



HAL
open science

Planification d'opérateurs de traitement d'images selon un processus incrémental et opportuniste

Régis Clouard, Christine Porquet, Marinette Revenu, Abderrahim Elmoataz

► **To cite this version:**

Régis Clouard, Christine Porquet, Marinette Revenu, Abderrahim Elmoataz. Planification d'opérateurs de traitement d'images selon un processus incrémental et opportuniste. AGI 94, 1994, Poitiers, France. pp.259-262. hal-00981654

HAL Id: hal-00981654

<https://hal.science/hal-00981654>

Submitted on 22 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PLANIFICATION D'OPÉRATEURS DE TRAITEMENT D'IMAGES SELON UN PROCESSUS INCRÉMENTAL ET OPPORTUNISTE*

Régis Clouard, Christine Porquet, Marinette Revenu, Abderrahim Elmoataz

LAIAC - ISMRA, 6 Boulevard du Maréchal Juin, 14050 Caen Cedex

e-mail : Régis.Clouard@L2i.ismra.fr

INTRODUCTION

Nous proposons de concevoir des applications de Traitement d'Images par planification dynamique d'opérateurs, en nous fondant sur le principe de l'intégration sémantique de ces opérateurs [1].

Il nous faut postuler au préalable, que l'on peut représenter n'importe quelle application de Traitement d'Images sous la forme d'un graphe d'opérateurs paramétrés. Un graphe d'opérateurs est une chaîne de traitements destinée à produire, à partir d'images présentées en entrée, de nouvelles images ou des attributs de caractérisation qui mettent en évidence l'information désirée. Un opérateur est ici un programme exécutable contenu dans une bibliothèque. Il prend en entrée des images pour produire en sortie de nouvelles images ou des attributs de caractérisation. Il dispose de paramètres qui permettent d'adapter son comportement à la nature des objectifs et aux caractéristiques des images. Dans un graphe, les opérateurs sont enchaînés de telle façon que les sorties de l'un servent d'entrée pour ceux que lui succèdent. Les résultats sont construits par exécution successive des opérateurs.

L'intégration sémantique des opérateurs s'oppose à l'intégration procédurale qui consiste à développer entièrement un programme spécifique de l'application traitée, en redéfinissant si besoin les opérateurs pour les adapter aux particularités de l'application. Dans cette approche les opérateurs ne sont pas représentés, c'est-à-dire que l'on ne dispose pas de connaissance explicite sur eux. La connaissance est entièrement codée dans le programme, et donc non directement accessible.

Par contre, l'intégration sémantique consiste à coder les opérateurs une fois pour toutes puis à les réifier. Selon cette approche, traiter une image revient alors à savoir sélectionner, paramétrer et enchaîner des opérateurs d'une bibliothèque prédéfinie, pour construire le graphe de

traitements idoïne. Notre but est de réaliser un système qui soit capable de construire automatiquement de tels graphes d'opérateurs à partir de la donnée d'une image et de spécifications fournies interactivement par un utilisateur.

1. SPÉCIFICATIONS D'UNE APPLICATION

Dans ce cadre, une application de Traitement d'Images est définie par une requête portant sur une image dans un contexte déterminé.

La requête décrit l'ensemble des objectifs à atteindre. Elle s'exprime en termes de tâches à accomplir et de contraintes à respecter. Les tâches font référence aux objectifs généraux du Traitement d'Images (*amélioration, restauration, segmentation, détection de formes*). Les contraintes définissent les exigences de qualité sur les résultats : le niveau de détail (*localisation des contours précise, sur-segmentation des régions*), des restrictions (*respecter les contours francs*) ou des critères à optimiser (*critère d'homogénéité en niveau de gris*).

Le contexte doit permettre de définir un sujet pour l'image en précisant les notions d'informations pertinentes pour l'application traitée. Trois niveaux de description sont utilisés pour décrire le contexte. La description physique renseigné sur la physique de formation de l'image (*type de capteur, condition d'acquisition*), la description perceptive sur les indices visuels présents dans l'image (*type des contours, nature des régions*) et la description sémantique sur la notion d'objet pour l'application considérée (*forme, taille, relation entre les objets*).

Nous proposons à l'utilisateur un langage d'expression formel [2] fondé sur la description symbolique des concepts et des propriétés du Traitement d'Images pour exprimer la requête et décrire le contexte. La requête est une conjonction de tâches choisies parmi une liste prédéfinie, puis paramétrées par des propriétés et des concepts du Traitement d'Images. Le contexte est une liste d'attributs-valeurs choisis eux aussi à partir d'une liste prédéfinie.

* Ce travail a été réalisé dans le cadre du pôle « Traitement et Analyse d'Images » de Caen.

