METHODOLOGIE DE CONCEPTION D'APPLICATIONS DE TRAITEMENT D'IMAGES

DALLE Patrice - DEJEAN Philippe - INGLEBERT Jean Michel

I.R.I.T. - Université Paul SABATIER
118, Route de Narbonne
31062 TOULOUSE Cedex

Mots clés : entités de traitement d'image - modélisation d'opérateurs - langage d'expression d'objectifs - planification - conception d'application - méthodologie.

Introduction

Une application de traitement d'image peut être représentée par un graphe d'opérateurs. Lors de l' exécution de ce graphe, les différents opérateurs effectuent des transformations successives de l'information présente dans les données initiales. L'information recherchée sera alors rendue explicite dans une des structures du résultat.

Notre objectif est de concevoir un système capable de planifier automatiquement ce graphe d'opérateurs à partir de spécifications fournies par un utilisateur ou par un système d'interprétation de plus haut niveau [MAT90].

Quels sont les éléments essentiels que doit comporter un tel système ?

Tout d'abord, on doit être capable de lui fournir une description de l'objectif à atteindre par ces traitements (concept de traitement d'image). Le système doit donc posséder un langage d'expression de ces objectifs. Celui-ci sera basé sur une description symbolique des entités de traitement d'image manipulées par les opérateurs.

On doit donc inclure un modèle de données de traitement d'image. Ce modèle doit considérer à la fois les valeurs de représentation manipulées par les modules informatiques implémentant les opérateurs, et les contenus associés dont la description doit rendre compte des traitements déjà effectués sur les données.

Pour pouvoir planifier des enchaînements d'opérateurs, le système doit être basé sur une modélisation d'opérateurs qui ne décrive pas simplement le résultat fourni par le traitement, mais rende bien compte de la transformation d'information qu'il a effectuée.

A partir de ces modélisations (concepts de traitement d'image, données, opérateurs) le système est alors construit autour d'un module de planification dont le rôle est de combiner les opérateurs symboliques de façon à construire une expression équivalente à celle décrivant l'objectif.

Dans ce cadre, concevoir une application [MAT88] [CLE93] ne consiste plus à déterminer les bonnes fonctionnalités, choisir les opérateurs qui les implémentent et fixer les valeurs de paramètres. Concevoir une application [ING89] revient à savoir formuler correctement l'objectif, c'est-à-dire à être capable de transformer l'expression d'un problème, des termes du domaine d'application en termes de concepts de traitement d'image.

Cependant, s'il n'a plus à connaître les opérateurs de traitement d'image, l'utilisateur doit toujours résoudre le problème, en déterminant les bons indices visuels et en décrivant les entités correspondantes, dans le formalisme proposé. Il peut être aidé, dans cette tâche, par un système d'aide intégrant un modèle de l'utilisateur et comportant un langage visuel [CAP92] et des outils de manipulation de concepts de traitement d'image.

1- Modèle de données de traitement d'image

Les données de traitement d'image sont des structures de représentation associant des valeurs de représentation à des contenus. Les valeurs peuvent être prédéfinies (exemple 0-1 des images binaires) ou représenter des caractéristiques (un signal, une étiquette, une mesure,…) . L'algorithme ne prend en compte que ces valeurs de représentation dans leur aspect numérique (ex. radiométrie) et spatial (géométrie).

A ces valeurs sont associés des contenus qui dépendent des traitements déjà effectués et qui ont produit la donnée traitée.

La description d' une entité de traitement d'image comporte donc différents champs [ING92] :

SM : structures manipulables par les opérateurs applicables à cette entité.
D : description symboliques des différentes structures manipulables.
OA : opérateurs applicables à l'entité.
EA : expressions symboliques accessibles, via les opérateurs applicables.
DA : données accessibles, par l'application de ces opérateurs.

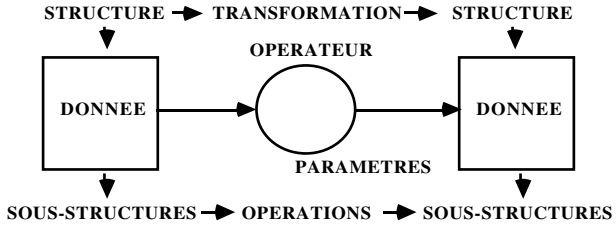
Elle est également structurée suivant les différentes valeurs de séparation de la caractéristique utilisée.

