Un langage de description de concepts pour la formulation d'objectifs d'analyse

DEJEAN Ph., DALLE P.

Institut de Recherche en Informatique de Toulouse – Université Paul SABATIER
Pôle 2IST / Équipe TCI
118, route de Narbonne – 31062 TOULOUSE Cedex

e-mail : dejean@irit.fr dalle@irit.fr

1 Introduction

Dans cet article, après une revue des méthodes de représentation des opérateurs de traitement d'images et de formulation des concepts des domaines d'application, nous montrerons qu'une modélisation cohérente des données et des opérateurs basée sur un formalisme précis de description d'indices visuels permet de construire un système de planification de chaînes d'opérateurs de traitement d'images résolvant des problèmes ou sous-problèmes d'analyse d'images.

2 Représentation et description d'un opérateur

Dans le cadre de la conception de modules ou de systèmes d'interprétation d'images (planifieurs, systèmes experts en traitement d'images) ou de leur exploitation (contrôle d'exécution), différents modèles de représentation des opérateurs de traitement d'images ont été proposés. Ainsi selon les objectifs considérés, chacun de ces modèles est développé pour rendre compte de particularités caractéristiques de ces opérateurs ou de familles d'opérateurs. Dans [Dej96b], nous avons introduit ces représentations en constatant que les modèles des données utilisés dépendant généralement du modèle d'opérateurs adopté. Nous avons alors dégagé six principaux modes de représentation : par l'approche "fonctionnelle", par la décomposition algorithmique ([Ser89], [Rit90, Dav93]), par la présentation des résultats en termes de types de données ([IUE94], [Bel85], [Mat90]) ou en termes d'effets ([Sch89]), par la description des propriétés mathématiques ([Sch92]), par l'utilisation ([Cle93], [Lie92]), ou guidées par des contraintes spécifiques de développement ([Jun93], [Ebe91]).
Notre problématique étant de permettre de décrire une image afin de formuler un objectif d'interprétation, nous avons proposé une nouvelle modélisation permettant de rendre compte du traitement de l'information effectué, ce qui est fondamental pour les tâches de planification [Ing92] ou d'explication et de validation de séquences d'opérateurs.

2.1 Le modèle de donnée

Le modèle que nous exploitons [Dej96b], s'insère dans le processus de description symbolique de données de traitement d'images exprimées en termes d'indices visuels. Toute donnée de traitement d'images sera considérée sous la forme d'une structure composée des cinq éléments suivants : la trame, le cache, l'index, le signal et le codage.

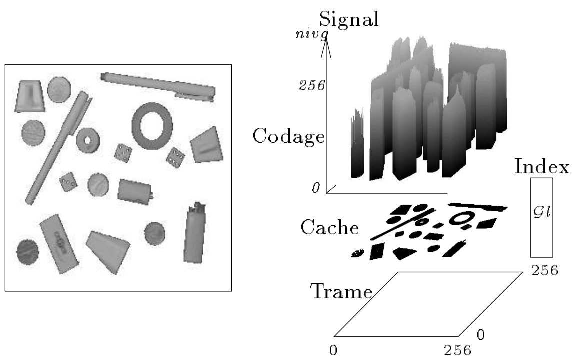


FIG. 1 - *Image et représentation*

2.2 Le modèle d'opérateur

Le modèle d'opérateurs de traitement d'images [Dal91, Dal93] décrit la transformation d'informations réalisée et non les conditions d'utilisation et/ou les résultats obtenus. Cette caractéristique introduit ainsi la notion de reconstruction des concepts du domaine de l'application, développée dans [Dej96a]. Ce modèle est représenté par une combinaison d'au plus cinq fonctions de modification d'expressions -*indexation*, *mesure*, *partition*, *codage*, *regroupement*- utilisant les notions de structuration, déstructuration, spécialisation, généralisation et nomination de concepts. L'intérêt de l'utilisation de ce modèle

