Comprendre et utiliser le traitement d'images

DEJEAN Ph., DALLE P.

IRIT-2IL/TCI Université Paul SABATIER
118, route de Narbonne
31062 TOULOUSE Cedex
Tél : 61 55 63 09
Fax : 61 55 62 58
e-mail : dejean@irit.irit.fr
dalle@irit.irit.fr

Abstract :

In this paper, we first list the different uses of the image processing and the kinds of help that have been proposed. We try to define the specifications of the functionalities to use, based on the comprehension and the manipulation of the image processing operators. Then, we present the different methods of image processing operator modelling. We show that a modelling focusing on information manipulations carried in the data structures provide a precise formalism. This one may be exploited by several learning or image processing exploitation systems we give the principles. Our representation of the image processing data is based on such a model allowing to account for the operator effects. Thus, the operator representation allows to note the propagation of these transformations along an operator chain.

Comprendre et utiliser le traitement d'images

DEJEAN Ph., DALLE P.

1 Introduction

Les utilisations du traitement d'images se multiplient de façon considérable dans le domaine industriel (robotique, contrôle de qualité, ...), spatial, médical et s'étendent à de nouveaux domaines, avec le multi-média, la vision de dispositifs autonomes, la surveillance, l'accès aux banques de données images, etc ... Ce phénomène a provoqué le développement de recherches de plus en plus complexes, tant au niveau des modèles et méthodes mathématiques que de l'algorithmique. Enfin, de nombreux produits sont maintenant commercialisés, allant de l'application spécialisée à l'environnement de développement. Le traitement d'images est devenu un domaine scientifique difficile à appréhender dans toutes ses dimensions et complexe à utiliser. Il est donc nécessaire de fournir des aides adaptées aux différents types d'utilisateurs du traitement d'images.

Dans cet article, après avoir répertorié les différentes classes d'utilisation du traitement d'images et les types d'aides qui ont été proposées, nous essayons de définir un cahier des charges des fonctionnalités à mettre en œuvre, basées sur la compréhension et la manipulation de l'opérateur de traitement d'images. Nous présentons alors les différentes méthodes de modélisation d'un opérateur de traitement d'images. Nous montrons qu'une modélisation centrée sur les manipulations de l'information véhiculée dans les structures de données procure un formalisme précis. Ce formalisme peut être exploité par différents systèmes d'apprentissage ou d'exploitation du traitement d'images, dont nous donnons les principes.

2 Utilisation du traitement d'images

2.1 Caractéristiques du traitement d'images

Une image numérique est obtenue à partir d'une scène réelle par un processus faisant intervenir l'éclairement de la scène, les objets (leur nature et leur géométrie), un capteur (son système optique et son dispositif de numérisation). Elle contient donc des informations géométriques et radiométriques. Le traitement a pour but d'extraire des informations concernant les objets de la scène correspondant à l'image. Il consiste à construire des entités représentant les concepts recherches, en inversant plus ou moins complètement le processus de formation de l'image. Les opérateurs utilisés implémentent imparfaitement les fonctionnalités du processus de reconstruction, à cause des difficultés introduites par la géométrie discrète, le caractère généralement mal conditionnée du système, la nature non inversible de la projection perspective, le bruit des données et le caractère incertain et imprécis des informations a priori sur la scène. Aussi les bibliothèques de traitement d'images comportent un grand nombre d'opérateurs concurrents pour chaque famille de traitements, et la résolution d'une application particulière peut souvent être abordée de différentes façons difficilement comparables. En outre, ces solutions sont encore le plus souvent élaborées par un processus d'essais-erreurs, l'ingénierie du traitement d'images n'en étant qu'à ses débuts.

2.2 L'utilisateur du traitement d'images

Suivant ses objectifs, un utilisateur peut (dans l'ordre croissant de compétences) :

- Apprendre le traitement d'images, afin de savoir utiliser un logiciel de traitement d'images plus ou moins spécialisé. Il s'agit donc de comprendre les fonctionnalités et les opérateurs du logiciel. Actuellement, la comparaison des opérateurs d'une même famille se fait, en examinant les résultats qu'ils produisent et parfois à l'aide d'outils spécifiques. De même, la compréhension du travail réalisé par un enchaînement d'opérateurs (script) se fait en examinant l'effet produit sur des données.
- Utiliser le traitement d'images. Il s'agit de construire une application (script), en intégrant des opérateurs d'une bibliothèque. Ceci se fait par essais-erreurs en observant les résultats et en effectuant des mesures sur les données produites. Actuellement les systèmes ne fournissent aucune indication sur les causes d'échec : type de fonctionnalité, choix d'opérateur, ajustement de paramètres à remettre en cause.
- Développer de nouveaux opérateurs de traitement d'images. On utilise pour cela un environnement de développement qui fournit des utilitaires de manipulation d'images et des implantations des principaux types de données utilisés en traitement d'images.

