de reconnaissance et le plan d'assemblage : Ces deux plans sont horizontaux et ont pour équations :

$$P_a \equiv z = k_a$$

$$P_r \equiv z = k_r$$

avec

$$k_a = d_1 - f$$

$$k_r = d_2 - f$$

f : distance focale

où d_1 et d_2 sont les distances respectives séparant les plans P_a et P_r et le centre de la lentille de la caméra.

Soit D l'intersection de l'axe vertical UV et du plan P_a , on définit par rapport au repère image $oxyz$ un nouveau repère $DXYZ$ (repère manipulateur) obtenu par translation de l'origine des axes :

$$\vec{OD} = x_0 \vec{x} + y_0 \vec{y} - k_a \vec{z}$$

On passe donc des coordonnées images (x_1, y_1, z_1) aux coordonnées manipulateur au moyen des relations suivantes :

$$\theta = \tan^{-1} \frac{y_1 - y_0}{x_1 - x_0}$$

$$r = \frac{x_1 - x_0}{\cos \theta}$$

$$z = z_1 - k_a$$

IV.3 - Définition d'un langage de commandes du manipulateur

Pour réaliser l'assemblage, le manipulateur devra pouvoir exécuter cinq types de commandes :

- . 'saisir une pièce' : SAISIR
- . 'soulever une pièce' : SOULEVER
- . 'poser une pièce' : POSER
- . 'transférer une pièce' : TRANSFERER
- . 'orienter une pièce' : ORIENTER

IV.3.1. - Commande 'SAISIR'

Quand on transmet au manipulateur la commande 'SAISIR' en lui fournissant deux types d'informations :

- informations proprioceptives concernant l'état actuel du manipulateur qu'on donne sous-forme d'un quadruplet

$$(\theta_0, r_0, z_0, \varphi_0)$$

- informations extéroceptives sur la pièce à saisir : reconnaissance et détermination de la position et de l'orientation

$$(F, \theta, r, z, \varphi)$$

le manipulateur exécute :

- une rotation par rapport à l'axe \vec{UV} : $\text{Rot} [\vec{UV}, (\theta - \theta_0)]$
- une translation le long de l'axe \vec{VW} : $\text{Tr} [(r - r_0) \vec{W}]$

- une rotation par rapport à l'axe \vec{WZ} : $\text{Rot} [\vec{WZ}, (\varphi - \varphi_0)]$

- et une translation le long de \vec{WZ} : $\text{Tr} [(k_r - z_0) \vec{Z}]$

où \vec{W} et \vec{Z} sont respectivement les vecteurs unitaires choisis sur les axes \vec{VW} et \vec{WZ} .

IV.3.2. - Commande 'SOULEVER'

Le manipulateur en recevant l'ordre ' SOULEVER ' exécute une translation le long de l'axe \vec{WZ}

$$\text{Tr} (- k_1 \vec{Z})$$

$$\text{si } k_a > k_r$$

$$\text{où } k_1 = \begin{cases} e \\ e + (k_r - k_a) \text{ sinon} \end{cases}$$

où e est un facteur qui intervient pour tenir compte de l'épaisseur des pièces à manipuler.

IV.3.3. - Commande 'POSER'

La commande 'POSER' provoque l'exécution d'une translation le long de l'axe \vec{WZ}

$$\text{Tr} [k_2 \vec{Z}]$$

$$\text{où } k_2 = \begin{cases} e & \text{si } k_r > k_a \\ e + (k_a - k_1) & \text{sinon} \end{cases}$$

IV.3.4. - Commande 'TRANSFERER'

Quand on transmet la commande 'TRANSFERER' en fournissant les informations sur la position de la pièce avant (θ_0, r_0) et après la tâche (θ, r) , le manipulateur exécute

. une rotation par rapport à l'axe UV : $\text{Rot} [UV, (\theta - \theta_0)]$

. suivie d'une translation le long de l'axe VW : $\text{Tr} [(r - r_0) \vec{W}]$

IV.3.5. - Commande 'ORIENTER'

En recevant la commande 'ORIENTER' et les informations portant sur l'orientation initiale (φ_0) et finale (φ) de la pièce, le manipulateur exécute une rotation par rapport à l'axe WZ

$$\text{Rot} [WZ, (\varphi - \varphi_0)]$$

IV.4 - Programme relatif à la tâche d'assemblage

On va écrire, à base du langage de commandes définie précédemment, le programme conduisant le télémanipulateur à l'exécution correcte de la tâche qui lui a été assignée.

