Planification en Traitement d'Image : Approche basée sur les données

Planning in Image Processing : Data based approach

DALLE P. DEJEAN Ph.[1]

Institut de Recherche en Informatique de Toulouse
2IL/TCI, Université Paul SABATIER
118, route de Narbonne — 31062 TOULOUSE Cedex
dalle@irit.fr

Résumé

La conception des chaînes d'opérateurs de traitement d'image résolvant des problèmes particuliers, et leur adaptation à des conditions opératoires variables nécessitent de disposer d'outils de configuration et de planification dynamique. Les approches traditionnelles utilisent une description fonctionnelle des opérateurs et introduisent une expertise pour la sélection et l'adaptation des opérateurs. Nous présentons une approche complémentaire, basée sur un formalisme unique permettant d'exprimer un objectif de traitement d'image et de représenter les effets des opérateurs sur les données initiales. Nous montrons qu'il est alors possible de planifier des chaînes de traitement d'image, d'implémenter des mécanismes intéressants tels que la réutilisation de plans et de fournir les éléments de base (construction et manipulation d'un vocabulaire exécutable) d'un environnement interactif de formulation d'objectif.

Mots Clef

Planification – Modélisation – Représentation des Connaissances – Traitement d'Images

Abstract

The elaboration of graph of Image Processing operators which solves specific problems, and adaptation to variability of application require tools for designing and dynamic planning. Traditionnal methods use functionnal description of operators and include expert knowledge to select and adapt these operators. We propose a complementary approch based on only one formalism to express Image Processing goal and to represent effects of operations on initial data. We show how it is possible to plan the programs, to implement interesting mechanisms as plans reuse, and to bring the bases -formation and handling executable vocabulary- of interactive environnement of goal formulation.

Keywords

Planning – Modelling – Knowledge Representation – Image Processing

1 Introduction

Ce travail se situe dans le cadre de la planification d'un enchaînement d'opérateurs réalisant une tâche d'analyse d'image. La conception d'un système d'interprétation d'image mettant en jeu des opérateurs travaillant à différents niveaux d'abstraction, peut être vue comme un processus de configuration des opérateurs pertinents d'une bibliothèque de traitement d'images (TI). La difficulté, pour un utilisateur final, de connaître le traitement d'image et de maîtriser la richesse croissante des bibliothèques de TI, le besoin de faire interagir différentes approches pour résoudre des problèmes complexes et enfin la nécessité d'adaptation au domaine d'application et au contexte d'utilisation, imposent d'essayer d'automatiser la planification ou d'assister l'utilisateur dans cette tâche, et de coupler planification et exécution.

La réalisation d'un planifieur peut se faire en introduisant des connaissances expertes sur le choix des opérateurs : celles-ci permettent de relier l'action d'un opérateur, ses effets et ses conditions d'utilisation, à un objectif de TI. Elles comprennent également des informations sur l'utilisation de ces opérateurs (initialisation et ajustement des paramètres) de façon à rendre le programme adaptatif.

1. Actuellement à REALIX Technologies, BP 146, 31676 LABEGE Cedex – dejean@realix.fr

Une autre approche, qui n'est pas exclusive de la première, vise à introduire une connaissance plus profonde sur le processus de TI, en décrivant l'action réellement opérée sur les données et sur l'information qu'elles véhiculent. La modélisation de ces opérateurs doit s'appuyer sur un modèle précis des données. Nous rappellerons les modèles de données et d'opérateurs qui ont été présentés en [11] et en [10]. Nous montrerons comment ces modèles peuvent être exploités en planification, à la fois pour exprimer un objectif et pour trouver la chaîne d'opérateurs qui le résout. Enfin nous montrerons que ces formalismes offrent de nouvelles possibilités d'expression des connaissances expertes sur l'utilisation du TI permettant ainsi la construction de systèmes autonomes et adaptatifs.

2 Planification d'opérateurs

2.1 Les éléments du planifieur

La conception d'un planifieur de TI dépend de trois éléments :

.1. le modèle de formulation d'objectifs : exprimé en terme d'opérations de TI à réaliser *(segmentation)*, d'information à extraire *(chercher les objets sombres allongés)* ou d'objets de la scène à identifier *(trouver le réseau routier)*. On retrouve les notions de planification par actions ou par états.

.2. le type de connaissance *a priori* [13] introduite dans le système et utilisée par le planifieur : liens buts-opérateurs, via leur fonctionnalité [16], liens indices visuel-effets d'opérateurs [6], information de sélection d'opérateurs d'après les types de données (préconditions) [15], squelettes de plans, hiérarchies d'opérateurs ou de données, information plus profonde -modélisation du processus de formation des images.

.3. l'articulation entre planification et exécution -planification puis exécution sans contrôle, planification de systèmes adaptatifs et réactifs, planification et exécution intégrées- et la manière dont est introduite l'information de contrôle [1] : mécanisme de généralisation-spécialisation, règles d'initialisation ou d'ajustement de paramètres, module de raisonnement, etc.