2.3 Les environnements de traitement d'images

Outre les systèmes spécialisés, dont nous ne parlerons pas et qui contiennent leurs propres systèmes d'aide, et les langages spécialisés ou étendus pour la programmation du traitement d'images (Olive [Fli90]), on trouve plusieurs familles de produits :

- les systèmes interactifs de traitement d'images, tels que Visilog [Noe92], comprenant une bibliothèque d'opérateurs de traitement et d'analyse d'images, des utilitaires et une couche interactive pour la présentation des opérateurs, leur sélection, leur enchaînement et pour la présentation des résultats. Cet environnement peut comporter un langage visuel (Khoros [Uni92]) qui facilite la manipulation des opérateurs mais pas leur compréhension ni leur sélection.
- les environnements de développement, construits sur un langage facilitant le prototypage (Smalltalk) et dans lequel sont re-déclarés les opérateurs d'une bibliothèque.

Certains de ces environnements intègrent des aides en ligne décrivant les fonctionnalités de chaque opérateur ou illustrant leur effet sur des imagettes.
D'autre part, la recherche sur les aides à la configuration de chaînes de traitement ([Cle93], [Clo93], [Cap94]), à la réutilisation de programmes ou au contrôle d'exécution (KBUP [KBU95]) est en plein développement et vise à faciliter l'exploitation du traitement d'images dans des applications.

3 Aides à la compréhension et à l'utilisation

On peut définir les objectifs et le cahier des charges d'un système d'aide :

1. **Étudier un opérateur** Le système doit rendre compte précisément de l'ensemble des opérations qu'il a effectuées : description des structures de données d'entrée et de sortie, transfert de l'information qu'elles supportent, effet produit dans les images.

2. **Comparer deux (des) opérateurs** Le système doit rendre compte de la différence de traitement et de résultat réalisé par ces opérateurs à partir d'une donnée commune. Cette information doit d'une part permettre de mieux comprendre la spécificité de chaque opérateur et d'autre part, faciliter la sélection du plus pertinent dans un contexte applicatif donné.

3. **Étudier une chaîne d'opérateurs** Une chaîne est un macro-opérateur. On doit pouvoir décomposer un tel opérateur, c'est à dire comprendre les étapes de transformation et de combinaisons qu'il réalise. Inversement, connaissant l'effet de chaque opérateur élémentaire, on doit pouvoir synthétiser leurs traitements individuels et obtenir une compréhension "compilée" de l'opérateur qu'ils composent.

4. **Concevoir une chaîne de traitement** Il s'agit d'aider à la construction itérative d'un graphe d'opérateurs. On doit donc fournir un moyen permettant à l'utilisateur d'établir plus facilement une correspondance entre le concept qu'il veut extraire et dont il imagine une instance visuelle et le traitement qui va transformer les données de telle sorte qu'un champ de la donnée de sortie représente ce concept. Ceci peut être réalisé en exprimant les concepts en termes d'indices visuels et en décrivant symboliquement les opérateurs comme constructeurs de ces indices. Il s'agira alors de faire correspondre les descriptions, le système apportant une aide à la formulation du problème, c'est à dire à la description du concept recherché.

4 Représentation et Description d'un opérateur

D'après l'analyse précédente on voit que le système d'aide va manipuler une représentation des opérateurs de traitement d'images. Cette modélisation doit permettre à la fois de comprendre et de planifier. Elle doit donc rendre compte du traitement effectué, de façon interne sur les données et de façon externe en terme d'effet produit, et permettre de relier un opérateur et un objectif de traitement d'images. Différents modèles de représentation ont été proposés, généralement 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).

Approche "fonctionnelle". Le traitement d'images est décrit en termes de fonctions et les opérateurs sont les outils informatiques qui implémentent ces fonctions. Cette approche est utilisée, par exemple, dans tous les environnement interactifs de traitement d'images ; elle permet de classer les opérateurs.

Décomposition algorithmique. Elle peut être décrite en termes de flux de données et être utilisée pour organiser de manière homogène la description d'une bibliothèque de traitement d'images. Elle permet une meilleure compréhension des algorithmes, l'insertion de nouveaux opérateurs dans la bibliothèque ou le développement d'une nouvelle fonction, par analogie [Ser89].

Présentation des résultats. Ceux-ci sont exprimés en termes de types de données. Cette approche peut être utilisée,

- pour standardiser les représentations de données et construire des environnements d'interprétation d'images, comme dans [IUE94],
- pour définir des langages de traitement d'images, construits sur des types abstraits de données comme dans SAPIN [Bel85], pour faciliter le développement de programmes spécifiques,
- pour planifier des opérateurs, en structurant l'espace des données sous la forme d'un graphe d'entités. Les opérateurs sont alors des processus de transfert entre ces types, comme dans LLVE [Mat90]. Des règles permettent alors de manipuler ces processus de transfert pour construire une segmentation de l'image.

