



HAL
open science

Un système pour la formulation d'applications de traitement d'images

Arnaud Renouf, Régis Clouard, Marinette Revenu

► **To cite this version:**

Arnaud Renouf, Régis Clouard, Marinette Revenu. Un système pour la formulation d'applications de traitement d'images. *Traitement du Signal*, 2007, 24 (5), pp.337-352. hal-00260086

HAL Id: hal-00260086

<https://hal.science/hal-00260086>

Submitted on 16 Apr 2013

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.

Un système pour la formulation d'applications de traitement d'images

System for the formulation of image processing applications

Arnaud Renouf, Régis Clouard et Marinette Revenu

GREYC – CNRS UMR 6072, 6 boulevard Maréchal Juin 14050 Caen cedex, France
Arnaud.Renouf@greyc.ensicaen.fr

Manuscrit reçu le

Résumé et mots clés

Nous présentons un système de formulation d'applications de traitement d'images destiné à des utilisateurs novices. La construction de ce système nous amène d'abord à concevoir un modèle de la formulation de telles applications afin d'identifier et d'organiser les informations nécessaires et suffisantes pour qu'un spécialiste ou un système de génération automatique de programmes puisse concevoir une solution acceptable. Ce modèle couvre toutes les catégories et tous les domaines d'application du traitement d'images. Nous proposons ensuite un modèle d'interaction qui permet d'aider et de guider les utilisateurs dans la formulation de leurs applications.

Applications de traitement d'images, Ingénierie des connaissances, Ontologie, Ancrage, Modèle de formulation.

Abstract and key words

We present a system for the formulation of image processing applications from inexperienced users. To construct such a system, we firstly design a formulation model of image processing applications to identify and to organize the necessary and sufficient information that an expert or an automatic programs generation system needs to conceive an acceptable solution. This model covers every category and domain of the image processing field. Secondly we propose an interaction model which helps and guides the end-users in the formulation of their applications.

Image processing applications, Knowledge engineering, Ontology, Anchoring problem, Symbol grounding, Formulation model.

1. Introduction

Depuis une cinquantaine d'années, un très grand nombre d'applications de traitement d'images ont été réalisées dans des domaines très divers (médecine, géographie, robotique, vision industrielle, ...). Pourtant, la définition formelle du problème adressé par une application de traitement d'images n'a été que

peu étudiée en tant que telle. Une telle formulation doit poser clairement le problème à résoudre, c'est-à-dire définir les résultats à construire et identifier le domaine de variation des données d'entrée. En pratique, les spécialistes développent des applications sans réelle définition du problème; la solution algorithmique obtenue constituant alors en général l'unique formulation. Or, ces informations sont un préalable à la construction de solutions plus fiables et plus robustes puisqu'elles permettent :

- de délimiter finement le champ d'applicabilité de la solution ;
- d'expliciter la logique de conception par la justification des choix réalisés ;
- de bâtir des règles d'évaluation adaptées.

Elles sont également une contribution majeure à la réutilisation, voire à la simple reproduction des solutions.

Si la formulation a été si peu étudiée, c'est parce qu'elle est essentiellement de nature qualitative et qu'elle relève de choix subjectifs. Ceci signifie qu'il n'existe pas de définition exhaustive ni exacte du problème mais simplement une caractérisation approchée du comportement souhaité de l'application [1].

Nous proposons ici de modéliser la formulation d'applications de traitement d'images. Afin d'éprouver les modèles proposés, nous avons conçu une interface homme/machine qui les met en œuvre. Cette interface permet à des utilisateurs novices du traitement d'images de formuler des applications de traitement d'images.

1.1. Notion d'application de traitement d'images

Par application de traitement d'images, nous entendons **un logiciel spécialisé dans l'accomplissement d'un objectif de transformations d'images en images, sans interprétation du contenu, pour des images d'une classe donnée**. Une définition plus précise des transformations d'images couvertes par le domaine du traitement d'images est donnée en Section 2.1.2. Les images sont des représentations 2D, 3D ou 4D d'un phénomène mesuré ou calculé (images iconiques, statistiques, intrinsèques et segmentées). La classe d'images d'entrée est définie par un ensemble de caractéristiques invariantes partagées par les images. Les images de sortie ne sont en général pas une fin en soi mais servent d'entrée à une tâche de post-traitement (comptage d'objets, mesure de caractéristiques des images ou d'objets, visualisation, interprétation, ...).

Nous déduisons de la définition précédente que la formulation d'une application comprend la spécification d'un ensemble d'objectifs de traitement d'images et la définition d'une classe d'images d'entrée. Si nous prenons comme exemple l'application de cytologie décrite dans [2], l'objectif est d'extraire les noyaux de cellules de séreuse. Les entrées sont des images acquises par une caméra CCD montée sur un microscope optique et les sorties sont des cartes de régions où chaque région identifie un noyau [Figure 1]. Ces cartes sont ensuite utilisées pour une tâche de post-traitement de classification de cellules. Cette application servira de fil rouge dans la suite de cet article.