2.2 Les difficultés particulières de la planification du TI

On retrouve, dans le cas du TI, les problèmes classiques de toute conception de planifieur mais avec des spécificités dues à la nature des données traitées et au type de problèmes à résoudre. Une difficulté essentielle provient de ce que dans tout système d'analyse du signal on doit rechercher ou manipuler des concepts qui sont présents dans les données mais de manière non explicite; en effet une des tâches principales du système est justement d'extraire, d'isoler et de caractériser ces éléments. Le premier problème va donc concerner la formulation de l'objectif : s'il est exprimé en termes du domaine, l'objectif et les résultats des étapes du plan seront exprimés dans deux formalismes différents, ce qui entraînera un module d'appariement complexe intégrant de la connaissance en TI. Inversement, si on voulait exprimer l'objectif en terme de traitements du signal image, il faudrait connaître les étapes de ce traitement et donc avoir résolu le problème. Les solutions retenues sont généralement intermédiaires entre ces deux extrêmes : soit fournir une résolution grossière sous la forme d'un squelette d'application qu'il faudra instancier, affiner et adapter [7] -il s'agit donc davantage de pilotage de programme que de planification [5]-, soit exprimer le problème en terme d'indices visuels, qui forment un niveau intermédiaire entre les objets du domaine et les types de données du TI. Cela exige donc de savoir reformuler le problème en utilisant ces indices, mais sans avoir à indiquer les opérateurs qui les détectent ; c'est la solution que nous avons retenue [3] en fournissant un environnement d'aide à la formulation d'objectif [8].

La planification par action, utilisée quand on fournit des squelettes d'application, soulève d'autres problèmes de mise en œuvre concernant :

.1. le degré de généralité du plan (ou de granularité des actions). Si le plan est trop général, il sera difficile à instancier. Inversement, s'il est trop précis, c'est un plan *ad hoc* qu'il faudra réinventer à chaque nouvelle application.

.2. la construction du squelette. Elle pose le problème de l'existence et de la validité de l'expertise ou celle d'outils facilitant cette construction.

.3. l'adaptation et l'instanciation du plan, qui nécessitent soit d'introduire des règles expertes d'ajustement, soit de construire un module de planification dynamique.

De son côté, la planification par états nécessite de savoir décrire le TI (opérateur) et les résultats (données) dans un formalisme cohérent et précis. S'il est illusoire de penser pouvoir décrire globalement tous les détails du TI d'une application complexe, la planification par états et le contrôle d'exécution imposent cependant une description bas niveau des états élémentaires. Pour lever cette contradiction, on devra associer la précision de représentation des sous-chaînes avec des mécanismes de généralisation et de combinaison de ces sous-chaînes.

Enfin, les caractéristiques des données traitées, bruitées et variables, et celles des modèles du monde, incomplets et incertains, entraîneront des erreurs lors de l'exécution du plan. Les formalismes et méthodes de planification de chaînes de TI doivent donc permettre aussi la planification de mécanismes de supervision, de contrôle d'exécution et de récupération d'erreurs (prévision, détection et réparation d'erreurs).

3 Approche par les données : Principes et Modèles

3.1 Cahier des charges d'un formalisme pour la planification

Le module de résolution d'objectifs génère des descriptions de données à partir des opérations de transformations associées aux opérateurs et de la description du but à atteindre. Il faut donc considérer deux caractéristiques particulières. La première porte sur l'appariement entre les descriptions construites et celle de l'objectif, tandis que la seconde porte sur la cohérence du plan construit.

Nous pouvons concevoir l'appariement entre les deux types de description selon les trois façons rapportées sur la figure 1.

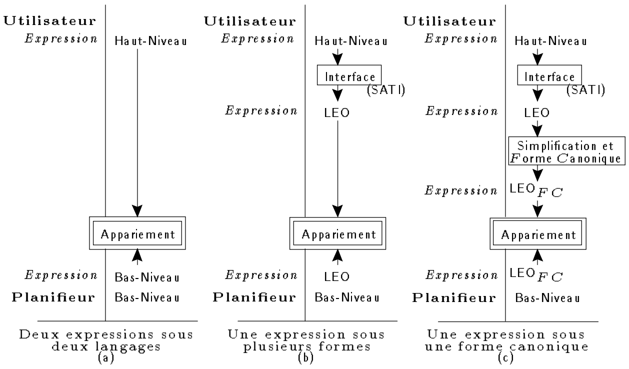


FIG. 1 - *Différents appariements envisageables*

Dans le premier cas (a), le travail du module d'appariement des expressions doit établir des correspondances entre deux expressions écrites dans deux langages de haut et bas-niveau, tout à fait différents. Ce problème est classiquement résolu (b) en proposant un langage de bas-niveau d'expression et de description des données construites par planification. LEO [12] a été écrit dans ce sens et permet, via une interface utilisateur, d'exprimer des objectifs en termes d'indices visuels. Les expressions construites et celles posant le problème étant générées par deux systèmes distincts, le module d'appariement doit, dans ce cas, effectuer le lien entre ces expressions en tenant compte de caractéristiques telles que l'organisation des concepts dans une disjonction, la redondance de termes, etc. Le principe de la simplification et de la mise en forme canonique des expressions (c) répond précisément à ce problème en proposant, au module d'appariement, deux expressions respectant des contraintes d'écriture. D'un côté, la description de l'objectif est analysée, simplifiée et réduite dès le problème soumis au processus de résolution. De l'autre côté, les transformations d'informations sont effectuées par les opérateurs symboliques au regard de règles de simplification et de réduction. Ainsi, le système d'appariement est allégé de ces phases de traduction et de mise en forme d'expressions, ce qui diminuera d'autant le temps de planification.

Le processus de simplification d'expressions permet en outre d'améliorer la lisibilité des expressions générées automatiquement et, de ce fait, de mieux appréhender les entités représentées. Il garantit enfin qu'un même objet sera toujours généré avec la même description -ce qui évite le développement de sous plans concurrents à partir d'états synonymes.

En conclusion, le but et le résultat s'exprimant dans le même formalisme, celui-ci est forcément construit à partir des données produites par les opérateurs. De plus, il faut que la représentation des opérateurs permette de propager les effets des transformations qu'ils réalisent sur les données. On ne peut donc pas se satisfaire d'une représentation qui qualifierait seulement les données d'entrées et fournirait les propriétés des données de sortie : en effet les données issues d'un même opérateur mais placé à des positions différentes de la chaîne de traitement doivent avoir des descriptions différentes. C'est donc la transformation $Entree \Rightarrow Sortie$ qu'il faut modéliser et représenter.