Ils peuvent aussi être exprimés en termes d'effets. On construit alors un vocabulaire décrivant les caractéristiques géométriques extraites ou mises en valeur par les opérateurs [Sch89]. Si un objectif peut être décrit à l'aide de ces effets prédéfinis, des règles de raisonnement permettront alors de sélectionner les opérateurs qui les produisent.

Description des propriétés, en termes mathématiques [Sch92]. Ceci permet de contrôler l'utilisation des opérateurs, en exploitant par exemple des propriétés d'idempotence ou d'associativité, pour contrôler le nombre d'itérations ou les combinaisons d'opérateurs.

Utilisation. Cette information sera exploitée dans un système de contrôle d'exécution tel qu'OCAPI [Cle93]. Pour cela, on établit des liens, lors de la conception d'un système spécialisé, entre objectifs à atteindre et opérateurs résolvant cet objectif. La donnée de traitement d'images est perçue comme un ensemble de caractéristiques mesurables. Des règles expertes évalueront ces caractéristiques et le contexte d'application, pour sélectionner les opérateurs ou ajuster les paramètres [Tho93].

Information traitée. C'est sous ce point de vue que nous avons proposé une nouvelle modélisation [Ing92], car elle permet de bien rendre compte du traitement effectué ce qui est fondamental pour les tâches de planification ou d'explication.

Nous présentons ci-dessous cette modélisation et son utilisation pour l'aide au traitement d'images.

5 Aide à l'utilisation du traitement d'images

5.1 Représentation du traitement d'images

5.1.1 Les modèles utilisés

Le modèle de donnée Le modèle que nous exploitons [Dej96], s'insère dans le processus de descriptions symboliques de données de traitement d'images exprimées en termes d'indices visuels. Ce processus permet ainsi de décrire des objectifs qui seront reconstruits par planification d'opérateurs par un système de résolution de problème. Comme tout vecteur de communication une image peut être considérée comme étant composée de deux éléments :

1. Le support : c'est la grille rectangulaire ou hexagonale en traitement d'image qui va permettre de manipuler et de véhiculer le signal.
2. La représentation de l'information : il peut s'agir d'une fonction mathématique, d'un simple codage, ou de tout autre système de représentation.

Lorsqu'on examine le travail effectué par un expert du domaine au cours d'un processus d'interprétation d'image, on s'aperçoit qu'il occulte, consciemment ou non, certaines parties de cette image. Les parties délaissées et celles qui déterminent une bonne interprétation peuvent être considérées comme des composantes particulières du support, masques ne laissant apparaître que l'information à traiter. Toute donnée de traitement d'images de dimension n sera alors considérée sous la forme d'une structure composée des cinq éléments suivants (fig.1) :

La trame (τ) est définie par l'ensemble des points p de l'image dont les coordonnées sont comprises entre deux bornes :
$\tau^n = \{p^n = (x_1, \cdots, x_i, \cdots, x_n) \;/\forall n \in \mathrm{IN}, \forall i \in [1,n], \exists (I_i, S_i) \in \mathrm{IN}^2, x_i \in [I_i, S_i[\}$

Le cache (κ) possédant une description δ est déterminé par l'ensemble des points p de τ vérifiant la 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 : $\kappa_\delta = \left\{ p \in \tau^n \Big/ f_\delta(p) = \mathsf{True} \right\}$