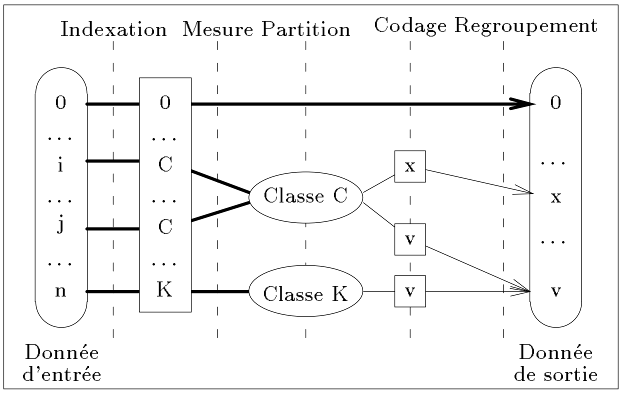


FIG. 2 - *Transformations d'informations*

réside dans la décomposition des divers traitements effectués et dans le formalisme rigoureux représentant l'effet de chaque traitement.

3 Un langage de description de concepts

3.1 Caractérisation

Des formalismes spécifiques et précis doivent être utilisés pour exprimer les données manipulées par les opérateurs et exprimer les fonctions du modèle des opérateurs eux-mêmes. Le langage de description des données, permettant de générer des expressions d'objectif -un objectif est une donnée particulière-, doit être facilement manipulable par les opérateurs qui transforment ces expressions. Les opérateurs doivent donc être implémentés sous la forme de fonctions de manipulation de ces expressions comme nous le verrons plus loin.
Les transformations d'informations peuvent être classées en deux groupes :

1. Les quantifications d'une entité de traitement d'image selon une caractéristique liée à l'opérateur *(par exemple : nombre de classes, surfaces des composantes connexes).* Ces opérations permettent d'exprimer des buts ou des sous-buts d'interprétation traduisant une mesure sur ces entités. Ils permettent de décrire des objectifs d'interprétation spécifiques ou de décrire les paramètres des critères de sélection d'entités. Ces sélections d'entités, fournissent des concepts de traitement d'images pouvant être combinées ou liés entre eux.

2. Les définitions d'une entité, qui comporte : les sélections d'entités de la donnée d'entrée selon un critère donné *(par exemple : composantes connexes dont la surface est supérieure à un seuil déterminé)* et la construction d'entité par combinaison d'entités spécifiques. La sélection va s'effectuer de deux façons différentes. La première consiste à spécialiser un concept. Cette spécialisation se fait en ajoutant une propriété, ou sélecteur, à un concept déjà formé ou en mettant en relation deux ou plusieurs concepts. La seconde consiste à généraliser un concept en étendant sa description par disjonction avec d'autres descriptions de concepts ou en lui ôtant une propriété.

Comme nous l'avons montré dans [Dal93] ce langage de description a été développé sur la base du calcul des prédicats annotés [Mic84]. Il permet de manipuler aisément les notions de transformations d'information, de les implanter et de les exploiter lors de la planification.
Une grammaire complète utilisée pour élaborer et contrôler ces objectifs est le support d'un analyseur de syntaxe vérifiant et simplifiant la description d'un objectif. Cette simplification, nous le verrons,

est nécessaire afin de permettre, dans le cadre d'une résolution d'objectif d'améliorer les possibilités du module de planification utilisé.