3.2 Les modèles

Une donnée de traitement d'image.

Le modèle doit permettre de rendre compte des deux types de transformations effectuées par les opérateurs : transformation de types de données et transformation de l'information qu'elles véhiculent. De plus il doit intégrer les deux types d'information associée aux pixels des données images : information radiométrique et géométrique ou spatiale. Dans [11], nous avons présenté en détail un modèle de donnée exprimé en termes d'indices visuels et que nous résumons ici. Toute donnée de TI est considérée sous la forme d'une structure composée des cinq éléments suivants (fig.2) : la trame, le cache, l'index, le signal et le codage.

• **La trame** (τ) est définie par l'ensemble des points p de la donnée dont les coordonnées sont comprises entre deux bornes.

• **Le cache** (κ) possédant une description δ est déterminé par l'ensemble des points p de τ vérifiant la

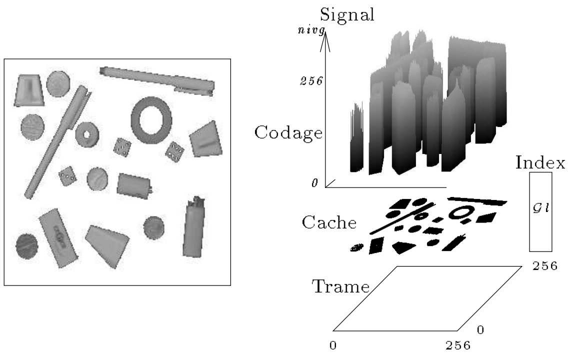


FIG. 2 - *Donnée image et représentation*

fonction cache f_δ à valeurs booléennes. Il a donc la même dimension que la trame et fournit une information de type géométrique ou morphologique sur les éléments pertinents de la donnée. Il permet ainsi de définir les zones d'intérêt.

• **L'index** (ι) permet de regrouper les zones d'intérêt représentant un même concept par étiquetage des composantes connexes de son cache κ associé.

• **Le signal.** (σ) Les points de cette donnée, vus au travers de son cache, renferment des informations d'ordre sémantique que l'on doit représenter. Nous appellerons signal la nature de l'intensité des points du cache reportés sur la donnée.

• **Le codage.** (ω) Le signal caractérisant la sémantique de l'information renfermée dans l'intensité du point de la donnée, le codage fournira les informations sur la valeur numérique du signal.

À cette structure composée de cinq champs se rajoute une structure de quatre éléments décrivant la partie complémentaire de la donnée décrite. Ainsi nous devrons toujours avoir complémentarité entre le cache et son complémentaire et complémentarité des codages par rapport à ce que nous avons appelé domaine de validité du signal de la donnée ; ce domaine de validité étant l'ensemble des valeurs pouvant être prises par le signal lorsque le cache est global.

Un opérateur de traitement d'image.

Un opérateur ayant une double action -sur les données et sur les concepts-, la représentation symbolique des transformations que nous proposons est basée sur les manipulations de ces concepts ([14],[4]) construisant et manipulant les entités de la donnée *(pixels, composantes connexes, ...)* en appliquant les mécanismes suivants :

• La *spécialisation* extrait l'information associée à un concept spécifiant un nouveau prédicat *(caractéristique, domaine)* vérifié par les instances de ce concept.

• La *généralisation* consiste à combiner de façon disjonctive plusieurs concepts ou à étendre les domaines de certains prédicats.

• La *nomination* de concepts est matérialisée par la représentation, dans le domaine d'expertise, de la donnée produite.

• La *structuration* consiste en un regroupement dans un niveau supérieur de considération des entités (pixels, composantes connexes), de celles possédant une caractéristique commune en respectant une certaine relation -distance- spatiale.

• La *déstructuration* permet à l'inverse de considérer, afin de les traiter séparément, les informations relatives aux éléments constituants de l'entité manipulée.

Les transformations effectuées par les opérateurs sont détaillées dans [9]. Elles font intervenir des descripteurs de TI auxquels sont associés des domaines de validité définis plus haut et intègrent les cinq étapes élémentaires qui peuvent avoir lieu au cours d'un traitement :

• L'*identification* détermine les sous-structures de la donnée initiale. Ces éléments de la donnée correspondant à même un identificateur feront l'objet d'un même traitement.

• La *mesure* produit une valeur directement calculée sur une sous-structure de la donnée. Cette évaluation est effectuée en utilisant une caractéristique spécifique, liée à l'opérateur.

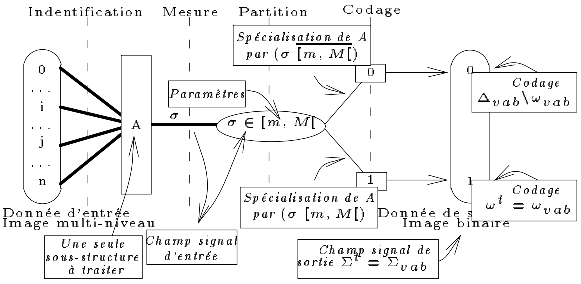
• La *partition* consiste en la séparation des entités en plusieurs classes conformément à un prédicat donné -sélecteur, domaine- constitué de la mesure précédente pour le sélecteur et des paramètres définissant le domaine, eux aussi très liés à l'opérateur. Ceci constitue donc une opération de spécialisation de l'information attachée à la sous-structure par le prédicat propre à l'opérateur.

• Le *codage* consiste en l'attribution d'une valeur de sortie aux différentes entités réunies dans les classes provenant de la partition.