1.2. Le contexte

Nos travaux s'inscrivent dans le projet Panthéon¹ qui a pour but de construire un système qui génère automatiquement des logi-

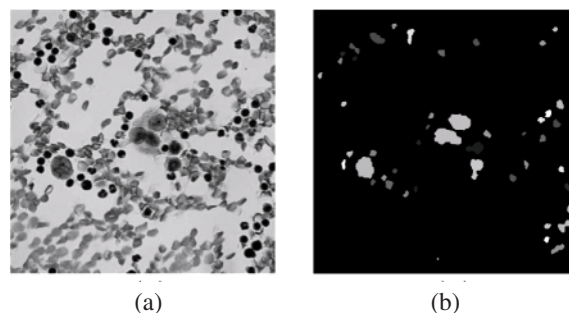


Figure 1. (a) Exemple d'image d'entrée de l'application de cytologie de séreuse. (b) Exemple de résultat obtenu par l'application (une extraction des noyaux de cellules de séreuse).

ciels de traitement d'images spécifiques d'une application définie par un utilisateur. Ce système est composé de deux sous-systèmes [Figure 2]:

- un système de formulation d'applications de traitement d'images ;
- un système de génération automatique de programmes de traitement d'images [3].

L'utilisateur spécifie le problème avec les termes de son domaine par interaction avec le niveau utilisateur du système de formulation. Cette partie du système correspond à une interface homme-machine qui utilise une *ontologie de domaine* pour présenter les informations destinées à l'utilisateur. Elle regroupe des concepts lui permettant de spécifier ses intentions de traitement et de définir sa classe d'images. La formulation ainsi obtenue au niveau utilisateur est une représentation du problème à résoudre dans les termes du domaine d'application. Le système de formulation traduit alors la formulation utilisateur dans les termes du traitement d'images en s'appuyant sur une *ontologie du traitement d'images*. La traduction de la formulation nécessite la résolution du problème de l'ancrage des symboles intro-

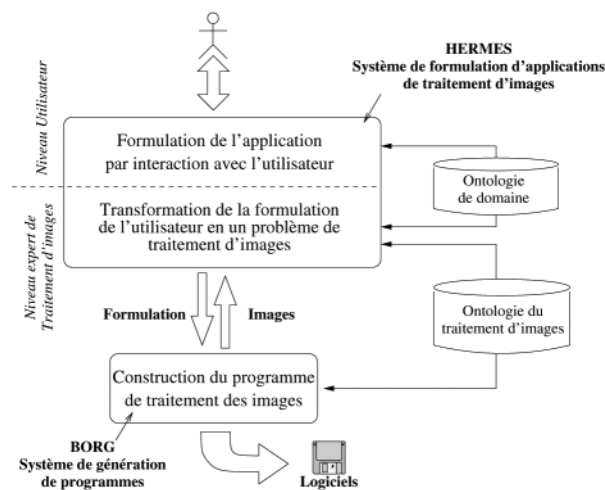


Figure 2. L'architecture du système du projet Panthéon.

1. Voir <http://www.greyc.ensicaen.fr/~regis/Pantheon/index-fr.html>

duits par l'utilisateur (c'est-à-dire les termes de son domaine). Cet ancrage des symboles vise à établir une correspondance entre les termes renseignés par l'utilisateur et les données image [4][5].

La formulation ainsi obtenue est transmise au système de génération de programmes qui raisonne sur les informations fournies pour construire une solution logicielle adaptée. Cette coopération entre les deux sous-systèmes nécessite le partage de l'*ontologie du traitement d'images*. Les images obtenues par l'exécution du programme généré sont envoyées au système de formulation pour être présentées à l'utilisateur. Celui-ci peut alors les évaluer et corriger sa formulation initiale, puis relancer la génération du programme. Le résultat final est un programme de traitement d'images validé par l'utilisateur.

Dans cet article, nous présentons uniquement le système de formulation d'applications. Nous n'aborderons pas les problèmes d'évaluation des résultats et de reformulation.

1.3. État de l'art de la formulation d'applications de traitement d'images

Les premiers travaux abordant le problème de la formulation d'applications de traitement d'images ont été conduits autour de systèmes à base de connaissances pour le traitement d'images tels que LLVE [6], CONNY [7], OCAPI [8], MVP [9] ou BORG [3]. Des connaissances *a priori* sur le contexte (effets du capteur, type de bruit, conditions d'illumination, ...) et sur les buts à atteindre sont pris en compte relativement explicitement dans la base de connaissances de ces systèmes. La formulation de l'application consiste alors en un choix d'un objectif de transformation dans une liste de tâches prédéfinies de traitement d'images et à renseigner quelques éléments du contexte. Ceci a permis à ces systèmes d'accroître leur champ d'applicabilité et d'améliorer leur robustesse. Néanmoins, ils restent souvent très limités à un domaine parce que trop de connaissances sur le problème restent implicites.

Une approche originale de la formulation d'applications de traitement d'images a été initiée par J. Hasegawa *et al.* [10] et dont l'idée a été reprise plus récemment par B. Draper [11]. L'utilisateur dessine sur un exemple d'image le résultat à obtenir. Le programme solution est alors construit en inversant une chaîne de traitement générée pour passer du résultat à l'image initiale. L'intérêt de cette approche est que la formulation obtenue est de nature quantitative et qu'elle prend ses données directement dans l'image. Néanmoins, elle ne peut s'appliquer qu'à un ensemble restreint d'objectifs de segmentation ou de détection bien connus *a priori* parce que les objectifs, comme la définition des éléments d'intérêt dans les images, ne peuvent se déduire d'une ou plusieurs images exemples. Pour couvrir toutes les tâches de traitement d'images, il est donc nécessaire d'avoir une représentation explicite du problème.