Par exemple, la requête "Isoler les objets de convexité supérieure à 80 avec une localisation précise des contours" s'écrit de façon formelle :

But : (Isoler objet (convexité > 80))
Contraintes : (contours précis)

Le contexte de l'application précise entre autre que l'image provient d'un microscope, qu'elle présente un fond clair sur lequel reposent des objets, dont la texture est homogène, et répartis de façon uniforme.

Contexte : (Type-Scène microscopique)
(Présence-Fond vrai)
(Couleur-Fond claire)
(Aspect-Objet homogène)
(Répartition dissociés)

L'utilisateur doit être un expert en Traitement d'Images qui maîtrise le langage d'expression proposé. En effet, la formulation d'une requête exige de lui qu'il sache traduire ses intentions relativement à un domaine d'application (*par exemple extraire les noyaux de cellules épithéliales* dans le cadre d'une application biomédicale), en sélectionnant les bonnes tâches et les bons indices visuels [3].

2. MODÈLE DE PLANIFICATION

La planification consiste à déterminer la séquence d'opérateurs à exécuter successivement sur l'image pour accomplir l'ensemble des tâches de la requête. Notre modèle de planification adopte le principe de décomposition hiérarchique pour générer ce graphe d'opérateurs. Cinq niveaux d'abstraction sont utilisés pour passer de la requête à la détermination des opérateurs. A chaque niveau, il faut définir une version du plan sous la forme d'un enchaînement d'actions. La construction se fait du haut vers le bas, en prenant en compte à chaque étape des contraintes de plus en plus techniques et spécifiques. Nous distinguons par ordre décroissant les niveaux Requête, Tâche, Fonctionnalité, Procédure et enfin Opérateur.

- Une **Requête** est la définition brute du problème formulé par l'utilisateur et se compose de la liste des tâches à accomplir avec leurs contraintes, de l'image initiale et de la description du contexte.
- Une **Tâche** se conçoit comme un sous-problème élémentaire portant sur des images (ex : *améliorer la qualité de l'image, éliminer le fond, former les objets à partir des régions...*). Ce sous-problème est élémentaire en ce sens qu'il est indécomposable en sous-problèmes plus simples.
- Une **Fonctionnalité** décrit un savoir-faire de Traitement d'Images pour d'atteindre un but particulier,

indépendamment de la façon dont ce but peut être atteint (ex : *détection, localisation des contours, extraction des régions par classification des pixels...*).

- Une **Procédure** représente la connaissance opératoire sur un outil du Traitement d'Images, indépendamment de son implémentation. Elle se rapproche de la notion d'opérateur mais sans qu'elle soit liée à une bibliothèque particulière (ex: *différenciation, seuillage par hystérésis, égalisation d'histogramme...*).
- Un **Opérateur** caractérise un algorithme exécutable de la bibliothèque retenue. Il décrit la façon d'exécuter cet algorithme en précisant notamment le typage des données d'entrée, le nombre, le type et les valeurs autorisées pour chacun des paramètres ainsi que la nature des données de sorties.

La planification se fait par affinage successif d'une action d'un niveau en un enchaînement d'actions au niveau directement inférieur, en ordonnant séquentiellement ces actions lorsque les images de sortie des unes constituent les images d'entrée des autres ou en les conservant en parallèle sinon. La planification procède en quatre étapes:

① La requête est tout d'abord explicitée sous la forme d'un enchaînement de tâches. Cette décomposition en tâches est unique, elle ne fait que décrire complètement le problème à résoudre par traduction, reformulation et élimination des ambiguïtés en faisant appel si nécessaire à l'utilisateur. Ces tâches constituent autant de sous-problèmes à résoudre pour réaliser l'application.

② Une tâche peut être accomplie par un enchaînement de fonctionnalités. Il faut alors faire le choix d'une démarche, compte tenu des différentes méthodes disponibles pour traiter cette tâche, en fonction des contraintes particulières et des particularités des images et du contexte de l'application. Le plan décrit à ce niveau représente la solution retenue pour résoudre la requête. Sa réalisation est décidée à l'étape suivante.

③ Chaque fonctionnalité est décomposée en un enchaînement de procédures, en faisant le choix des moyens. A ce niveau, le plan définit la séquence des actions primitives à entreprendre.

④ La détermination du graphe d'opérateurs se fait dans la continuité de la planification. Chaque procédure est décomposée en un enchaînement d'opérateurs exécutables, pour lesquels, il faut en outre déterminer la valeur de chacun de leurs paramètres. Comme cette paramétrisation est un problème difficile a priori, on ne décide pas d'une valeur unique mais d'un intervalle de valeurs potentielles.

Le plan est donc représenté par un arbre à cinq niveaux (Figure 1). Chaque nœud interne de l'arbre se décompose en un enchaînement de nœuds au niveau immédiatement inférieur selon des combinaisons d'arcs ET, PUIS, OU. Les arcs PUIS permettent d'ordonner séquentiellement les nœuds entre eux, les arcs ET n'imposent aucun ordre a priori (les nœuds sont indépendants entre eux, même s'ils participent du même objectif) et les arcs OU permettent de conserver les différentes alternatives possibles.