• Le *regroupement* rassemble, sous la même valeur de la donnée produite, les éléments issus des opérations précédentes qui ont reçu la même valeur de codage.

L'opérateur est donc décrit en indiquant comment ces opérations sont réalisées sur chacun des 5+4 champs de la donnée. À partir de là, nous pouvons fournir une représentation fonctionnelle des opérateurs de TI (fig.3 – Extrait de [9]) permettant d'exprimer des propagations d'informations le long d'un enchaînement d'opérateurs et de décomposer des opérateurs en une succession d'opérateurs plus simples dont l'enchaînement produit les mêmes transformations d'informations (fig.4).

Opérateur KHOROS `vthresh`
a) opérateur réel :
·*Script associé :* `vthresh -i X -l m -u (M-1) -o R`
·*Paramètres :*
  Donnée d'entrée : image multi-niveaux `-i X`
  Donnée de sortie : image binaire `-o R`
  Seuils : entiers / réels `-l m -u (M-1)`
b) traitement :
·*Identification :* une seule sous-structure A.
·*Mesure :* aucune.
·*Partition :* des valeurs du signal selon le paramètre intervalle fourni.
·*Codage : valeur 1 pour la sous-classe de A issue du résultat de la partition tel que "signal $\in$ intervalle" et valeur 0 pour l'autre sous-classe issue de la partition.*
·*Regroupement :* aucun.



c) représentation fonctionnelle :

$$
Sll(X,m,M) = \left(\begin{array}{l} \left[\begin{array}{l} \tau^t = \mathcal{I}d^\tau \\ \kappa^t = \kappa^-_{\{X.\sigma \in [m,M[\}} \\ \iota^t = \Gamma^- \\ \sigma^t = \Sigma_{vab} \\ \omega^t = \omega_{vab} \end{array} \right] \\ \left[\begin{array}{l} \kappa^t = \kappa^+_{\{X.\sigma \in (\Delta_{X.\sigma} \setminus [m,M[)\}} \\ \iota^t = 0 \\ \sigma^t = \Sigma_{vab} \\ \omega^t = \Delta_{vab} \setminus \omega_{vab} \end{array} \right] \end{array} \right)
$$

FIG. 3 - *Opérateur de seuillage (base KHOROS)*

Le travail qui consiste à caractériser chaque opérateur de traitement par l'identification des fonctions de transformations qu'il met en œuvre, tant sur le plan du traitement de la donnée que sur le plan de la construction de concepts de l'application, n'est pas simple. Il fait appel à l'expertise du TI. Tous ces opérateurs peuvent être alors structurés selon des critères particuliers relatifs aux champs de la donnée atteints par ces transformations. De là, ce modèle peut être exploité pour faciliter l'aide à la mise au point de programme, pour implémenter des stratégies de planification spécifiques ou pour développer des outils d'apprentissage du TI.

Seuillage par hystérésis :
·*Paramètres :* donnée `D`, seuils `b` et `h`, voisinage `v`
·*Expression :*
`Hyst(D,h,b,v) = Ou( Rec(Db,Dh,v) , Dh )`
  avec `Dh = Sll ( D , h , max(`$\Delta_{D.\sigma}$`) )`
  et `Db = Sll ( D , b , max(`$\Delta_{D.\sigma}$`) )`
·*Schéma :*
`Hyst(D,h,b,v)=Ou(Rec(Sll(D,b,`$\Delta^+_{D.\sigma}$`),Sll(D,h,`$\Delta^+_{D.\sigma}$`),v),`
`Sll(D,h,`$\Delta^+_{D.\sigma}$`))`

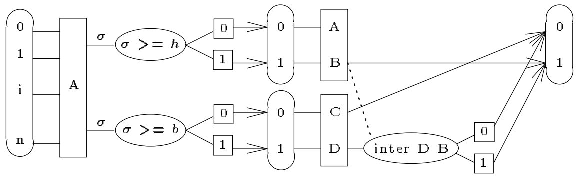


FIG. 4 - *Composition de l'opérateur d'hysteresis*

Un formalisme d'expression.
Pour décrire les concepts élémentaires manipulés lors de ces transformations d'information, pour exprimer de façon rigoureuse les champs des données et pour formuler des objectifs en utilisant des termes propres au domaine de l'application, il nous a fallu développer un formalisme d'expression. Il s'exprime à partir des descripteurs utilisés pour décrire les opérateurs. C'est ce qui va permettre de comparer les expressions du but et des données produites par une chaîne de traitement. Ce formalisme nous a conduit à proposer une hiérarchie de trois langages imbriqués (fig.5).

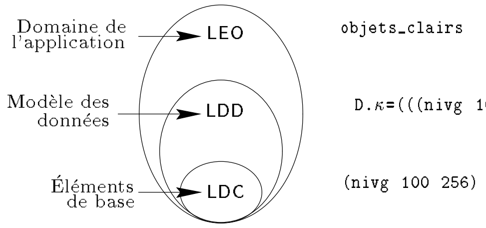


FIG. 5 - *Hiérarchie de trois langages*

• Le LDC, langage de description de concepts. Le Langage de Description de Concepts est constitué de huit catégories syntaxiques définissant les éléments de base de cette hiérarchie de langages. *Les sélecteurs* sont des éléments atomiques ou complexes par lesquels sont exprimées les mesures à effectuer sur les données. *Les mesures* désignent toute entité numérique fournie directement ou issue d'une réelle mesure sur les données. *Les fonctions sur les mesures* permettent d'exprimer les différents calculs arithmétiques utilisés lors des descriptions. *Les domaines* correspondent à des ensembles