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

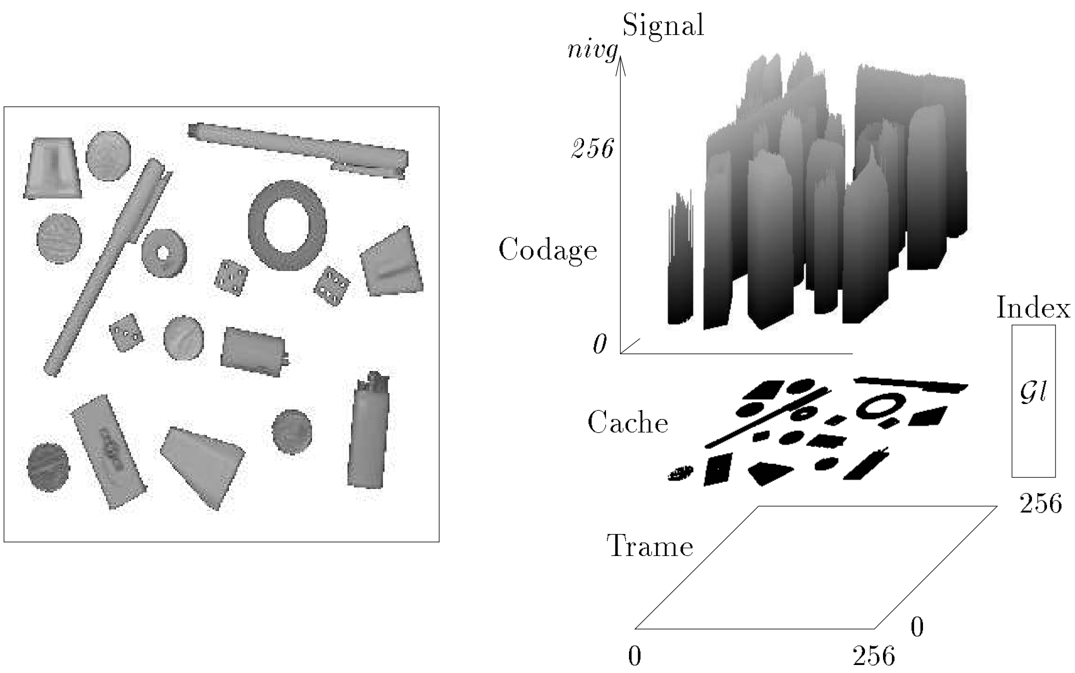

FIG. 1 - *Image et représentation*

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.

Le modèle d'opérateur Le modèle d'opérateurs de traitement d'images [Dal91] décrit la transformation d'informations réalisée et non les conditions d'utilisation et/ou les résultats obtenus. Cette notion de transformation d'informations est également exploitée par Schmitt [Sch89] qui utilise la Morphologie Mathématique et l'Intelligence Artificielle pour développer un système de programmation automatique.

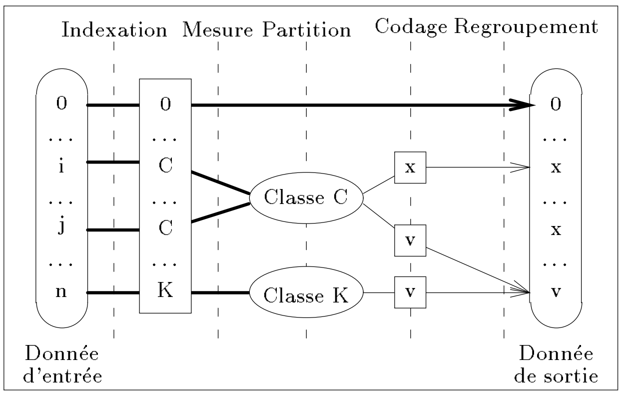

FIG. 2 - *Transformations d'informations*

Dans notre approche, cette transformation est en fait effectuée par un ensemble d'opérations (*indexation*, *partition*, *codage*, *regroupement*, *mesure*) manipulant la structure et les entités présentes dans ces données (fig. 2). Ce modèle est représenté par une combinaison de fonctions de modification d'expressions utilisant les notions de *structuration*, *déstructuration*, *spécialisation*, *généralisation* et *nomination* de concepts. Ainsi un opérateur de traitement d'images sera décrit par un ensemble de cinq combinaisons de fonctions, une pour chaque champs de la donnée. L'intérêt de l'utilisation de ce modèle pour la compréhension et la manipulation du traitement d'images, réside dans la décomposition des divers traitements effectués et dans le formalisme rigoureux représentant l'effet de chaque traitement.

Exemple : Application de l'opérateur de seuillage $Sll(d,m,M)$ sur une donnée image D :
Script KHOROS associé : `vthresh -i D -l m -u (M-1) -o R`

$$Sll(d,m,M) = \left[\begin{array}{l} \tau^t = \mathcal{I}d^\tau \\ \kappa^t = \kappa^-_{\{d.\sigma\in[m,M[\}} \\ \iota^t = \Gamma^- \\ \sigma^t = \sigma_{vab} \\ \omega^t = \omega_{vab} \end{array}\right] \begin{array}{ll} \mathcal{I}d^\tau & : \text{Fonction d'identité de la trame} \\ \kappa^-_{\{d.\sigma\in[m,M[\}} & : \text{Fonction de spécialisation du cache} \\ \Gamma^- & : \text{Fonction de structuration de l'index} \\ \sigma_{vab} & : \text{Fonction de structuration du signal} \\ \omega_{vab} & : \text{Fonction de fixation du codage} \end{array}$$