. 1 SAISIR / la pièce P_m /

- informations proprioceptives : $(\theta_0, r_0, z_0, \varphi_0)$ sur l'état
actuel du manipulateur

- informations extéroceptives : $(F_m, \theta_m, r_m, \varphi_m)$ sur la pièce P_m

$$\theta_m = \tan^{-1} \frac{y_m - y_0}{x_m - x_0}$$

$$r_m = \frac{x_m - x_0}{\cos \theta_m}$$

φ_m : orientation angulaire de P_m

. 2 SOULEVER

. 3 TRANSFERER

- position initiale de la pièce : (θ_m, r_m)

- position prévue de la pièce (θ'_m, r'_m)

. 4 ORIENTER

- orientation initiale φ_m

- orientation finale φ'_m

$$\varphi'_m = \beta_m + \text{signe}(\beta_m) \left| \sin^{-1} \left(\frac{d}{r'_m} \right) \right|$$

$$\beta_m = \begin{cases} \gamma_m - \theta_m & \text{si } \tan \gamma_m > 0 \\ \pi + (\gamma_m - \theta_m) & \text{sinon} \end{cases}$$

$$d = \left| y_m \cos \gamma_m - x_m \sin \gamma_m - c_m \right|$$

Au terme de cette rotation, la Direction d'assemblage coupe l'axe \vec{UV} .

. 5 POSER

. 6 SAISIR / la pièce P_n /

- état actuel du manipulateur : $(\theta'_m, r'_m, k_a, \varphi'_m)$

- informations concernant la pièce P_n : $(F_n, \theta_n, r_n, \varphi_n)$

$$\theta_n = \tan^{-1} \left(\frac{y_n - y_0}{x_n - x_0} \right)$$

où x_n et y_n sont les coordonnées du point N.

. 7 SOULEVER

. 8 TRANSFERER

- position actuelle de la pièce P_n : (θ_n, r_n)

- position finale : (θ'_n, r'_n)

$$\theta'_n = \theta_n + \alpha'$$

où

$$\alpha' = \alpha - \text{signe } (\alpha) [(\varphi'_n - \varphi_n) + (\varphi'_m - \varphi_m)]$$

$$\alpha = \begin{cases} (\vec{CN}, \vec{T}) & \text{si } |\vec{T}| = r_n - r_m \\ (\vec{CN}, -\vec{T}) & \text{sinon} \end{cases}$$

$$\varphi'_n = \begin{cases} \varphi_n + \theta_n - \gamma_n & \text{si } \tan \gamma_n > 0 \\ \varphi_n + \pi + (\theta_n - \gamma_n) & \text{sinon} \end{cases}$$

et
$$r'_n = \frac{x_M}{\cos \theta_M}$$

où
$$\theta_M = \tan^{-1} \left(\frac{y_M - y_0}{x_M - x_0} \right)$$

$$x_M = x_m + MG_m \cdot \cos (\theta_m \text{ et } \varphi_m)$$

$$y_M = y_m + MG_m \sin (\theta_m + \varphi_m)$$

MG_M étant la distance séparant le point M du centre de gravité G_m de la pièce P_m .

. 9 ORIENTER

- orientation initiale : φ_n

- orientation finale : φ'_n

. 10 POSER

. FIN

CONCLUSION

Le calculateur chargé de générer les signaux de commandes du manipulateur a fait appel à deux types de données extéroceptives :

- celles concernant l'état actuel de l'environnement, il s'agit de :

- reconnaître les pièces : (F_m, F_n)

- les localiser : (θ_m, r_m) , (θ_n, r_n)

- et déterminer leur orientation : φ_m, φ_n

ceci afin de permettre leur saisie par l'organe de préhension

- et celles relatives à la tâche d'emboîtement des pièces qu'on a désigné par 'paramètres d'assemblage':

. Direction d'assemblage : Δ

. Centre d'assemblage : c

. et vecteur associé : \vec{f}

C'est à base de la donnée de ces " paramètres d'assemblage " qu'a été élaboré le programme de commandes qui fait progresser le télémanipulateur vers l'exécution correcte de la tâche d'assemblage.