au cardinal pair de mesures définissant des unions d'intervalles semi-ouverts. *Les propriétés* définissent les prédicats de partition. Elles sont de ce fait constituées d'un sélecteur et d'un domaine. *Les concepts* sont des descriptions d'entités élémentaires présentes dans les données. Ils sont constitués d'accumulations de propriétés plus ou moins complexes, de relations, ou de descriptions d'entités géométriques. C'est à cause de ces caractéristiques que nous avons donné son nom au langage. *Les bases* sont construites sur des disjonctions de concepts Elles définissent les éléments de plus haut niveau d'expression. *Les bases déstructurées* correspondent à des modifications des bases sur lesquelles elles s'appliquent en mettant en exergue la façon d'examiner les sous-structures décrites par ces mêmes bases.
• Le LDD, langage de description de données. Du fait qu'il manipule des entités conceptuellement différentes, le langage LDC est d'un niveau plus abstrait. Son rôle en tant que langage de base devient plus concret dès la prise en compte du modèle des données. Nous allons donc construire le Langage de Description de Données à partir du LDC. Ce nouveau langage va utiliser toutes les catégories syntaxiques du LDC pour définir au plus haut niveau d'expression cinq nouvelles catégories syntaxiques. Ces catégories définissent ainsi les syntaxes de description des cinq champs de la structure du modèle des données.
• LEO, le langage d'expression d'objectifs. L'entité produite par un graphe de résolution de problème étant une donnée de TI, nous considérons l'expression des objectifs comme la description d'une donnée particulière. Le Langage d'Expression d'Objectif est donc basé sur le LDD dont il va utiliser toutes les caractéristiques. De plus, il intègre la possibilité d'exploiter les termes du domaine de l'application. Pour cela il met en oeuvre des règles de transformation permettant de faire le lien entre lui-même et le LDD. Ces règles utilisent des outils de manipulations permettant de conserver la cohérence des expressions ainsi développées. Ainsi, LEO peut être utilisé comme le langage des descriptions d'objectifs fournies au système de résolution par une Interface Homme-Machine.

Grâce à ces modèles et cette hiérarchie de langages nous possédons un ensemble cohérent d'outils pour la résolution de problèmes par reconstruction des concepts manipulés dans le domaine de l'application. Cet ensemble est formé de quadruplets -(*nom*, *objectif*, *plan*, *programme*)- liés au domaine d'application et est appelé vocabulaire exécutable de cette application. Il est ainsi constitué des concepts du domaine, de leur description en tant qu'objectifs exprimés en LEO, des graphes d'opérateurs symboliques résolvant le problème et du programme exécutable d'opérateurs réels de TI instanciant ce graphe. Chaque élément de ce vocabulaire établit donc un pont entre les concepts manipulées par l'expert et le programme directement exécutable.

4 Approche par les données : Conception du planifieur

4.1 Planification : mécanismes de base

Le planifieur est basé sur un mécanisme de propagation qui rend compte, dans l'expression de sortie, des transformations effectuées sur les données représentées par l'expression d'entrée. Ce mécanisme permet donc de rendre compte du traitement réalisé par un enchaînement d'opérateurs sur les données depuis la donnée initiale. Pour effectuer la fonction inverse (configurer cet enchaînement), le planifieur possède les quatre fonctions suivantes :
.1. acquisition de l'expression d'un but à atteindre et, par l'utilisation de règles transformationnelles, traduction d'une expression de haut-niveau à une expression sous la forme canonique de LEO, dont l'existence a été montrée dans [9],
.2. comparaison de la situation courante à ce but. D'une part, il s'agit de détecter l'égalité de l'expression de l'objectif et celle, construite, du but atteint. Ceci est possible grâce à l'existence d'une forme canonique d'expression du but et à un mécanisme de simplification des expressions décrivant les données produites. Si les deux expressions -état courant et objectif- sont différentes, la progression vers le but s'effectue selon deux axes : l'axe généralisation-spécialisation *(relation est-un)* par le procédé d'accumulation des propriétés est utilisé quand le concept isolé dans l'image et correspondant à l'expression courante est encore trop général par rapport à l'objectif. L'axe composition structurelle *(relation tout-partie)* par la combinaison logique d'expressions correspond au cas où le concept courant est un élément ou un sur-ensemble de l'objectif.
.3. choix de l'opérateur pertinent pour réaliser la progression souhaitée. Cela consiste tout d'abord à sélectionner les opérateurs applicables ; ce sont ceux qui référencent des descripteurs figurant dans l'expression du but et ont des types de données d'entrée cohérents avec le type de la donnée courante. Il faut ensuite instancier les opérateurs sur les données. Cette phase permet de réduire les opérateurs candidats en introduisant deux types d'élimination d'opérateurs : élimination par vérification de cohérence des domaines et par contrôle des domaines de validité de ces descripteurs et élimination experte par application de règles logiques ou de règles expertes en TI sur l'utilisation

des opérateurs. Enfin, le planifieur sélectionne un seul des opérateurs instanciés ou conserve l'ensemble des opérateurs applicables *(recherche d'une ou de toutes les solutions)*.

.4. application l'opérateur. Le planifieur génère l'expression symbolique qui serait produite par l'application de l'opérateur applicable, teste cette expression (comparaison à celle du but) et, si elle est prometteuse, c'est-à-dire si elle contribue à la résolution de l'objectif et si elle n'a pas déjà été rencontrée, l'opérateur est inclus dans le plan et fournit la nouvelle expression courante.

Enfin, après planification de ce graphe d'opérateurs symboliques, le système génère un script, qui est une traduction du plan en termes d'opérateurs réels d'une bibliothèque de TI. Ce script peut alors être exécuté et fournir des données réelles, soit en phase de conception, lors de la formulation incrémentale de l'objectif par le concepteur à l'aide d'un environnement IHM, soit en phase d'exécution ; il constitue le système dédié recherché. Pour une application réelle et donc nécessairement adaptative, cette traduction *plan* $\Rightarrow$ *script* doit intégrer un processus de génération de points de contrôle d'exécution et de mécanismes de réparation. Ce travail est en cours d'étude.