3.2 L'implémentation

L'implémentation de ces éléments -langage de description, donnée et opérateur- est faite en langage Lisp [LeL91]. Ils ont étés intégrés dans une nouvelle version du système de planification automatique de chaîne d'opérateurs : ExTI [Ing92]. Chacun des cinq constituants de la donnée s'exprime sous la forme d'expressions régulières du langage de description en fonction d'items grammaticaux de bases qui permettant de construire des descriptions d'images de plus en plus complexes syntaxiquement et sémantiquement. Une donnée sera alors implémentée sous la forme d'un couple de vecteur de cinq éléments décrivant respectivement les cinq champs de la données d'une part et la trame plus les quatre champs complémentaires d'autre part.
Pour exemple, l'expression suivante décrit la donnée de la figure 1 :

```
(  #[ ((0 256) (0 256)) (((nivg 100 256))) (gl) nivg (1 256) ]
   #[ ((0 256) (0 256)) (((nivg 0 100))) (gl) vab (0 1) ]        )
```

Les opérateurs sont représentés par une fonction Lisp qui transforme les champs des données en leur appliquant des fonctions de nomination, de structuration/déstructuration et/ou de spécialisation/généralisation conformément au rôle de l'opérateur sur les concepts manipulés. Chacun d'eux va alors déclencher, pour chacun des neuf champs à traiter, une fonction Lisp particulière. Le schéma général d'implémentation ci-dessous montre le principe d'une telle caractérisation d'un opérateur par les concepts manipulés pour l'implémentation : chaque intersection pourra être cochée selon la transformation (τ^t, κ^t, ι^t, σ^t, ω^t) effectuée sur le champ considéré de la donnée traitée et selon la nature de la transformation.