$$R = \left[\begin{array}{l} \tau^t = \mathcal{I}d^\tau \\ \kappa^t = \kappa^-_{\{d.\sigma\in[m,M[\}} \\ \iota^t = \Gamma^- \\ \sigma^t = \sigma_{vab} \\ \omega^t = \omega_{vab} \end{array}\right] \left(\left\{ \begin{array}{l} \tau = D.\tau \\ \kappa = D.\kappa \\ \iota = D.\iota \\ \sigma = D.\sigma \\ \omega = D.\omega \end{array} \right\} \right) = \left\{ \begin{array}{l} \tau = \mathcal{I}d^\tau(D.\tau) \\ \kappa = \kappa^-_{\{d.\sigma\in[m,M[\}}(D.\kappa) \\ \iota = \Gamma^-(D.\iota) \\ \sigma = \sigma_{vab}(D.\sigma) \\ \omega = \omega_{vab}(D.\omega) \end{array} \right\} = \left\{ \begin{array}{l} \tau = D.\tau \\ \kappa = (D.\sigma\ m\ M) *\ D.\kappa \\ \iota = (\mathcal{G}l\ [1,\cdots,1]) \\ \sigma = vab \\ \omega = (1\ 2) \end{array} \right\}$$

Tous ces opérateurs pourront 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 traitement d'images.

5.1.2 Un langage de description de concepts

Introduction Les transformations d'informations peuvent alors 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)*.
2. Les définition d'une entité, qui comporte : les sélections d'entités de la données 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

Les opérations de quantification d'une entité 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. La sélection d'entités 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.

Afin de manipuler aisément ces notions, de les implanter et de les exploiter lors de la planification, les caractéristiques du calcul des prédicats nous ont semblés bien adaptées. L'utilisation d'une telle représentation ramène donc toutes les notions précédentes -spécialisation, généralisation, etc.- à celle utilisées en intelligence artificielle. Ainsi, une "expression accessible" (description d'une donnée) sera considérée comme une "Well Form Formula" (WFF) composée de termes regroupant les propriétés ou les relations sur d'autres expressions.

Les propriétés $P = [p\ \delta]$ correspondront à une adaptation à notre domaine de la définition des sélecteurs décrits par Michalski [Mic83], [Mic84] qui définit le calcul des prédicats annotés. Elles sont composées d'un descripteur p *(ex : surface, niveau de gris, distance, ...)* et d'un domaine δ, déterminant leurs valeurs associées, constitué d'union d'intervalles semi-ouverts [x,y[. Le domaine de toute propriété P est inclus dans ce que nous nommerons le domaine de validité associé au descripteur p de P. Ce domaine de validité, Δ_p, permet de déterminer la propriété inverse $\neg P$ de P définie comme : $\neg P = [p\ \neg\delta]$ *avec* $\Delta_p = \neg\delta \cup \delta$. Conformément à cette définition de la propriété, la sélection par mise en relation de plusieurs concepts sera formée par des prédicats composés de type *"R(wff1 wff2 ...)"*.

La grammaire complète utilisée pour élaborer et contrôler ces objectifs est développée ci-dessous. Elle est le support d'un analyseur de syntaxe vérifiant et simplifiant la description d'un objectif. Dans le cadre d'une résolution de cet objectif cette simplification permettant d'améliorer les possibilités du module de planification utilisé. Cette grammaire utilise les éléments terminaux suivants :

π	: descripteur de propriété.	χ_π	: élément modificatif de propriété.
υ	: valeur numérique.	μ	: descripteur de mesure.
θ	: descripteur d'opération arithmétique.	σ	: descripteur de signal.
χ_σ	: élément modificatif de signal.	ρ	: descripteur de relation.
ϕ	: descripteur de figure.	ν	: descripteur de jonction.
ι	: descripteur d'index.	δ	: descripteur d'élément déstructurant.

<u>*Six items élémentaires*</u> Chaque élément de la donnée s'exprime en fonction d'items grammaticaux de bases permettant de construire des descriptions de plus en plus complexes syntaxiquement et sémantiquement.

1. Mes : classe de définition d'une **mesure**.

   Seront considérés comme des mesures, les valeurs numériques -υ-, les mesures effectives sur les données -$(\mu|\sigma\ Dst)$ ou $((\chi_\sigma^+\ \sigma)\ Dst)$- et les opérations arithmétiques sur les mesures

-(θ Mes Mes). $\boxed{Mes = \upsilon \mid (\mu|\sigma\ Dst) \mid ((\chi_\sigma^+\ \sigma)\ Dst) \mid (\theta\ Mes\ Mes)}$

2. Dom : classe de définition du **domaine**.

   Un domaine sera défini par une juxtaposition de 2n mesures -$x_1\ y_1 \ldots x_n\ y_n$- et représentera l'union des intervalles semi-ouverts $[x_i, y_i[$. $\boxed{Dom = \{Mes\ Mes\}^+}$

3. Ppt : classe de définition d'une **propriété**.

   Une propriété sera formée d'un descripteur et d'un domaine (voir 5.1.2). $\boxed{Ppt = (\pi\ Dom)|((\chi_\pi\ \pi)\ Dom)}$