Des approches plus récentes apportent des modélisations plus explicites [12] [13] [14] [15] mais elles se concentrent toutes sur la modélisation de la description des objets métier pour des tâches de détection ou de segmentation, dans le cadre de la recherche d'images par l'exemple, de l'annotation d'images ou de reconnaissance d'objets. Elles utilisent des ontologies qui fournissent les concepts nécessaires à cette description (une ontologie de concepts visuels pour la reconnaissance d'objets dans [13], une ontologie de descripteurs visuels pour l'annotation sémantique des images et des vidéos dans [16] ou des primitives du traitement d'images dans [12]) ou elles proposent de faire l'acquisition des connaissances métier au cours d'interviews des spécialistes du domaine d'application (*e.g.* utilisation de la méthode NIAM/ORM dans [14] pour collecter puis faire le lien entre connaissances métier et connaissances vision). Par contre, elles ne prennent pas ou peu en compte le contexte d'application (le système d'acquisition et son environnement de mesure) et les effets induits par ce contexte sur les images à traiter. De même, elles ne donnent pas les moyens de décrire le contenu des images lorsque les objets sont *a priori* inconnus (*e.g.* en robotique, dans des applications de restauration ou d'indexation). De plus, elles supposent également que l'objectif est unique et connu *a priori* (détecter, extraire ou reconnaître un objet avec un ensemble de contraintes restrictives) et, par conséquent, ils n'abordent pas le problème de sa spécification. Y. Saidali [17] présente bien un modèle pour représenter les objectifs de traitement et le contexte d'applications de traitement d'images mais il est limité au domaine du document numérique et se focalise plus sur la modélisation des solutions que sur la formulation du problème en lui-même.

1.4. Plan de l'article

Afin de répondre au problème de la formulation d'une application de traitement d'images, nous proposons un modèle complet qui permet d'identifier et d'organiser les informations nécessaires et suffisantes pour qu'un système de génération, ou un spécialiste du traitement d'images, produise une solution acceptable. Ce modèle doit couvrir toutes les tâches de traitement d'images et uniquement ces tâches, et rester indépendant de tout domaine particulier. Dans la Section 2, nous présentons les contraintes imposées par la définition d'un tel modèle, les hypothèses sur lesquelles il repose, et le modèle conceptuel qui en découle. Dans la Section 3, nous détaillons le modèle de formulation au niveau expert de traitement d'images. Dans la Section 4, nous présentons le modèle de formulation au niveau utilisateur et nous exposons le problème de l'ancrage et les moyens fournis pour le résoudre.

2. Les contraintes et hypothèses du modèle de formulation au niveau expert de traitement d'images

Dans cette section, nous identifions les contraintes imposées par la problématique du traitement d'images à la définition d'un modèle de formulation d'applications. Nous décrivons ensuite les hypothèses sur lesquelles reposent la validité de ce modèle et le modèle conceptuel élaboré.

2.1. Les contraintes

2.1.1. Le modèle doit rester indépendant à tout domaine d'application particulier

Pour espérer couvrir n'importe quel domaine d'application, le modèle doit être dépourvu de toute référence à un quelconque domaine. Par contre, le modèle doit permettre de représenter les connaissances d'un domaine particulier acquises auprès de l'utilisateur.

2.1.2. Le modèle doit intégrer toutes les tâches de traitement d'images

Nous voulons prendre en compte toutes les tâches de traitement d'images. Il nous faut pour cela poser au préalable une définition de ce domaine puisqu'il n'existe pas de définition consensuelle. D'un point de vue intentionnel, nous définissons le traitement d'images à partir de six catégories de tâches [Tableau 1]. Elles résultent de la combinaison des trois transformations de base – ajout, modification et suppression – appliquées au niveau signal et produisant des informations à considérer au niveau signal ou au niveau symbole. L'objectif de la restauration est de retrouver l'image d'origine à partir de sa version dégradée par ajout d'information au niveau signal connaissant la fonction de dégradation (ou au moins une estimation). L'amélioration a pour but d'adapter au mieux les données à une exploitation visuelle par modification de l'information au niveau signal en utilisant uniquement les valeurs de l'image d'entrée. La compression a pour objectif de réduire la quantité de données nécessaire pour stocker une image numérique en supprimant de l'information identifiée comme redondante au niveau signal (même si l'image d'entrée est une image segmentée). La reconstruction consiste à créer une nouvelle image par ajout d'information au niveau symbole, comme la troisième dimension ou le mouvement, déduite d'une image (e.g. « shape from shading ») ou de plusieurs (e.g. stéréovision). La segmentation structure les pixels à un niveau d'information supérieur (régions et contours).

La détection fournit un masque positif des zones de l'image contenant les objets d'intérêt et supprime toutes les autres informations.

Tableau 1. Les six catégories de tâches du traitement d'images.

	Signal → Signal	Signal → Symbole
Addition	Restauration	Reconstruction
Modification	Amélioration	Segmentation
Suppression	Compression	Détection

2.1.3. Le modèle doit intégrer des connaissances hétérogènes

La formulation d'un problème définissant une application de traitement d'images intègre des données de natures diverses, numériques et symboliques, des relations entre concepts, des descripteurs, des concepts du domaine d'application et des primitives visuelles. Nous devons également définir les types de valeurs qui peuvent être utilisés dans la formulation : des termes prédéfinis, des variables linguistiques, des comparaisons, des valeurs numériques, des intervalles, des ensembles d'intervalles, des matrices, des parties d'images (i.e. « patches »), ou des images.