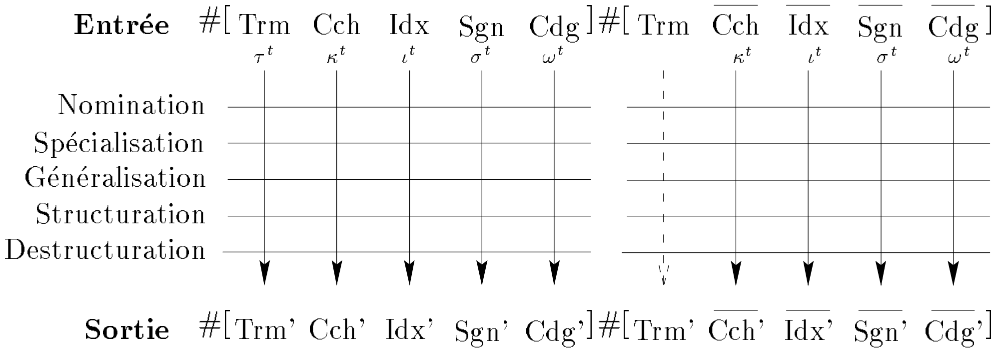


FIG. 3 - *Schéma général d'implémentation*

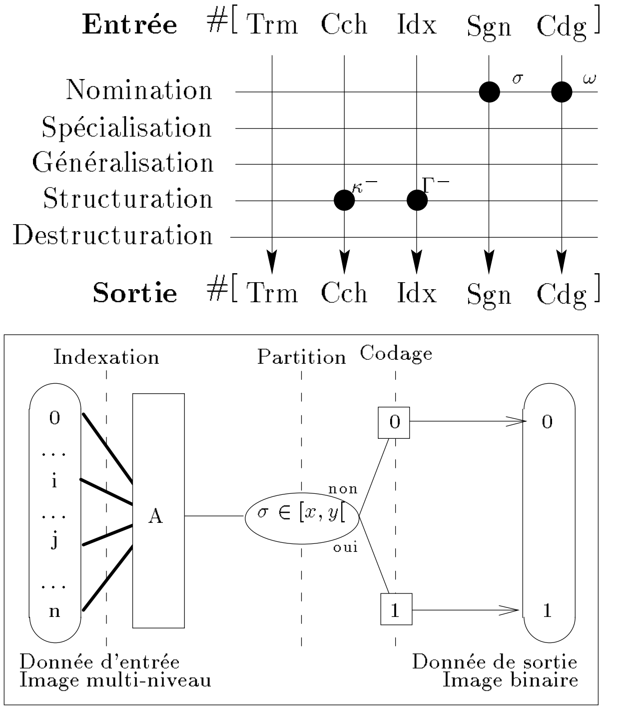
4 De la propagation au planifieur

4.1 La propagation d'information

Les opérateurs manipulant les champs des données, nous obtenons par enchaînement de ces manipulations des expressions de plus en plus complexes et vérifiant la notion de propagation d'information. Cette notion peut se traduire par le fait qu'à chaque étape d'une séquence d'opérateurs, la description de la donnée manipulée peut s'écrire à l'aide des termes du langage de description de concepts auxquels s'ajoutent les seuls éléments de description de la donnée initiale. Pour illustrer ceci, prenons l'enchaînement simple formé l'opérateur de seuillage et de l'opérateur de mesure de distance de KHOROS [Uni92] dont les description et implémentations sont les suivantes (fig.4).

Scripts KHOROS :
Seuillage : `vthresh -i` d `-l` i `-u` $(s-1)$ `-o` r
Distance : `vdistance -i` d `-o` r

Opérateur de seuillage	Opérateur de distance
$Sll(d,i,s) = \begin{bmatrix} \tau^t = \mathcal{I}d^{\tau} \\ \kappa^t = \kappa^-_{\{d.\sigma\in[i,s[\}} \\ \iota^t = \Gamma^- \\ \sigma^t = \sigma_{vab} \\ \omega^t = \omega_{vab} \end{bmatrix}$	$Dis(d) = \begin{bmatrix} \tau^t = \mathcal{I}d^{\tau} \\ \kappa^t = \mathcal{I}d^{\kappa} \\ \iota^t = \mathcal{I}d^{\iota} \\ \sigma^t = \Sigma^+_{dist} \\ \omega^t = \omega_{dist(d.\tau)} \end{bmatrix}$



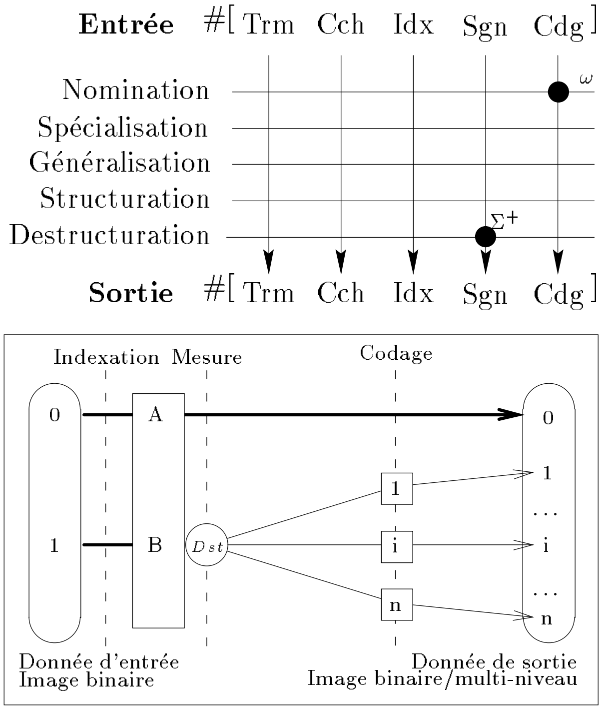
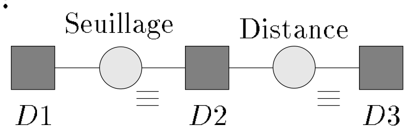