4. Cpt : classe de définition d'un **concept**.

   Un concept est la description complète d'un indice visuel de la donnée. Il est constitué d'accumulation de propriétés spécialisant une éventuelle base déstructurée -(Ppt^+)|(Ppt^+ Dst)-, de relation entre des sous-objectifs ou bases, -(ρ Dst Bse^+)-, de descriptions de masques géométriques -(ϕ Mes^+)- ou de jonctions (conjonctions/disjonctions) de sous-objectifs -(ν Bse Bse^+). $\boxed{Cpt = (Ppt^+)|(Ppt^+\ Dst)|(\rho\ Dst\ Bse^+)|(\phi\ Mes^+)|(\nu\ Bse\ Bse^+)}$

5. Bse : classe de définition d'une **base**.

   Une base est formée d'une disjonction non vide de concepts. Elle constitue l'élément principal de la description du cache de la donnée. $\boxed{Bse = (Cpt^+)}$

6. Dst : classe de définition d'une **déstructuration** de base.

   La déstructuration d'une base permet de considérer, l'information attachée aux éléments de niveau inférieur de l'entité manipulée pour la spécialiser par une relation ou une accumulation de propriété. $\boxed{Dst = (\ (\delta)\ Bse)}$

<u>*Les cinq champs*</u> de la donnée vont être décrits en utilisant les items grammaticaux définis ci-dessus :

La trame est représentée par les mesures I_i et S_i de chacune de ses dimensions :

$\boxed{Trm = (\{(Mes\ Mes)\}^*)}$

Le cache peut être global -t- et considère tous les points de la trame, ou est défini par une base.

$\boxed{Cch = t|Bse}$

L'index est défini par un descripteur unique ou faisant référence à un élément déstructurant.

$\boxed{Idx = ()|(\iota)|(\iota\ \delta)}$

Le signal considère plusieurs niveaux de transformation de la donnée : σ et $(\chi_\sigma^+\sigma)$ décrivent le signal d'une donnée de même dimension que la donnée de l'état initial alors que $(\sigma\ Dst\ Bse^*)$ et $((\chi_\sigma^+\sigma)\ Dst\ Bse^*)$ témoignent en plus d'un changement de dimension de la donnée.

$\boxed{Sgn = \sigma|(\chi_\sigma^+\sigma)|(\sigma\ Dst\ Bse^*)|((\chi_\sigma^+\sigma)\ Dst\ Bse^*)}$

Le codage décrit les valeurs que peut prendre le signal. Il est donc formé par une union d'intervalles : le domaine du signal.

$\boxed{Cdg = (Dom)}$

5.1.3 Un planifieur

Possédant des modèles de donnée et d'opérateurs construits sur la même notion d'indices visuels, nous pouvons alors développer un système général de génération de plans permettant de résoudre un problème de traitement d'images dans un domaine d'application définit en utilisant ses concepts et la connaissance de ses experts.

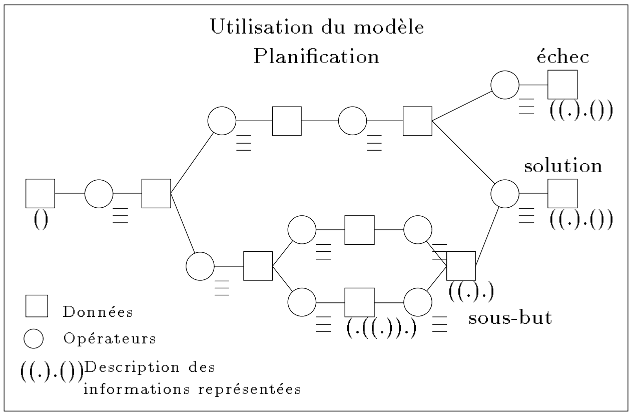

FIG. 3 - *Graphe d'opérateurs*

Dans le but de fournir une aide à la compréhension et l'analyse de la chaîne de traitement d'images il doit aussi permettre d'accéder à l'information liée au graphe d'opérateurs planifié (fig.3). Ces informations pourront alors être exploitées pour développer un langage descriptif du domaine d'application, pour maîtriser l'*historique* du plan pour des ré-utilisations et pour faciliter le contrôle de son exécution. Ce système de planification, ExTI, met en œuvre un module de filtrage permettant de constater la distance qui sépare la donnée produite à l'objectif et en fusionnant les réponses aux filtres sur les cinq champs de la donnée. Le moteur de planification est chargé de construire le graphe d'opérateurs *(arcs)* à appliquer aux données *(nœuds)* pour parvenir à l'objectif visé. De cette façon, un tel graphe, ou 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.