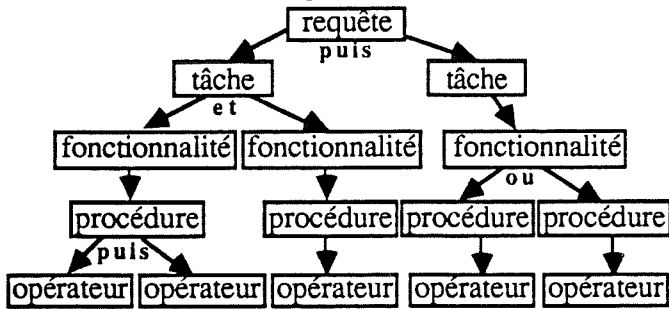


Figure 1 : La structure de plan.

Un nœud correspond à une décision de transformation d'images. Le calcul de la décomposition d'un nœud repose sur l'analyse des transformations d'images le long des différentes chaînes de sous-nœuds possibles. Chacune des images initiale, finale ou intermédiaires doit donc posséder une description symbolique de son contenu, sur laquelle le planificateur va pouvoir raisonner pour déterminer les décompositions les plus adaptées.

3. MODÈLE DE LA CONNAISSANCE

Pour aborder la modélisation du domaine et du processus de planification de façon modulaire, nous avons choisi le modèle du Tableau Noir [4]. Ce modèle permet de définir 2 entités (Figure 2) : une base de données qui contient le plan en cours de formation, sous la forme d'un arbre d'actions, et des Sources de Connaissances qui ont en charge le développement du plan sur cette base de données. Une Source de Connaissances (SC) détient l'expertise nécessaire pour développer une partie du plan total, il faut donc la coopération de plusieurs Sources de Connaissances, par l'intermédiaire de la base de données, pour construire entièrement le plan solution.

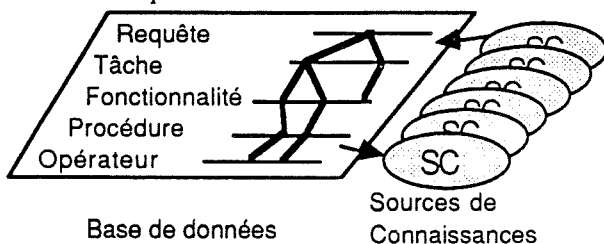


Figure 2 : Le modèle du Tableau Noir.

Un nœud du plan est représenté par les attributs:

NIVEAU - Le niveau d'abstraction du nœud.
BUT - L'action prescrite relativement au niveau d'abstraction.
CONSTRAINTES - La liste des contraintes sur le but.
IMAGES - La liste des images d'entrée concernées par le but.
IMPORTANTANCE, FIABILITE - Une estimation de l'importance et de la fiabilité du nœud dans la représentation du plan.
PREDECESSEURS - La liste des nœuds qui le précèdent dans une séquence de type « puis ».
DECOMPOSITION - La liste des nœuds qui forment la décomposition de ce nœud au niveau inférieur.

Une SC fonctionne selon le schéma condition/action. La condition doit permettre de détecter sur la base de données, un sous-problème que la partie action peut contribuer à résoudre. Une SC est définie par les attributs suivants:

DECLENCHEUR - Une liste de prédicats qui testent l'existence sur le plan, d'un nœud ayant un but déterminé.
PRECONDITION - Une liste de prédicats qui testent l'état actuel de la résolution de manière à en extraire un contexte d'application.
ACTION - Une liste de règles de production.
IMPORTANTANCE, FIABILITE, COÛT-TEMPS - Des fonctions qui permettent d'estimer des valeurs d'importance, de fiabilité et de coût de l'exécution de la SC compte tenu de l'état actuel du plan.

Compte tenu des niveaux d'abstraction adoptés, trois catégories de connaissances doivent être représentées:

- 1 Les connaissances sur les **concepts, le vocabulaire et le contexte** de Traitement d'Images permettent de comprendre la requête formulée par l'utilisateur.
- 2 Les connaissances sur les **méthodes et les outils** de Traitement d'Images codent l'expertise et apportent des solutions à chacune des tâches extraites de la requête. Elles portent sur l'analyse des caractéristiques du contexte et des images pour proposer leur solution.
- 3 Les connaissances **sémantique et syntaxique** sur les **opérateurs** permettent la sélection des opérateurs et garantissent des enchaînements syntaxiquement corrects pour réaliser des procédures.