2- Modèle d'opérateur de traitement d'image

Les opérateurs différencient plusieurs sous-structures dans les données qu'ils traitent. Certaines sous-structures sont transmises sans modification dans les structures de la donnée produite. D'autres sont partitionnées, leurs valeurs de représentation sont modifiées et sont recombinées dans différentes sous-structures du résultat.

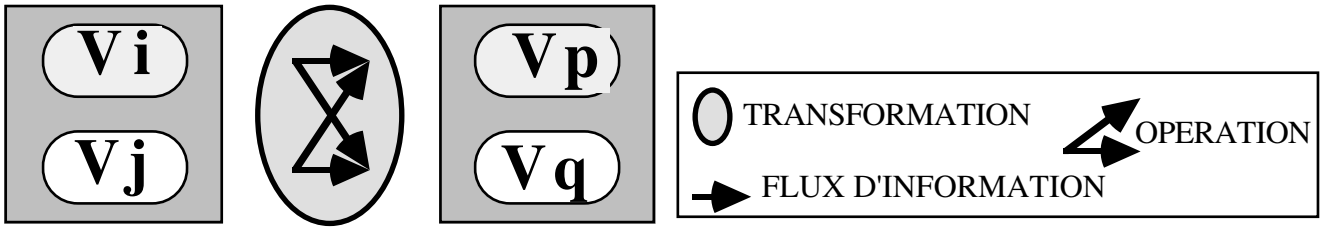
Les éléments qui interviennent dans la desciption de l'opérateur sont donc [DAL91] :

• le flux d'information, c'est à dire :
- la structure globale sur laquelle l'opérateur s'applique,
- les sous-structures qu'il considère et sur lesquelles son action se différencie,
- la structure globale des données produites,
- ses sous-structures associées,
- et les transferts entre structures d'entrée et de sortie.



• la transformation effectuée, c'est à dire l'ensemble des opérations sur les sous-structures :
- la partition des structures d'entrée :
  - le type de partition (directe ou via une mesure),
  - le type de caractéristique utilisée (surface, distance,…)
  - le type d'entité traitée (pixel, composante connexe, …),
  - les valeurs de séparation utilisées (seuil, domaine, …) (V_i , V_j ,…)

- la production des structures de sortie :
  - les valeurs de représentation des données produites,
  - le regroupement des valeurs calculées ou transmises, dans les sous-structures de sortie (V_p , V_q ,…)



DONNEE 1 **OPERATEUR** DONNEE 2

Le modèle d'opérateur permet d'une part de qualifier les concepts de traitement d'image qu'il produit et donc fournit les bases d'un langage de description de ces concepts.

D'autre part, il rend compte fidèlement des transformations d'information opérées, ce qui permet d'en propager les effets les long d'une chaîne de traitement.

3- Langage de description

Il doit permettre à l'utilisateur de spécifier l'objectif à atteindre (description d'un concept de traitement d'image), et au système de décrire les résultats des étapes intermédiaires.

L'élément de base est la propriété d'une entité, qui peut refléter une mesure sur cette entité (ex : surface de composantes connexes) ou une qualification d'un concept (ex : niveau_de_gris > 50). Ces propriétés peuvent être accumulées formant ainsi un vecteur de propriétés qui réalise une spécialisation du concept sur lequel elles s'appliquent.

La spécialisation de concept peut également être exprimée en spécifiant une relation le liant à un ou plusieurs concepts.

La généralisation est réalisée par combinaison de concepts à l'aide de l'opérateur de disjonction.

Enfin, le but exprimé par l'utilisateur peut concerner non seulement l'extraction d'entités ayant certaines caractéristiques mais aussi une évaluation sur ces entités. Le langage permet donc d'exprimer des concepts comme étant une évaluation numérique d'un autre concept.

La grammaire utilisée pour élaborer et contrôler ces buts est développée ci-dessous.

Notation : <C>* = NIL |<C>|<C><C>|<C><C><C>|... où NIL désigne le mot vide.