FIG. 4 - *Descriptions et schémas d'implémentation*

Enchaînement des opérateurs :



$$
\begin{aligned}
D2 &= Sll(D1, I, S) \\
&= \begin{bmatrix} \tau^t = \mathcal{I}d^{\tau} \\ \kappa^t = \kappa^-_{\{d.\sigma\in[i,s[\}} \\ \iota^t = \Gamma^- \\ \sigma^t = \sigma_{vab} \\ \omega^t = \omega_{vab} \end{bmatrix} \left(\left| \begin{array}{l} D1.\tau \\ D1.\kappa \\ D1.\iota \\ D1.\sigma \\ D1.\omega \end{array} \right. , \ I \ , \ S \right) = \left| \begin{array}{l} D2.\tau = D1.\tau \\ D2.\kappa = (D1.\sigma\ I\ S) *\ D1.\kappa \\ D2.\iota = (\mathcal{G}l) \\ D2.\sigma = vab \\ D2.\omega = [1\ 2[\end{array} \right. \\
D3 &= Dis(D2) \\
&= \begin{bmatrix} \tau^t = \mathcal{I}d^{\tau} \\ \kappa^t = \mathcal{I}d^{\kappa} \\ \iota^t = \mathcal{I}d^{\iota} \\ \sigma^t = \Sigma^+_{dist} \\ \omega^t = \omega_{dist(d.\tau)} \end{bmatrix} \left(\left| \begin{array}{l} D1.\tau \\ (D1.\sigma\ I\ S) *\ D1.\kappa \\ (\mathcal{G}l) \\ vab \\ [1\ 2[\end{array} \right. \right) = \left| \begin{array}{l} D3.\tau = D1.\tau \\ D3.\kappa = (D1.\sigma\ I\ S) *\ D1.\kappa \\ D3.\iota = (\mathcal{G}l) \\ D3.\sigma = dist \\ D3.\omega = [1\ Max(\Delta_{dist(D1.\tau)})[\end{array} \right.
\end{aligned}
$$

Nous constatons alors que la propagation des informations à travers la mini séquence (seuillage, distance) s'exprime uniquement en fonction de la donnée initiale $D1$.

4.2 Le planifieur

Nous possédons un modèle de donnée basé sur les indices visuels et un modèle d'opérateurs construit sur la manipulation de ces indices visuels permettant d'exprimer de façon rigoureuse des propagations

d'informations le long d'une séquence d'opérateurs ; de plus, un véritable langage de description de concepts permet d'exprimer en termes visuels des objectifs d'analyse. Nous avons donc développé un système de génération de plans pour automatiser la résolution d'objectif d'analyse.
Les spécificités du traitement d'images ont conditionné le choix du planifieur. Pour effectuer ce choix, il convient d'examiner différents points, éléments clefs d'un planifieur :

–Le monde :
Dans des approches classiques, les actions manipulent les objets constituant le monde connu dans l'état initial. Or en traitement d'images, le monde, constitué des images et des résultats de mesures effectuées, est construit tout au long de la phase d'exécution -réelle ou simulée- de la chaîne et seuls les opérateurs manipulent les données. De ce fait, la gestion de l'évolution de l'environnement par des événements extérieurs n'est pas à prendre en compte (SPEEDY [Bas96]).

–La représentation du monde :
Certains systèmes de planification utilisent un modèle d'opérateurs mettant en jeu des pré-conditions ou post-conditions d'utilisation (STRIPS [Fik71]). Ceci est nécessaire lorsque l'on souhaite diriger la création du plan en introduisant de la connaissance experte sur les opérateurs pour garantir la cohérence du plan. Comme nous l'avons vu précédemment, chaque "état" du monde dans lequel nous évoluons est décrit dans un langage précis, en termes d'indices visuels. Les opérateurs qui manipulent ces états n'ont alors pas de pré-condition ou post-condition concernant leur utilisation car c'est au moment de la planification que l'on vérifie si l'opérateur va produire l'effet voulu grâce à la description pécise de l'objet créé ; cette description pouvant être comparée à l'objectif.