Les spécificités du traitement d'images ont conditionné le choix du planifieur. Dans des approches classiques (monotones, par défaut, STRIPS [Fik71]), les actions manipulent les objets constituant le monde connu dans l'état initial. Par contre, en traitement d'images puisque le monde est construit tout au long de la phase d'exécution de la chaîne. Nous voulions de plus que la description du monde à l'état ε du plan témoigne de l'historique des transformations d'informations opérées pour y parvenir. Nous avons donc développé un planifieur spécifique à notre objectif défini de la façon suivante :

État initial : `Donnée` D_i `et sa représentation symbolique`

$$\mathcal{R}(D_i) = (D_i.\tau, D_i.\kappa, D_i.\iota, D_i.\sigma, D_i.\omega)$$

État final : `Représentation symbolique de l'objectif`

$$\mathcal{R}(D_f) = (D_f.\tau, D_f.\kappa, D_f.\iota, D_f.\sigma, D_f.\omega)$$

Transformer(D_x): *(Algorithme de transformation de D_x selon D_f)*

- Tant que `L'objectif n'est pas atteint` ($\mathcal{R}(D_i) \neq \mathcal{R}(D_f)$) faire
  - `Instancier la liste des opérateurs applicables à` D_x `avec les données du monde connues.`
  - Tant que `un opérateur` op `est applicable` faire
    - `Générer` $\mathcal{R}(op(D_x))$
    - Si $Filtrage(\mathcal{R}(op(D_x)), \mathcal{R}(D_f))$ alors
      - `Créer la donnée de monde` $D_y = (op(D_x))$
      - Si `la donnée` D_y `n'appartient pas déjà au monde` ($\mathcal{R}(D_y) \neq \mathcal{R}(D_\alpha)$ $\forall \alpha \in Plan, y \neq \alpha$) alors
        Transformer(D_y)

Nous possédons alors un système de planification linéaire [Lar94] mais sans retour dans l'arborescence grâce au processus d'instanciation des opérateurs applicables et aux descriptions des données complètes toujours présentes et accessibles.

5.2 Aide à l'utilisation du traitement d'images

La modélisation des opérateurs et des données que nous avons implantée permet de fournir à l'utilisateur trois niveaux d'aide qui répondent au cahier des charges présenté au paragraphe 3 :

- un complément d'information sur les traitements réalisés, lors d'une conception manuelle d'un graphe d'opérateurs,
- une nouvelle méthodologie de conception mieux adaptée à l'utilisateur car privilégiant la spécification du problème à résoudre plutôt que la sélection des opérateurs.
- une description précise des opérations pour le spécialiste du traitement d'images.

5.2.1 Aide à la conception d'application

Assistance lors la conception manuelle d'un graphe Il s'agit d'assister l'utilisateur d'un logiciel de traitement d'imageslorsqu'il sélectionne des opérateurs dans un menu, afin de constituer un script résolvant son problème d'analyse ou d'interprétation d'image. L'utilisation de notre modèle permet de fournir, lors de l'élaboration du script, une source d'informations supplémentaire correspondant à la description de ce qui s'est produit lors de l'application de l'opérateur. En plus du script et des résultats de son exécution, l'utilisateur a ainsi accès à la description formelle des données obtenue par propagation des transformations le long de la chaîne et exprimée en termes d'indices visuels. En cas d'erreur, il dispose ainsi d'indications pour ajuster les paramètres de l'opérateur ou remettre en cause sa sélection.

Aide à la conception Cette possibilité de construire, par propagation, la description d'un résultat d'enchaînement est exploitée pour proposer une nouvelle méthodologie de conception beaucoup plus adaptée aux compétences de l'expert d'un domaine. Au lieu de lui présenter des opérateurs de traitement d'images, ce qui supposerait qu' il est compétent en traitement d'images, l'interface lui procure des outils lui permettant de décrire son objectif à l'aide d'indices visuels basés sur la notion de mesure. Le système construit alors la chaîne d'opérateur qui produit la même description. La construction d'une application se fait de façon incrémentale, l'utilisateur définissant et combinant progressivement les éléments de plus en plus complexes qui composent son domaine d'application. Il dispose d'un retour visuel affichant les éléments extraits de l'image. Les principales opérations disponibles sont la nomination de concepts, la généralisation et la spécialisation. Des outils spécifiques d'interaction (langage visuel) agissant dans plusieurs espaces (image, caractéristiques, description formelle), correspondant à plusieurs vues du même concept, permettent d'effectuer ces opérations dans la vue la plus pertinente pour manipuler le concept. Ils consistent en des interactions très explicites pour l'utilisateur, telles que la désignation d'exemples ou de contre-exemples ou la définition d'un domaine. Ils permettent également de garantir l'exactitude syntaxique et la pertinence sémantique de ces interactions, déchargeant l'utilisateur des contraintes spécifiques à l'implantation du traitement d'imageset lui permettant de se concentrer sur l'objectif de son application. Enfin ce formalisme fournit également une justification "contractuelle" du résultat.