2.2. Les hypothèses du modèle de formulation

Face à la complexité de la tâche de modélisation de cette formulation, nous émettons trois hypothèses qui permettent d'orienter le choix des informations qui sont à représenter. Ces hypothèses sont par conséquent très fortes puisqu'elles conditionnent le modèle ainsi construit.

L'invalidité d'une de ces hypothèses signifierait donc que le modèle ne permet pas de représenter les informations nécessaires et suffisantes au développement d'une solution logicielle.

2.2.1. L'hypothèse téléologique

Le traitement d'images n'est pas une fin en soi. En effet, le problème à résoudre n'est pas dans les données (les images d'entrée) tout d'abord parce que les données image sont par nature incomplètes, dégradées et erronées, et ensuite parce que l'image n'a pas de sens en elle-même ; elle n'a de sens que pour un objectif donné.

L'hypothèse téléologique affirme alors qu'une application de traitement d'images se définit par ses finalités. Toutefois, nous considérons que ces finalités ne peuvent pas toujours être représentées de façon consistante par la description des résultats attendus (comme dans l'approche proposée par A. Nouvel [18]) en raison de la faiblesse d'expressivité des descripteurs face à la variabilité des configurations possibles, et de l'impossibilité de

construire des résultats exemples (comme pour la restauration d'images par exemple). En conséquence, nous choisissons de spécifier les finalités par la liste des tâches à accomplir (et des contraintes qui les spécialisent) qu'il est toujours possible de donner.

2.2.2. L'hypothèse sémiotique

L'hypothèse sémiotique considère une image à traiter comme un système de signes mis pour représenter une chose réelle ou artificielle [19]. En s'appuyant sur cette hypothèse, la sémiotique nous amène à définir la classe d'images en distinguant trois niveaux de description :

- Le niveau du signe (nommé ici niveau physique) se focalise sur le signal mesuré. Il s'intéresse à la caractérisation des effets de la chaîne d'acquisition sur les images produites (*e.g.* bruit, distorsions, illumination, ...).
- Le niveau du signifiant (nommé ici niveau perceptif) se focalise sur la « syntaxe » de l'image. Il s'intéresse à la description des primitives visuelles qui composent les images sans référence aux objets (*e.g.* l'image est composée de régions de couleur homogène, de contours contrastés et d'un fond très homogène).
- Le niveau du signifié (nommé ici niveau sémantique) se focalise sur la scène observée. Il s'intéresse à l'identification visuelle des objets du domaine présents dans les images (*e.g.* le cytoplasme entoure le noyau et sa couleur est bleue et plus claire que la couleur des noyaux).

2.2.3. L'hypothèse phénoménologique

L'hypothèse phénoménologique affirme qu'une description de la manifestation visuelle de la scène suffit et qu'il n'est pas nécessaire de la décrire dans sa réalité. Ces informations diffèrent de celles utilisées en interprétation d'images qui nécessitent en plus des informations de nature ontologique. L'hypothèse phénoménologique a deux conséquences :

- seuls les objets et les effets de l'acquisition qui ont une manifestation visuelle dans les images doivent être représentés. Par exemple, à une résolution faible, il n'est plus possible de distinguer le noyau et le cytoplasme. D'un point de vue phénoménologique, ils se confondent en un seul objet (la cellule), et par conséquent seul l'objet cellule est décrit.
- il n'est pas non plus nécessaire de décrire les objets du domaine dans leur complexité. Par exemple, il est inutile pour traiter les images de savoir que, dans une image de tissu sain, il y a plus de cellules épithéliales que de cellules de séreuse. C'est par contre une information précieuse pour l'interprétation.

Il en découle qu'une dénotation des images sous la forme d'une liste descripteur-valeurs est suffisante pour concevoir une solution de traitement d'images.

3. Le modèle de formulation au niveau expert de traitement d'images

Dans cette section, nous présentons d'abord le modèle conceptuel et ensuite le détail des deux parties du modèle (la spécification des objectifs et la définition de la classe d'images) au niveau expert de traitement d'images. Une description détaillée de l'ensemble de ce modèle est donnée dans [20].

Remarque : dans cette section, tous les éléments de l'ontologie sont donnés en anglais qui est la langue dans laquelle elle a été définie.

3.1. Le modèle conceptuel

L'hypothèse téléologique (Section 2.2.1) conduit à exprimer les objectifs par des tâches. Pour ne pas avoir à gérer le problème du cadre (le « Frame problem » dans la littérature anglaise), nous sommes amenés à considérer les tâches une à une de façon indépendante et donc à associer une définition de la classe d'images à chacune. En effet, il est très difficile de prédire les effets exacts d'une tâche sur les images d'entrée et donc de modifier automatiquement la définition de la classe d'images. L'enchaînement de plusieurs tâches de traitement d'images ne peut donc se contenter d'une définition initiale unique qui serait modifiée de tâche en tâche.

Une application globale peut donc être composée de plusieurs tâches enchaînées de façon conditionnelle, séquentielle ou parallèle. Nous considérons alors chaque tâche comme une application à part entière et traitée de façon indépendante. Sur notre exemple d'application de cytologie présenté dans la section 1.1, nous obtenons alors la liste des applications suivantes :

1. Correction de l'illumination ;
2. Détection de présence des noyaux de cellule de séreuse ;
3. Élimination des globules rouges ;
4. Extraction des noyaux de cellule de séreuse.

La détection de présence permet d'arrêter les traitements s'il n'y a pas de noyaux de cellules de séreuse.