Ces connaissances se répartissent en 2 types de SC. Les SCs de **description** permettent de calculer les descriptions des images de sortie d'un type de nœud donné à un niveau donné. La condition identifie le type de nœud dont elle peut décrire les sorties. L'action crée les images de sortie et calcule pour chacune d'elles leur description. Une description est construite à partir des listes attribut-valeur des images d'entrée auxquelles on ajoute ou retire des couples attribut-valeur à la façon de Strips ou Noah [5]. Il

existe une SC de description par type de nœud et niveau d'abstraction possibles.

Une SC de décomposition code un savoir-faire pour décomposer un nœud d'un niveau. La condition identifie le type de nœud qu'elle peut décomposer. L'action correspond à la décomposition de ce nœud en un enchaînement de nœuds au niveau inférieur pour lesquels elle doit déterminer leur but, leurs contraintes et la façon de récupérer leurs images d'entrée. Il existe une SC par façon possible de décomposer un même type de nœud et par niveau. Ce sont autant d'alternatives de décomposition d'un nœud.

4. MODÈLE DE RAISONNEMENT

Notre modèle de raisonnement ne suit pas un processus fixe de développement hiérarchique du haut vers le bas et de gauche à droite du plan, au contraire il adopte un comportement incrémental et opportuniste.

La construction du plan est faite par planification incrémentale au travers de l'accroissement et l'agglomération de plans partiels. Le plan est construit pas à pas par intégrations successives d'hypothèses qui tiennent compte de l'état courant du plan. Chaque exécution d'une SC provoque la remise en cause de la liste des SCs à exécuter, par ajout de nouvelles SCs devenues exécutables, retrait de celles devenues caduques et réévaluation des performances en importance et fiabilité de celles qui restent encore exécutables. Le rôle de l'ordonnanceur est de gérer la boucle de résolution. Une étape de résolution est composée de :

① L'exécution de la SC choisie par interprétation des règles de production de sa partie action. Cette exécution provoque des changements sur la base de données.

② La mise à jour d'un agenda contenant les SCs exécutables, en ajoutant les nouvelles et supprimant les caduques.

③ Le choix de la prochaine SC à exécuter.

Le choix d'une SC repose sur un compromis entre les SCs exécutables et les SCs désirables. Les SCs exécutables sont celles de l'agenda. Les SCs désirables sont décrites par le contrôle de la résolution sous la forme de foci d'attention et d'heuristiques. Il s'agit de déterminer à chaque instant ce qui est le plus opportun de faire, compte-tenu de l'état courant de la résolution.

Les heuristiques définissent les préférences intrinsèques sur les caractéristiques des SCs, comme par exemple préférer les SCs les plus fiables, les plus importantes, les moins coûteuses en temps et les plus récemment déclenchées.

Les foci stipulent quel doit être le foyer d'attention du planificateur. Certains implémentent une stratégie générale de type largeur d'abord, qui se traduit par une séquence de foci sur les niveaux hiérarchiques successifs de la base de données, et d'autres réagissent de manière opportuniste à certaines situations comme par exemple préférer les actions qui agissent au niveau Opérateur lorsque cela est possible.

Ce problème du contrôle de la planification est lui-même résolu par un Tableau Noir de type BB1 [6].

CONCLUSION

Les plans ainsi produits vont ensuite être les supports de l'exécution des opérateurs sélectionnés, de l'évaluation de leurs résultats intermédiaires et de la correction des parties de plans non optimales.

Bien que l'utilisateur n'ait plus à connaître les opérateurs, le système exige encore de lui de fortes compétences en Traitement d'Images. C'est un outil d'expérimentation destiné à l'étude et la définition d'applications. Il a notamment été utilisé pour concevoir deux applications : l'une traitant d'images biomédicales et l'autre traitant d'images aériennes [2].

RÉFÉRENCES

- [1] M. Revenu, « Conception d'un atelier logiciel d'intégration de connaissances en traitement et interprétation d'images: un guide méthodologique », *Mémoire d'habilitation à diriger des recherches*, Caen, 1994.
- [2] R. Clouard, « Raisonnement incrémental et opportuniste appliqué à la construction dynamique de plans de Traitement d'Images », *Thèse de doctorat de l'Université de Caen*, 1994.
- [3] P. Dalle, P. Dejean, J.M. Inglebert, « Méthodologie de conception d'applications de traitement d'images », *4èmes journées Orasis du GDR-PRC communication homme/machine*, 1993, pp 92-95.
- [4] B. Hayes-Roth, F. Hayes-Roth, S. Rosenschein, S. Cammarata, « Modeling Planning as an incremental, opportunistic process », *6th IJCAI*, 1979, pp 375-383.
- [5] H. Laruelle, P. Régner, S. Thiébaux, « planification et action », *Bulletin de l'AFIA*, Janvier, 1994.
- [6] B. Hayes-Roth, « A blackboard architecture for control », *Artificial Intelligence*, Vol 26, 1985, pp 251-321.