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

- /1/ J. VERTUT, J. CHARLES, P. COIFFET, M. PETIT
" Reflecting Manipulator System "
Proceeding of second international symposium on theory
and practice of robots and manipulators, Varsovie, Sept.76
- /2/ C. BIGUEREAU, B. ESPIAU, J.J. BORRELY, C. LAURGEAU
" Recognition and sorting of mechanical pieces "
8th international symposium on industrial robots
30.5.78 - 1.6.78 Stuttgart, West Germany
- /3/ Kh. OURIACHI, M. BOURTON
" An assembly procedure of two dimensional objects by robot "
Proceedings of the 1 st international conference on
' ASSEMBLY AUTOMATION '
Brighton, UK. March 25-27 1980

CONCLUSION ET PERSPECTIVES D'EXTENSION

I - CONCLUSION ET PERSPECTIVES D'AMELIORATION

Les améliorations, nombreuses, apportées à la procédure de traitement d'images, ont permis de limiter sensiblement son coût en temps de calcul et en espace mémoire. Pour ne citer, à titre d'exemple, que les deux opérations, généralement, les plus importantes en traitement d'images, à savoir la recherche du contour et sa segmentation, on constate à la lumière des méthodes appliquées : la méthode gradient améliorée et la méthode de segmentation fondée sur les critères d'ATTNEAVE et de FREEMAN; le souci de recherche d'un compromis précision- temps calcul qui les rend opérationnelles.

Si l'on compare la méthode gradient améliorée à celle de HUECKEL, par exemple, dont la performance est comparable, on mesure l'avantage que procure la nouvelle méthode en évitant la lourdeur et la complexité de la mise en oeuvre de l'algorithme de HUECKEL.

Une remarque similaire serait faite si l'on compare la nouvelle méthode de segmentation du contour aux méthodes classiques telle la méthode itérative de RAMER /1/ ou celle dite d'interpolation de BIRKHOFF /2/ ou celle suggérée par PAVLIDIS qui utilise l'algorithme ' Slit / Merge ' /3/.

Ces deux nouvelles méthodes citées peuvent, dans une perspective très proche, motiver la recherche d'autres techniques plus performantes.

Le processus de localisation (recherche du Centre de gravité) des pièces dans la scène, est apparu relativement lourd, car en général pour localiser un objet, on se contentera d'une estimation grossière (l'erreur étant de l'ordre de trois à cinq pixels) de la position d'un point quelconque de cet objet. Néanmoins cette complexité est à la mesure de l'importance que revêt la précision du calcul du point de localisation qu'on a confondu volontairement avec le centre de gravité. En effet, ce calcul était intervenu, par la suite, directement pour la détermination de l'axe minimum d'inertie ou le calcul de la signature polaire du contour.

Dans la procédure de reconnaissance automatique, c'est la phase d'apprentissage qui est apparue, sans conteste, la plus délicate. Elle soulève en même temps un problème théorique et un problème pratique. Il s'agit en effet d'inférer une grammaire modélisant une forme. Avec ce problème d'inférence, on a touché à un problème relativement nouveau en théorie des langages. Et, toutes les recherches entreprises dans le cadre de la reconnaissance syntaxique butent sur un obstacle majeur : l'automatisation de l'inférence grammaticale. Ainsi remarque t'on le très peu de réalisations applicables à des problèmes concrets de reconnaissance des formes, qui ont vu jour en ce domaine. La raison est simple : s'il est, en effet, possible de décrire certaines formes par des grammaires, il est cependant plus précis et efficace encore, pour le moment, d'en réaliser l'apprentissage " à la main ". Pour notre part, nous nous sommes attachés à automatiser l'inférence d'une grammaire ou plus précisément d'un automate fini déterministe en appliquant une méthode d'inférence régulière qui utilise comme heuristique la recherche, dans un échantillon de phrases, des récursions caractéristiques de la grammaire qui l'engendre. Cette recherche de récursions est rendue particulièrement aisée, dans notre cas, en raison principalement de la structure des phrases qui composent l'échantillon d'inférence. Cela explique l'efficacité dont a fait preuve en cette occasion, la méthode d'inférence appliquée.

Cependant, il n'en demeure pas moins que cette méthode automatise un apprentissage de type particulier, en s'adressant précisément aux modèles réguliers réputés les plus simples des modèles syntaxiques. Donc elle n'est pas générale, et bien plus les modèles qu'elle infère se révèlent souvent inaptés à formaliser des structures réelles par ailleurs souvent entachées d'un bruit aléatoire qui rend vaine l'émergence de toute structure régulière.