Suivant les hypothèses émises, la formulation de chaque application est alors complètement définie par les catégories d'information suivantes :

1. un objectif exprimé par (*cf.* hypothèse téléologique) :
 - une tâche de transformation à accomplir ;
 - des contraintes sur cette tâche.

2. une définition de la classe d'images sur trois niveaux (cf. hypothèses sémiotique et phénoménologique):

- le niveau physique;
- le niveau perceptif;
- le niveau sémantique.

3.2. La spécification de l'objectif

L'objectif est spécifié par une tâche et un ensemble de contraintes. En se référant à l'analyse systémique, chaque tâche consomme des images issues d'un système de production d'images et fournit des images à un système de post-traitement (Figure 3). Le système de production d'images est soit une application externe, soit une autre application de traitement d'images amont, soit un dispositif d'acquisition d'images. Le post-traitement est soit une application de traitement d'images aval, soit un système de plus haut niveau d'abstraction comme un système de visualisation, d'interprétation, ou de classification.

La spécification de la tâche est précisée par l'ajout de trois types de contraintes [1]:

- les contraintes de régulation qui spécifient le focus d'attention de la tâche;
- les contraintes de rétroaction qui précisent la composition du résultat attendu de la tâche;

- les contraintes de contrôle qui permettent de prendre en compte l'environnement d'exécution de la tâche.

Nous donnons dans le tableau 2 la spécification de l'objectif pour l'application d'extraction des noyaux de cellule de séreuse de l'application de cytologie. Les différents éléments présentés dans ce tableau seront présentés dans la suite de cette section.

3.2.1. La tâche

Une tâche est définie par une action associée ou non à un argument. L'action est spécifiée par un verbe à l'impératif qui traduit une intention de traitement sur les images. Ces actions sont groupées suivant les six catégories d'objectif de traitement d'images présentées en Section 2.1.2. L'argument est le complément d'objet du verbe définissant l'action. Il précise sur quel élément porte la tâche. Il est un objet métier défini au niveau sémantique de la classe d'images (p. ex. « Extract <business object> »), ou une primitive visuelle définie au niveau perceptif (p. ex. « Detect <edge> ») ou une propriété générale de l'image affectée par la chaîne d'acquisition caractérisée au niveau physique (p. ex. « Correct <photometry> » ou « Enhance <colorimetry> »).

3.2.2. Les contraintes de régulation

Les contraintes de régulation portent sur la tâche. Nous définissons ces contraintes par des **critères à optimiser** qui identifient

Tableau 2. Spécification de l'objectif d'extraction de noyaux de cellule de séreuse.

Task	Extract {Serous Nucleus}
Regulation constraint (Criterion to be optimized)	- Maximize hits Acceptable error: prefer false alarm than miss - Optimize boundary localization Acceptable error: prefer boundary inside the region
Feedback constraint (Level of detail)	- Separate just-touching Acceptable error: prefer no separation - Exclude border touching
Control Constraint	- Performance criterion: Optimization = time - Quality criterion: Ability = reliability

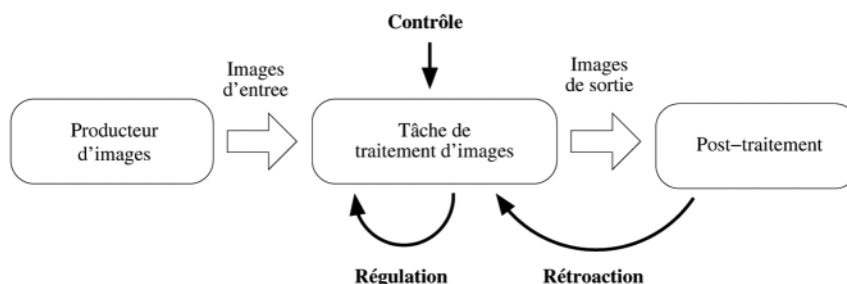


Figure 3. Représentation systémique d'une tâche de traitement d'images.

les éléments sur lesquels doit se focaliser la tâche. Chaque tâche peut donc être associée à un ensemble de critères à optimiser, chacun pouvant être lui-même associé à une **erreur acceptable** pour donner une préférence en cas de compromis à décider.

Par exemple, pour la tâche d'extraction de l'application de cytologie, un critère à optimiser est la localisation des frontières des noyaux puisque l'utilisateur désire mesurer la surface des noyaux lors du post-traitement. L'erreur acceptable associée est alors de préférer placer la frontière à l'intérieur de la région plutôt qu'à l'extérieur, dans le but d'éviter des erreurs de mesures photométriques menées également durant ces tâches de post-traitement.

3.2.3. Les contraintes de rétroaction

Les contraintes de rétroaction portent sur les résultats attendus de la tâche. Nous définissons ces contraintes par des **niveaux de détail** qui précisent les limites des effets de la tâche. Chaque niveau de détail peut être associé à une **erreur acceptable** qui permet alors de résoudre les éventuels conflits pouvant apparaître lors de la construction du résultat.

Par exemple, pour la tâche d'extraction où nous voulons obtenir une région par noyau de cellule, un niveau de détail indique de séparer les noyaux qui se touchent et de garder en amas les noyaux qui se chevauchent. Un conflit entre « séparer » et « ne pas séparer » peut alors se produire lorsque la position des frontières des objets ne permet pas de décider si ceux-ci se chevauchent ou ne font que se toucher. Dans ce cas, l'erreur acceptable permet alors de décider de « ne pas séparer » puisque l'utilisateur préfère garder en amas des noyaux qui se touchent plutôt que de séparer des noyaux qui se chevauchent.