Types de buts : <but> ::= <ou_de_concepts> | <mesure>
Disjonction de concepts : <ou_de_concepts> ::= (<concept> <concept>*)
Formes de concepts : <concept> ::= <vecteur> | <relation> | <denombrement>
Vecteur de propriétés : <vecteur>::= (<propriété> <propriété>*) |
(<propriété> <propriété>* <ou_de_concept>)
Transformations décrites : mesure, partition
< propriété> ::= (P_ D) | (P_ D CP <valeur>)
Formes de valeurs : <valeur> ::= entier | <courbe> | <mesure>
... avec : <courbe> ::= (C_ D <entier>*)
Relations entre deux concepts : <relation> ::= (R_ D <ou_de_concepts> <ou_de_concepts>)

Dénombrements sur les concepts : <dénombrement> ::= (D_ D <ou_de_concepts>)
Mesures : <mesure>::= (M_ D <ou_de_concepts>) |(A_ D <mesure>)

Les ensembles des descripteurs sont définis comme suit :
Descripteurs de propriétés : P_ D={ coordonnee, max, min, moyenne, niveaux-de-gris, distance-au-bord, périmètre, surface, nb_voisins, ... }
Descripteurs de relations : R_ D={ dans, hors, intersecte, nintersecte, ...}
Descripteurs d'évaluation: D_ D={ nb_zones, valeur_max, valeur_min, 1+, 1-, ...)
Descripteurs de courbes : C_ D={ cercle, fenetre, ... }
Descripteurs de mesures : M_ D={ nb_zones, valeur_max, valeur_min, ... }
Descripteurs arithmétiques : A_ D={ 1+, 1-, ... }
Comparateurs : CP={ >=, <, = }

4- Planification

Ayant une description symbolique de l'objectif et étant capable de décrire chaque résultat intermédiaire, le planifieur peut explorer et retenir toutes les combinaisons d'opérateurs qui fournissent une expression équivalente au but recherché. Le caractère fortement structuré des expressions permet d'éviter une explosion combinatoire, en abandonnant très vite des chemins qui ne mènent pas à l'objectif.

Cependant des recherches sont encore nécessaires pour structurer l'espace des concepts et pour affiner la notion d'équivalence entre expressions. Ceci permettra d'implanter des stratégies de planification.

Notons enfin qu' une étude des graphes d'opérateurs produits offre deux aspects intéressants, outre le chemin solution. D'une part la description des nœuds intermédiaires permet de définir un vocabulaire spécifique du domaine en terme de description d'image, qui pourra faciliter l'expression de nouveaux problèmes dans le même domaine. D'autre part l'analyse des nœuds échecs proches du but permet de spécifier des opérateurs manquants.

5- Méthode de conception

Ces formalismes et ces outils permettent d'envisager une nouvelle approche pour la conception d'une application. A l'aide de ce vocabulaire de description d'image, d'outils graphiques et d'une interface traduisant le formalisme interne en termes accessibles, l'utilisateur fournit une première formulation de son objectif. Celle-ci reflète la façon dont il pense traduire son problème en termes d'indices visuels. Après planification et exécution, il évalue la différence entre les résultats réels et attendus. Il dispose en outre d'une description symbolique de tous les résultats intermédiaires et peut donc juger de la pertinence des indices visuels qu'il a utilisés. A l'aide d'outils de spécialisation et de généralisation, il peut ajuster ces descriptions au cours d'une série de cycles reformulation-résolution-évaluation.

Conclusion

L'introduction, dans un système, d'une modélisation fine des entités et des opérateurs de traitement d'image permet de définir un formalisme d'expression de buts de traitement d'image. Ce formalisme est utilisé d'une part pour planifier la ou les chaînes d'opérateurs qui permettent d'atteindre le but fixé. D'autre part il permet à l'utilisateur de faire appel au traitement d'image sans en connaître les opérateurs et d' exploiter alors sa compétence réelle dans le domaine. Cependant, il reste nécessaire de savoir formuler son problème en termes d'indices visuels.

Bibliographie

[CAP92] Capdevielle O., Dalle P., Gurdjos P., Inglebert JM. Langage visuel pour la formulation d'objectifs de traitement d'images. congrès IHM92 Paris 1992

[CLE93] Clément, V., and Thonnat, M. A knowledge-based approach to integration of image processing procedures. CVGIP: Image Understanding vol. 57 n°2 march 1993

[DAL91] Dalle P. Inglebert JM. Modèle de description d'opérateurs de traitement d'images. 8ème congrès RFIA, Lyon 1991 pp. 257-266

[ING89] Inglebert, JM., and Dalle, P. Conception de systèmes dédiés d'interprétation. 7ème Congrès RFIA Paris 1989 p. 1083-1090 (1989).

[ING92] Inglebert JM. Planification automatique de chaînes de traitement d'images Doctorat UPS Toulouse III septembre 1992

[MAT88] Matsuyama, T. Expert systems for image processing - knowledge-based composition of image analysis processes. ICPR 1988 Rome Italy p. 125-133 (1988).

[MAT90] Matsuyama, T., and Hwang, V. SIGMA : A knowledge-based aerial image understanding system. Plenum Press, 1990.