4.2 Mécanismes évolués

Vocabulaire exécutable.
Le formalisme que nous proposons cherche à exploiter au maximum les connaissances de l'expert dans son domaine en lui permettant d'exprimer ses objectifs non plus en termes d'images à traiter, mais en termes d'indices visuels, c'est à dire en termes d'éléments spécifiques à extraire de ces images. L'expert peut alors étiqueter ces objectifs par un terme générique ou propre à son domaine. Il peut ainsi constituer une bibliothèque de plans, associés à des objectifs qui sont nommés: notion de vocabulaire exécutable de TI (`ObjetsAllongés`) ou spécifique à un domaine d'application (`Artères`). Ce vocabulaire facilite l'expression de nouveaux buts en permettant à l'utilisateur de se placer à un plus haut niveau d'abstraction. Grâce à l'utilisation de règles transformationnelles, le système sait inclure la description associée à ces termes dans des descriptions d'objectifs, en manipulent les concepts correspondants -mécanisme d'expansion.

Réutilisation de plans.
La notion de vocabulaire exécutable est la base d'un important mécanisme du module de planification d'ExTI mettant en oeuvre une réutilisation de plans déjà produits et identifiés en tenant compte des entités déjà générées et validées. La planification des objectifs ne peut être menée avec des méthodes classiques de planification non-linéaire. En effet, la décomposition en sous-objectifs des expressions étendues est difficile voire impossible à effectuer, si nous voulons conserver la représentation sous forme de transformations d'information des opérateurs de la base. Ceux-ci, décrivant des réalités du TI, génèrent des données dont les descriptions symboliques ne sont pas toujours décomposables en plusieurs sous-objectifs. La réutilisation de plan décompose l'objectif à atteindre à l'aide des sous-objectifs exprimés. Contrairement à des systèmes de planification tels TWEAK[2], il ne s'agit pas ici de planification par conjonction de buts mais de fusion de plans déjà développés. Si l'expression du but à résoudre comprend un de ces noms, on peut associer directement le plan à ce nom (rapidité de planification) et remplacer ce nom par son expression, ce qui rend possible des simplifications d'expression, et donc l'obtention de plans plus simples. Inversement, à partir d'une expression développée, on peut, par appariement, identifier des sous-expressions déjà rencontrées et leur associer des noms et donc des plans qui peuvent être déjà implémentés, ce qui introduit le mécanisme de macro-opérateurs.

Lorsque le système de planification fonctionne en mode réutilisation, cinq processus correspondant à des phases distinctes de traitement sont mis en œuvre. Afin d'illustrer complètement cette décomposition en plusieurs phases, nous allons prendre un exemple générique de planification d'un objectif O. Cet objectif est décrit à l'aide de trois sous-objectifs étiquetés O_1, O_2 et O_3 ; les deux premiers eux-mêmes décrits en utilisant respectivement les sous-objectifs étiquetés A et B. Les sous-objectifs O_1 et O_2 possèdent des plans que le système va chercher à réutiliser.

•*Phase d'initialisation*
Pour un objectif donné, il s'agit de déterminer, avant expansion des sous-objectifs étiquetés, quels sont ceux pour lesquels un plan a été développé et validé. La récupération des étiquettes est effectué au cours de l'analyse syntaxique de l'objectif principal, qui, dans un même temps, effectue l'expansion. Il suffit alors d'éliminer de cette liste des sous-objectifs utilisés, les étiquettes pour lesquelles il n'existe pas de plan ; on récupère donc la partie réutilisable. Ceci est effectué par une simple recherche dans la base des plans.

•*Phase 0 de régénération de plans.* Cette phase consiste à développer les plans à réutiliser -O_1 et O_2- en fusionnant leurs parties communes (fig.6). Cette phase est effectuée en utilisant le module de planification simple en tant que processus de création de plan, les étapes de filtrage et d'appariement de description n'étant pas déclenchées.

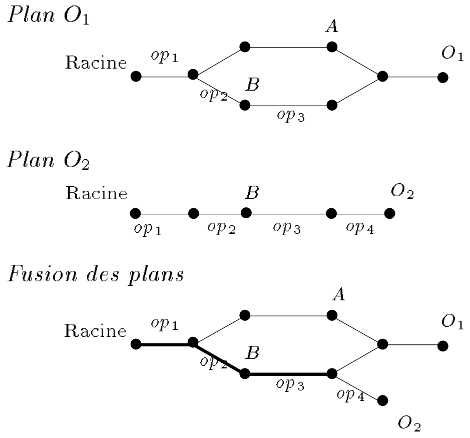


FIG. 6 - *Plans de O_1 et O_2 et fusion des plans*

•*Phase 1 de planification sur les plans générés.* Le processus de planification de la résolution de O est déclenché sur les données résultats, produites lors de la phase précédente, qui résolvent les sous-objectifs (ici O_1 et O_2). Cette phase 1 a pour effet de tenter de résoudre O en allongeant simplement le graphe de résolution. Si l'objectif n'est pas encore atteint, on obtient un graphe tel que celui décrit dans la figure 7, et on poursuit le processus par la phase suivante.

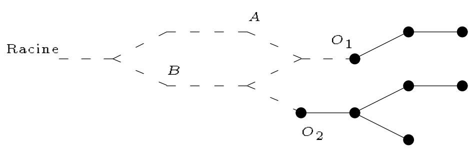


FIG. 7 - *Planification sur les plans générés*

•*Phase 2 de planification sur les sous-objectifs étiquetés.* Cette phase consiste à déclencher le processus de planification sur les sous-objectifs étiquetés reconnus (ici A et B -fig.8-) lors de la phase de reconstruction des plans (ici O_1 et O_2) et non à partir de la racine. Ceci a pour effet de privilégier les données représentant des concepts du domaine de l'application.