3.2.4. Les contraintes de contrôle

Les contraintes de contrôle reflètent les limites imposées par l'environnement d'exécution à la solution à concevoir. Elles sont donc associables à tous les types de tâche définis. Deux types de contraintes de contrôle sont définis :

- **Les critères de performance** expriment les optimisations visées sur l'utilisation des ressources mémoires et sur le temps de traitement (l'acceptation du mot optimisation doit être rapprochée de celle utilisée en compilation où il s'agit en fait d'un critère à privilégier et non d'un optimum à rechercher).
- **Les critères de qualité** expriment les exigences qualitatives sur les résultats à atteindre.

Nous avons défini un critère permettant de spécifier les capacités de l'application à obtenir. Deux différentes valeurs mutuellement exclusives peuvent être associées à ce critère :

- « fiabilité » précise que la fiabilité des résultats est la plus importante et qu'il faut obtenir les meilleurs résultats possibles dans les conditions spécifiées dans la description du contexte. Cela signifie donc ne pas obtenir de solution plutôt que d'en obtenir une qui soit approximative quand l'image sort légèrement des conditions idéales.

- « robustesse » précise que l'application doit être robuste. Cela signifie que lorsque l'image sort légèrement des conditions idéales, un résultat même médiocre doit être obtenu quitte à fournir de moins bons résultats dans les conditions idéales qu'avec une application où la fiabilité est recherchée.

Dans notre exemple d'application d'extraction des noyaux, le cytologiste désire obtenir rapidement des résultats : une optimisation en temps est donc visée (Tableau 2). La fiabilité est également une contrainte importante puisque des résultats très précis pour les mesures faites dans les post-traitements sont attendus.

3.2.5. La formalisation de la spécification des objectifs

Nous présentons sur la figure 4 la formalisation de la spécification des objectifs en utilisant le formalisme CML² de CommonKADS qui couvre la spécification d'ontologies [21]. Sur la figure 5, nous montrons l'exemple de la spécification de la tâche d'extraction des noyaux de cellule de séreuse.

Sur la figure 6, la spécification des contraintes de contrôle est présentée.

3.3. La classe d'images

L'hypothèse sémiotique (Section 2.2.2) nous amène à définir une classe d'images en considérant trois niveaux de description : les niveaux physique, perceptif et sémantique. Suivant l'hypothèse phénoménologique (Section 2.2.3), la description de chacun de ces niveaux est fournie par une liste de paires descripteur-valeurs.

3.3.1. Le niveau physique

L'hypothèse phénoménologique conduit à décrire le niveau physique par les effets de la chaîne d'acquisition et des conditions d'acquisition sur les images, et non les éléments qui constituent la chaîne d'acquisition des images. Par contre, l'analyse des différents éléments constituant les chaînes d'acquisition (éclairage, optique, capteur, convertisseur analogique-numérique, stockage) donne la liste des effets et des défauts qui peuvent être décrits. L'ontologie de *traitement d'images* contient la liste des propriétés générales des images (colorimétrie, photométrie, géométrie, échantillonnage, quantification, format, bruit) et les descripteurs associées à chacune de ces propriétés pour permettre de décrire ces effets et défauts. Seuls ceux se manifestant dans les images d'entrée doivent être renseignés. Sur notre exemple « fil rouge » nous obtenons une liste présentée en partie dans le Tableau 3.

2. Conceptual Modeling Language.

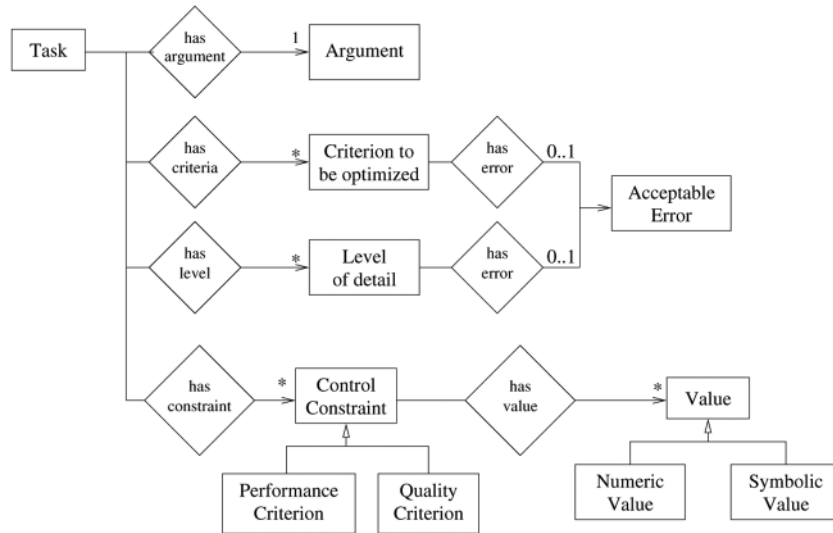


Figure 4. Représentation CML de la spécification des objectifs.