D'où la nécessité d'extension de cette méthode. En effet, la procédure de reconnaissance décrite précédemment utilise une information du type " tout ou rien " : une phrase est acceptée ou non par l'automate de reconnaissance. Il est donc tout à fait souhaitable de donner un peu de souplesse à l'algorithme original en écrivant une procédure capable de calculer une " distance " de chaque phrase à chaque grammaire. Cette distance

pourrait être définie par exemple comme le nombre minimum de modifications de lettres à faire subir à une phrase donnée pour qu'elle devienne acceptable par un automate. Ce procédé d'analyse syntaxique tolérante qui permet de décider d'une manière moins tranchée que dans le cas présenté, peut être à la base d'une idée de formalisation d'une structure non régulière (bruitées) par un modèle régulier modélisant une structure très " voisine ".

Pour élaborer les commandes du télémanipulateur qui font programmer celui-ci vers l'exécution correcte de la tâche qu'on lui a assignée, on a eu recours à deux types d'informations extéroceptives:

- informations sur l'environnement pour identifier les pièces présentes dans la scène, et déterminer leurs positions sur le plan de reconnaissance, et leurs orientations angulaires.
- données relatives à l'emboîtement partiel ou total des pièces : paramètres d'assemblage.

Ces informations extéroceptives limitées ici à des informations visuelles, risquent de devenir insuffisantes quand parfois en raison des imprécisions des calculs ou surtout de l'imprécision du mouvement du télémanipulateur (les résolutions des déplacements angulaires θ et de la translation τ sont respectivement $0,5^\circ$, $0,7^\circ$, $0,3$ cm) ou de la conjugaison de ces deux facteurs, de petites corrections deviennent nécessaires afin de parfaire l'assemblage. D'où la nécessité d'adjoindre un organe tactile 'compensateur'. Cet organe est un capteur de forces de contact qui fournit des informations " Force-Feedback " et qui n'interviendrait que dans l'ultime acte de l'assemblage. Mais cependant, la mise en oeuvre d'un tel capteur est complexe et très coûteuse, de plus malgré les progrès réalisés ces dernières années, le domaine d'action actuel des capteurs force-feedback se limite à l'emboîtement de pièces ayant une forme cylindrique.

II - PERSPECTIVES D'EXTENSION DE L'ASSEMBLAGE A DES PIÈCES TRIDIMENSIONNELLES

L'extension de l'assemblage aux pièces tridimensionnelles est en particulier dans un premier temps, à des pièces assimilables à des polyèdres solides, soulève des problèmes nouveaux tant au niveau de l'acquisition

des formes des pièces en vue de leur reconnaissance, qu'au niveau de la détermination de leur orientation en vue de la tâche de saisie par l'organe de préhension.

Pour le choix du capteur, on penche vers une solution économique qui utilise une seule caméra mobile solidaire de l'organe terminal du manipulateur. En partant de l'hypothèse qu'une seule pièce est présente dans la scène analysée, la procédure d'acquisition de la forme utilise une approche qui consiste à ignorer la nature tridimensionnelle de la forme en la représentant par plusieurs images bidimensionnelles. Ces images sont des projections en perspective type " dessin industriel ".

La mobilité de la caméra peut être judicieusement exploitée si l'on élabore une procédure qui permet d'associer une stratégie de prises de vue à l'acquisition des informations sur la structure de la forme. Ainsi, à partir d'une position dite "générale" de la caméra, par rapport à la pièce - position telle qu'après de petites perturbations, il en résulterait une image similaire - On peut déduire le mouvement de la caméra qui conduit à des prises de vue riches en informations structurelles. Cette procédure est fondée sur l'analyse de la texture de l'image et sur l'interprétation des lignes (ligne convexe, ligne concave, ligne convexe cachée ...) et des sommets (quatre types de sommets ayant 13 représentations distinctes) détectés dans l'image.

L'orientation d'une pièce tridimensionnelle qui repose en équilibre stable, sur le plan de reconnaissance peut être déterminée en choisissant comme fonctions caractéristiques la combinaison de certains moments centraux de l'image invariants par rapport à la translation et la rotation de l'image sur elle-même /5/.

Pour la manutention des pièces tridimensionnelles, le télémanipulateur devrait posséder au moins six degrés de liberté.

- trois : θ , τ , et z pour amener l'organe de préhension en un lieu déterminé de l'espace atteignable par cet organe.

- et trois: φ , ψ et σ pour orienter convenablement cet organe par rapport à la pièce à saisir.

Le modèle du télémanipulateur serait donc :

(\square , θ , τ , z , φ , ψ , σ)