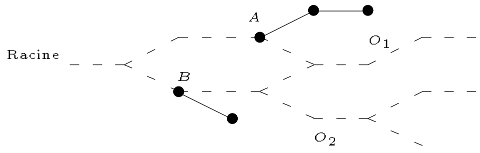


FIG. 8 - *Planification sur les sous-objectifs étiquetés*

•*Phase 3 de planification sur les sous-objectifs non étiquetés.* Cette phase (fig.9) consiste à tenter de résoudre l'objectif principal en déclenchant le processus de planification simple sur les données intermédiaires, produites lors de la phase 0 et ne correspondant pas à un sous-sous-objectif étiqueté.

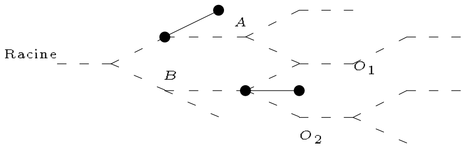


FIG. 9 - *Planification sur les sous-objectifs non étiquetés*

•*Phase 4 de planification sur la racine.* Si l'objectif O n'est toujours pas atteint, on planifie la résolution à partir de la racine (fig.10). Il s'agit donc d'une planification simple, avec cependant un graphe de résolution partiel, déjà constitué.

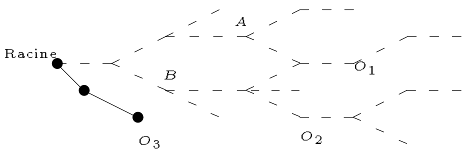


FIG. 10 - *Étape intermédiaire de planification sur la racine*

L'obtention du résultat est garanti par cette dernière phase correspondant, dans le cas le plus extrême à une planification sans réutilisation du graphe de résolution partiel. Cependant, on aura auparavant -phases 2 et 3- cherché à exploiter tous les nœuds intermédiaires, ce qui garantit l'optimalité du résultat.

Base minimale.
La notion de base minimale a été introduite lors de l'étude des performances et du potentiel de résolution du planifieur de ExTI. La décomposition rigoureuse et fine des opérateurs nous a conduit en effet à émettre l'hypothèse qu'il existait une base minimale d'opérateurs symboliques pour notre système. L'existence d'une telle base est difficile à démontrer et à concevoir matériellement. Sachant qu'un opérateur complexe a la même description que celle résultant de la décomposition en opérateurs élémentaires, on n'a aucun intérêt, du point de vue de la planification, à l'introduire dans la base des opérateurs.

Macro-opérateurs.
Si des opérateurs ne sont pas introduits dans la base d'opérateurs symboliques, ils doivent cependant figurer dans la bibliothèque d'opérateurs réels de TI. Ils se-

ront utilisés pour écrire un script correct et performant d'opérateurs exécutables. Le problème qui se pose est de déterminer la séquence d'opérateurs du plan à remplacer et à traduire par le "*macro-opérateur*". Il s'agit donc d'un problème de reconnaissance de plans relativement délicat étant donné la non unicité, pour un même macro-opérateur,des séquences à reconnaître et l'imbrication sur plusieurs niveaux de ces macro-opérateurs. Ainsi, l'ouverture morphologique est décrite comme une combinaison des opérateurs de dilatation et d'érosion, eux-même considérés comme des macro-opérateurs ; l'expression de la transformation par l'opérateur d'érosion intervenant même dans la description de la dilatation. Quant au remplacement de ces séquences, on doit traiter le cas où la chaîne remplacée par un macro-opérateur contient des nœuds, points de départ de sous-graphes (fig.11).

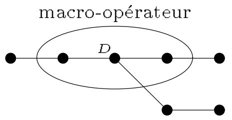


FIG. 11 - *D, point de départ d'un sous-graphe*

5 Analyse et intérêt de l'approche

5.1 Solutions aux problèmes classiques de planification

La démarche utilisée et la possibilité de manipuler formellement l'expression d'un but ou d'une donnée de TI permettent de résoudre plusieurs problèmes classiques rencontrés en planification :

.1. expression du but : grâce à la représentation des concepts en termes de descripteurs de TI, nous avons pu [8] définir un système d'explicitation des objectifs basé sur les interactions avec un utilisateur intervenant dans différents espaces de représentation des concepts en cours de définition,

.2. l'explosion combinatoire : elle est évitée en introduisant plusieurs niveaux de filtrage des opérateurs (§ 4.1),

.3. échec et branches infinies : nous utilisons un même formalisme pour exprimer le but et décrire l'état courant. Nous pouvons donc facilement détecter si un nœud est plus spécialisé que l'objectif, ce qui signifierait qu'il est sur une branche ne menant pas à la solution.

.4. les boucles : un mécanisme vérifie que la situation courante n'a pas déjà été rencontrée. D'autre part, grâce à la forme canonique, un même concept ne peut pas avoir deux formulations différentes. On ne peut donc produire de boucles de parcours.

5.2 Propriétés

Les modèles.
Par cette analyse globale des entités du TI, nous avons représenté analytiquement les données de TI -qu'elles soient de dimension 1 comme un vecteur de mesure, de dimension 2 comme une simple image monochrome, de dimension 3 comme une image couleur ou une paire stéréo.- et les transformations d'information effectuées par les opérateurs. Les modèles que nous proposons rendent compte de la réalité du TI en fournissant à l'intérieur d'une description de donnée l'*historique* -ou trace- des traitements successifs. Ils sont très précis et permettent de représenter, en termes de données, les concepts du domaine d'application.