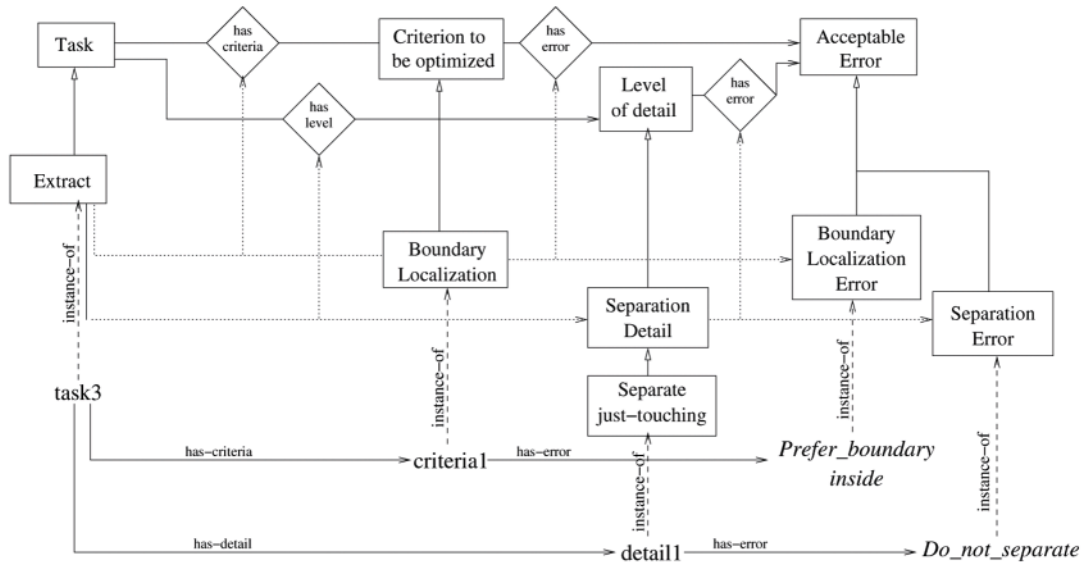


Figure 5. Spécification de la tâche d'extraction du concept1 (i.e. noyau de cellule de séreuse).

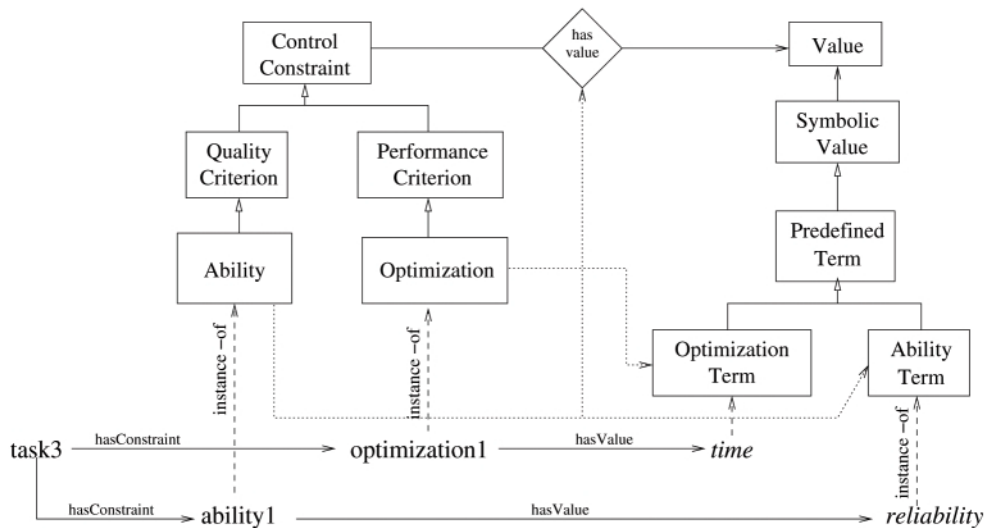


Figure 6. Spécification des contraintes de contrôle de l'application.

Tableau 3. Une partie de la description au niveau physique pour l'application de cytologie.

Acquisition effect	Acquisition effect descriptor	Value
Photometry	Illumination Temporal Condition	stable
	Illumination Spatial Condition	heterogeneous
	Illumination Model	<i>lighting.pan</i>
Image Format	Width	512 pixels
	Height	512 pixels
Colorimetry	Color Space	RGB
Quantification	Bits per pixel	24
	Function	linear
Noise	Composition	additive
	Distribution	Gauss
Blur	Direction	null
	Strength	Level = very low

3.3.2. Le niveau perceptif

Le niveau perceptif décrit le rendu visuel des images par les primitives visuelles sans référence aux objets métier. Il fournit des informations sur la composition visuelle des images sans introduire de notion d'objets. L'ontologie de *traitement d'images* référence toutes les primitives visuelles (région, contour, fond, point d'intérêt, zone d'image) et les caractérisations associées. Dans notre exemple « fil rouge », nous obtenons une description en termes de fond d'image, régions et contours [Tableau 4].

Tableau 4. La description au niveau perceptif pour l'application de cytologie.

Visual Primitive	Visual Descriptor	Value
Background	Lightness	[0.6 , 0.8]
	Saturation	[0 , 0.1]
Region	Texture	Type = no-texture
	Size	[10 , 50] pixels
Edge	Type	ramp

3.3.3. Le niveau sémantique

Le niveau sémantique décrit les objets métier de l'application tels qu'ils se manifestent sur les images d'entrée. La description se fait par les primitives visuelles qui les composent, par leurs relations topologiques (en utilisant le modèle RCC-8 [22]) et leurs relations spatiales (*e.g.* à gauche de, à droite de, au-dessus de, en-dessous de, devant, derrière, à une distance de, ...). Pour

ces dernières, le référentiel est celui de l'image. Il donne un sens au contenu des images en identifiant les objets ou les zones d'intérêt. Nous donnons l'exemple de la description de l'objet {noyau de cellule de séreuse} dans le Tableau 5 et les relations topologiques entre objets dans la figure 10.