–La recherche d'une solution :
Trois types de recherches sont couramment utilisées. La première est la recherche dans un espace d'états : recherche en avant (application des opérateurs de l'état initial vers l'état objectif), recherche en arrière ou mixtes. La seconde est la recherche dans un espace de sous-problèmes dans lequel l'objectif est décomposé en un arbre ET/OU de sous-objectifs. Enfin la troisième est la recherche dans un espace de plans (NOAH [Sac75]) décomposant lui aussi l'objectif en sous-problèmes réalisés par des plans déjà construits -problème de reconnaissance de plans, planification non-linéaire (ABTWEAK [Yan90]). Encore une fois le traitement d'images pose ses propres contraintes car les décompositions d'objectifs en sous-objectifs ne sont généralement pas simple à mettre en œuvre, l'état courant étant composé d'un unique élément : une image ou une mesure sur l'image.

–La stratégie :
Nous employons donc une recherche en avant et en profondeur d'abord mettant en œuvre un module de filtrage permettant de constater la différence entre la donnée produite et l'objectif (GPS [Win77]) afin de déterminer le point d'arrêt. Cette différence ne peut être calculée que grâce à la mise en œuvre de règles de simplifications d'expressions garantissant une forme pseudo-canonique de celles-ci.
De plus, un algorithme de réutilisation de plans a été développé afin d'optimiser la recherche de la solution en utilisant la notion vocabulaire du domaine -une expression d'objectif est attaché à un terme du domaine d'application (voir [Dej96a]).

–Le contrôle d'exécution :
Nous étudions actuellement l'implantation de la supervision et du contrôle d'exécution de la chaîne générée. Cette étude est menée afin de déterminer, à partir des modèles des entités manipulées, quels types de contrôles sont à implémenter et comment les gérer dans le planifieur.

Nous avons donc développé un planifieur spécifique à notre objectif. Nous possédons un système de planification linéaire d'opérateurs de traitement d'images, sans retour ("backtracking") dans l'arborescence grâce à un processus de conservation des opérateurs instanciés applicables aux données dont les descriptions complètes sont toujours présentes et accessibles. Ce processus permet d'éviter l'utilisation d'heuristiques (savoir-faire en utilisation du traitement d'images), et les inconvénients d'une explosion combinatoire du fait de l'utilisation de règles de simplification et de mesure de différence entre descriptions. De cette façon, un plan peut être vu comme la trace de la propagation des transformations contenues dans les diverses données, qu'elles figurent à un niveau intermédiaire, ou terminal -solution ou même échec. Cette planification permet entre autres, et cela pour chaque nœud du graphe, de connaître la description en termes d'entités de traitement d'images de l'information qu'il renferme.

5 Conclusion

Nous avons présenté des formalismes cohérents de description des entités manipulées en traitement d'images en termes d'indices visuels : les données, les opérateurs, les objectifs d'analyse dans des domaines d'applications spécifiques. Un langage précis et rigoureux permet de manipuler les expressions de ces entités afin de les simplifier, de les comparer, de les enrichir (spécialisation) ou de les étendre (généralisation). Les modèles mis en place vérifient la notion de propagation d'information effectuée par les opérateurs sur les données et permettent ainsi de développer un système original de planification de chaînes d'opérateurs de traitement d'images résolvant des problèmes ou sous-problèmes d'analyse d'images et utilisant ainsi le langage de description et les règles de manipulation d'expressions.

Références

[Bas96] Ch. Bastié et Pierre Régnier. – Intégration de la planification et du contrôle d'exécution en environnement dynamique: le système SPEEDY". *In : 10e RFIA, Rennes (France)*, 1996, pp. 879–888.

[Bel85] A. Belaïd. – Un langage de traitement d'images fondé sur des concepts de types abstraits. *Techniques et Sciences Informatiques*, vol. 4, n° 3, 1985, pp. 309–323.

[Cle93] V. Clement et M. Thonnat. – A knowledge-based approach to integration of image processing procedures. *CVGIP : Image Understanding*, vol. 57, n° 2, Mars 1993, pp. 166–183.

[Dal91] P. Dalle et J.-M. Inglebert. – Modèle de description d'opérateurs de traitement d'images. *In : 8e RFIA, Lyon-Villeurbanne (France)*, 1991, pp. 257–266.