5.2.2 Aide à l'apprentissage

Le spécialiste en traitement d'images, qui doit développer de nouvelles fonctionnalités, utiliser ou modifier un opérateur, doit avoir une information très fine sur l'effet des opérateurs de traitement d'images sur les divers types de données. Un opérateur d'une bibliothèque est toujours un macro-opérateur. Le formalisme proposé met en évidence la composition de fonctions élémentaires qu'il réalise. Il permet de concevoir des outils de visualisation des effets d'un opérateur sur chaque champ des données -trame, cache, index, signal, codage -. La description symbolique de l'opérateur sur ces cinq champs permet de concevoir des variantes d'un opérateur : l'utilisateur peut apporter une modification sur l'un de ces champs, simuler son utilisation dans une chaîne et ne le réaliser que s'il donne satisfaction. De même ce modèle permet de bien mettre en évidence les différences et les points communs de deux opérateurs implémentant la même fonctionnalité. Ces comparaisons peuvent être exprimées en terme d'opérations effectuées ou en terme d'effets produits sur les images.

5.2.3 Limites actuelles

Une application de traitement d'images mettant en jeu une part de reconstruction de la scène. Pour une application complexe, cette reconstruction peut être assez complète. Le résultat recherché devra être exprimé dans l'espace 2D 1/2 ou 3D que notre modèle ne prend pas en compte. De même le langage visuel est encore incomplet pour exprimer toutes les relations spatiales ce qui limite les capacités de formulation du problème. Enfin, il sera nécessaire de structurer l'espace des mesures de façon à fournir à l'utilisateur un langage de plus haut niveau pour exprimer ses objectifs.

6 Conclusion

Le traitement d'images est une technique informatique de traitement et d'extraction de l'information contenue dans un signal. Il est toujours utilisée dans une application spécifique à un domaine, manipulant cette information. Il faut donc pouvoir étudier le traitement d'images sous ce triple aspect : son action, en terme de traitement du signal image, son action sur les structures de représentation des données, son utilisation, c'est à dire son action sur l'information (la sémantique) associée à ces structures de représentation, autrement dit, la manipulation de concepts qu'il permet de réaliser. Notre représentation est basée sur un modèle précis de donnée qui permet de rendre bien compte de l'action informatique de l'opérateur dont la représentation permet de constater la propagation de ces transformations le long d'un enchaînement d'opérateurs. Notre modèle de la donnée de traitement d'images permet de dégager un formalisme de description de concepts faisant ainsi le lien avec le domaine d'application. Il permet donc de construire un environnement complet de compréhension et de simulation du traitement d'images. Il offre de nombreuses possibilités d'exploitation pour l'enseignement du traitement d'images aussi bien pour la compréhension des fonctionnalités du traitement d'images que pour son utilisation ou pour son enrichissement par de nouvelles fonctionnalités.

Références

[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.

[Cap94] O. Capdevielle et P. Dalle. – Image Processing chain construction by interactive goal specification. *In : ICIP, Austin Texas (U.S.A.)*, 1994.

[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.

[Clo93] R. Clouard, Ch. Porquet et M Revenu. – Resolution of image processing problems by dynamic planning within framework of the blackboard model. *SPIE: intelligent Robots and Computer Vision XII*, vol. 2056, 1993. – Boston.

[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.

[Dej96] Ph. Dejean et P. Dalle. – Modèle symbolique de la donnée de traitement d'images. *In : 10e RFIA, Rennes (France)*, 1996, p. 746.

[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.

[Fli90] M. Flicker, M. Lavin et S.Das. – An object-oriented langage for image and vision executive (olive). *In : 10e ICPR, Atlantic city, New Jersey*, 1990, pp. 561–571.

[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.

[KBU95] First international workshop on Knowledge-Based systems for the (re)Use of Program libraries. INRIA - Sophia Antipolis, Novembre 1995.

[Lar94] H. Laruelle, P. Regnier et S. Thiebaux. – *Dossier Planification et Action.* – Rapport de recherche n° 94/1/R, IRIT, 1994.

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

[Mic83] R.S. Michalski. – A theory and methodology of inductive learning. *Artificial Intelligence*, vol. 20, n° 2, 1983, pp. 111–161.

[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.

[Noe92] Noesis, 13 Av M Saulnier, 78140 Vélisy. – *Visilog 4 Reference Guide*, 1992.

[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.

[Tho93] M. Thonnat, V. Clement et J. van den Elst. – *Supervision of Perception Tasks for Autonomous Systems : the OCAPI Approach.* – Rapport de recherche n° 2000, INRIA, Juin 1993.

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