Appariement.
La rigueur dans la définition d'un formalisme d'expression des entités manipulées est absolument nécessaire pour une planification basée sur les données. Ceci impose de concevoir un module robuste de reconnaissance de ces expressions. Il faut toutefois noter que ce module d'appariement est complètement déconnecté du planifieur : il n'est pas *ad hoc*, mais basé sur le formalisme; il peut donc être écrit directement à partir de la grammaire. Ce travail est actuellement en cours de réalisation.

Réutilisation de plans.
Il ne s'agit en aucun cas de reconnaissance de plan, mais, par l'intermédiaire d'expressions d'objectifs, de fusionner les plans associés aux étiquettes de concepts du domaine d'application.

Macro-opérateurs.
Dans le système ExTI, l'étude de la prise en compte des macro-opérateurs pour la génération du script n'est pas encore terminée. Leur intérêt est pourtant considérable car ils permettent d'utiliser des opérateurs exécutables existant dans une bibliothèque, et probablement optimisés du point de vue algorithmique, au lieu d'exécuter leur décomposition en opérateurs élémentaires. Mais le problème de leur configuration dans le système est relatif à l'introduction même de ces macro-opérateurs dans le plan, interdisant certaines simplifications d'expression du fait de la non unicité de la chaîne de substitution.

Stratégies.
Cette approche est complémentaire des autres approches de planification basées sur une expertise d'utilisation des opérateurs, de leur sélection ou de leur mise en œuvre. Elle n'empêche pas d'ajouter cette expertise si celle-ci est disponible. Elle permet au contraire d'exprimer des stratégies à différents niveaux :

.1. Résolution de problème. Au cours de la mise au point de l'application, ou pour son adaptation en cours d'exécution, on peut vouloir généraliser l'expression d'un objectif -recherche d'objets manquants. Plusieurs stratégies sont possibles. Si l'expression de l'objectif est :

`((desc1 dom1) (desc2 dom2) DescripConcept)`

où un concept de base est spécialisé par deux propriétés exprimées par des couples `(descripteur domaine)`, la généralisation peut se faire par :

· extension du domaine d'un descripteur : `dom1` $\Rightarrow$ `dom1'`

`((desc1 dom1') (desc2 dom2) DescripConcept)`

tel que `dom1` $\subset$ `dom1'`

· suppression de propriétés :

`((desc2 dom2) DescripConcept)`

· changement d'un descripteur : `desc1` $\Rightarrow$ `desc1'`

`((desc1' dom1) (desc2 dom2) DescripConcept)`

tel que `desc1'` généralise `desc1`

Le dernier point suppose une structuration de l'espace des descripteurs (ou de sous-espaces) en niveaux d'abstraction. Ceci fait actuellement l'objet d'une recherche.

.2. Planification. La description des opérateurs par leur action sur les différents champs de la donnée permet de formuler des nouvelles stratégies de sélection de ces opérateurs.

.3. Exécution et adaptation.

· L'étude des macro-opérateurs introduit des nouvelles stratégies de génération du script d'opérateurs réels à partir du plan d'opérateurs symboliques.

· La description des concepts à l'aide de descripteurs et de leur domaine fournit de nouveaux moyens d'implanter les règles d'ajustement de paramètres.

6 Conclusion

Les environnements de planification de chaînes de traitement d'image et les systèmes de pilotage d'algorithmes utilisent des heuristiques basées sur une expertise du traitement d'image et sur les connaissances du domaine d'application. Ces règles expertes et ces heuristiques doivent pouvoir s'appuyer sur des formalismes bas niveau qui représentent avec précision les données de traitement d'image et rendent compte fidèlement des transformations effectuées par les opérateurs. Nous avons proposé un formalisme basé sur les données et montré comment il peut être exploité pour représenter les objectifs de traitement d'image, pour planifier des chaînes et les réutiliser dans des plans plus complexes et pour construire des mécanismes d'adaptation et de contrôle d'exécution des plans générés.

Références

[1] First international workshop on Knowledge-Based systems for the (re)Use of Program libraries. INRIA - Sophia Antipolis.

[2] D. Chapman. – Planing for conjunctive goals. *Artificial Intelligence*, no32, 1987.

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

[4] J.–M. Inglebert. – Un langage de description de concepts de traitement d'images. *In : 10e RFIA, Rennes (France).*

[5] J. Van Den ELST. – *Modélisation de connaissances pour le pilotage de programmes de traitement d'images.* – Thèse de PhD, Université de Nice Sophia-Antipolis (France), 1996.

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

[7] M. Thonnat, V. Clement et J. van den Elst. – Supervision of Perception Tasks for Autonomous Systems : the OCAPI Approach, juin 1993.

[8] O. Capdevielle. – *Formulation d'objectifs de traitement d'images : Une exploitation interactive d'un système de planification automatique de chaînes d'opérateurs.* – Thèse de PhD, Université P. Sabatier de Toulouse (France), 1995.

[9] Ph. Dejean. – *Un formalisme pour les entités du traitement et de l'analyse des images.* – Thèse de PhD, Université P. Sabatier de Toulouse (France), 1996.

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

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

[12] Ph. Dejean et P. Dalle. – Un langage de description de concepts pour la formulation d'objectifs d'analyse. *In : Journées ORASIS'96 – Pôle vision du GDR-PRC 'CHM', Clermont-Ferrand (France)*, pp. 219–224.

[13] R. Bodington. – A software environment for the automatic configuration of inspection systems. *In : Proceedings of the International Workshop on Knowledge Based systems for the (re)Use of Program librairies – Sophia Antipolis*, pp. 99–108.

[14] 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.

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

[16] 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.