[Dal93] P. Dalle, Ph. Dejean et J.-M. Inglebert. – Méthodologie de conception d'applications de traitement d'images. *In : 4e Journées ORASIS, Mulhouse (France)*, 1993, pp. 92–95.

[Dav93] J.L. Davidson. – Classification of Lattice Transformations in Image Processing. *CVGIP : Image Understanding*, vol. 57, n° 3, Mai 1993, pp. 283–306.

[Dej96a] Ph. Dejean et P. Dalle. – Image analysis operators as concept constructors. *In : IEEE Southwest Symposium on Image Analysis and Interpretation, San Antonio, Texas (USA)*, Avril 1996, pp. 66–70.

[Dej96b] Ph. Dejean et P. Dalle. – Modèle symbolique de la donnée de traitement d'images. *In : 10e RFIA, Rennes (France)*, 1996, pp. 746–755.

[Ebe91] D. Eberly et D. Wenzel. – Adaptation of Group Algebras to Signal and Image Processing. *CVGIP : Graphical Models and Image Processing*, vol. 53, n° 4, Juillet 1991, pp. 340–348.

[Fik71] R.E. Fikes et N.Nilson. – STRIPS : A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, vol. 2, n° 3/4, 1971, pp. 189–208.

[Ing92] J. M. Inglebert. – *Planification automatique de chaînes de traitement d'images.* – Thèse de 3e cycle, Université P. Sabatier de Toulouse (France), 1992.

[IUE94] *Image Understanding Environment, Program – IUE Overview Document.* – ftp.aai.com (192.190.241.40), 1994.

[Jun93] E. Jungert et S.K. Chang. – An Image Algebra for Pictorial Data Manipulation. *CVGIP : Image Understanding*, vol. 58, n° 2, Septembre 1993, pp. 147–160.

[LeL91] *LE-LISP version 15.25.* – I.N.R.I.A. Domaine de Voluceau Rocquencourt – 78153 Le Chesnay (France), 1991.

[Lie92] C. E. Liedtke et A. Blömer. – Architecture of the knowledge based configuration system for image analysis CONNY. *In : 11e ICPR, The Hague (The Netherlands)*, 1992, pp. 375–378.

[Mat90] T. Matsuyama et V. Hwang. – *SIGMA : a knowledge-based aerial image understanding system.* – Plenum Press, 1990.

[Mic84] R.S. Michalski, J.G. Carbonell et T.M. Mitchell. – A theory and methodology of inductive learning. *In : Machine learning : an artificial intelligent approach*, pp. 83–134. – Berlin : Springer-Verlag, 1984.

[Rit90] G.X. Ritter, J.N. Wilson et J.L. Davidson. – Image Algebra : An Overview. *CVGIP*, vol. 49, n° 3, Septembre 1990, pp. 297–331.

[Sac75] E.D. Sacerdoti. – The non linear nature of plans. *In : IJCAI*, 1975.

[Sch89] M. Schmitt. – Mathematical Morphology and Artificial Intelligence : An automatic programming system. *Signal Processing*, vol. 16, n° 4, 1989, pp. 389–401.

[Sch92] M. Schmitt. – *La morphologie mathématique en analyse d'images.* – Rapport technique n° 4, Thomson-CSF, Décembre 1992.

[Ser89] V. Serfaty–Dutron, B. Hoeltzener–Douarin et B. Zavidovique. – Exemple de conception mécanique d'Opérateurs de T.I. *In : 7e RFIA, Paris (France)*, 1989, pp. 75–88.

[Uni92] University of New Mexico. – *Khoros Reference Manual*, 1992.

[Win77] P.H. Winston. – *Artificial Intelligence.* – Addison-Wesley Publishing Compagny, 1977, pp. 130–143.

[Yan90] Q. Yang et J.D. Tenenberg. – ABTWEAK : abstracting a nonlinear, least commitment planner. *In : IJCAI*, 1990.