Tableau 5. La description au niveau sémantique du concept de {noyau de cellule de séreuse} pour l'application de cytologie.

Business Object	Visual Primitive	Visual Descriptor	Value Value
Noyau de cellule de séreuse	Region	Lightness	[0 , 0.8]
		Hue	$[9\pi/8 , 3\pi/2]$
		Convexity	Level = very high
		Boundary Nature	edge
	Edge	Contrast	Level = low
		Type	ramp

3.3.4. La formalisation de la définition de la classe d'images

Nous présentons sur la figure 7 la formalisation de la structure de la définition de la classe d'images. Cette figure représente le plus haut niveau des concepts de l'ontologie du *traitement d'images*.

Sur la figure 8, nous montrons l'exemple de la définition des contours du concept de {noyau de cellule de séreuse} dans l'ontologie du *traitement d'images*.

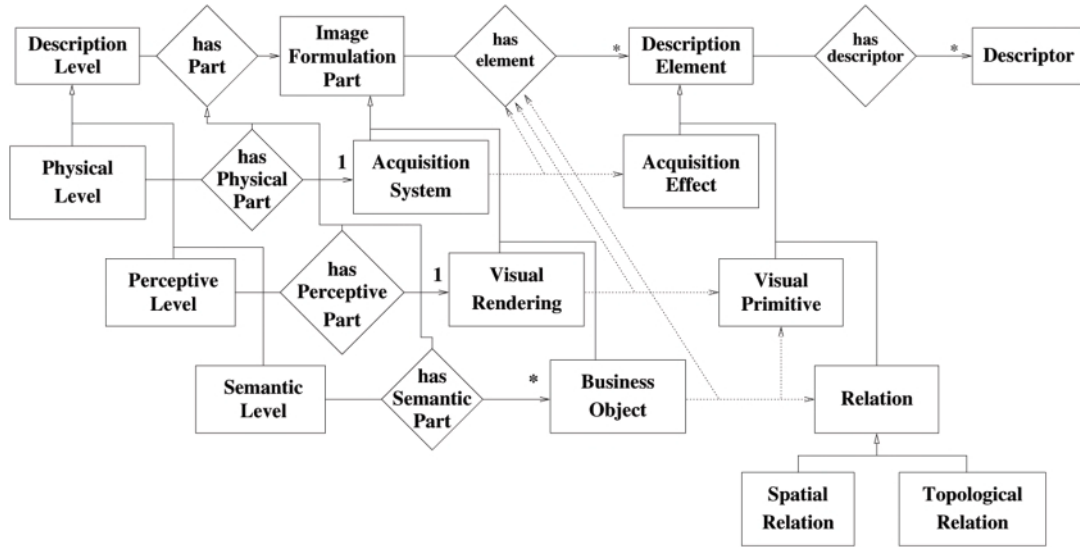


Figure 7. Représentation CML de la définition de la classe d'image.

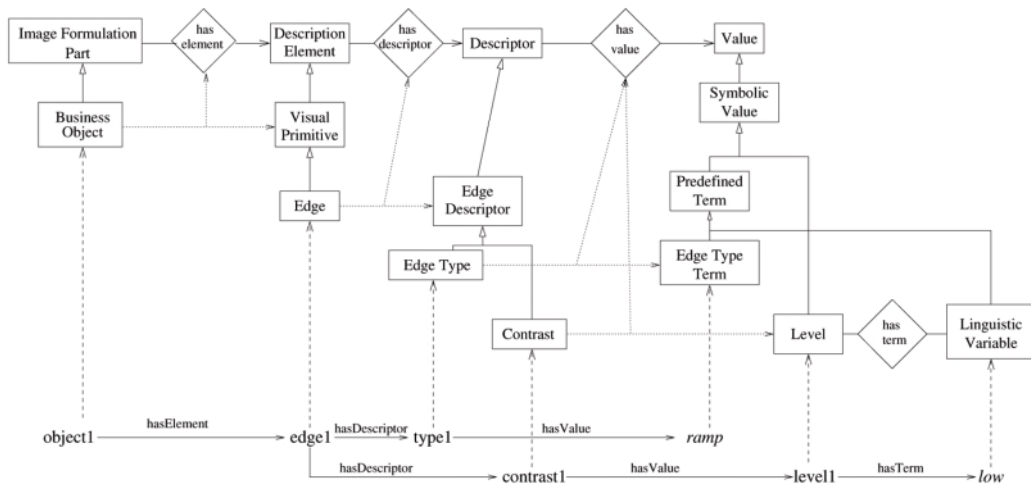


Figure 8. Définition des contours des noyaux de cellule de séreuse.

3.3.5. Discussion

Chaque observation définissant une classe d'images s'analyse selon les 3 niveaux. En fonction des connaissances *a priori* sur le problème, la définition de la classe d'images sera plus ou moins importante selon les niveaux. En particulier, le niveau perceptif est renseigné à défaut d'informations au niveau sémantique, c'est-à-dire quand les objets ne sont pas connus *a priori*. Dans la figure 9, nous représentons cette importance par la position de l'application dans le triangle sémiotique qui définit le référentiel de modélisation de la classe d'images. Pour illustrer le principe, trois types d'applications différentes sont placés dans ce triangle. La distance de la position du point au pôle (qui représente un niveau) est inversement proportionnelle à l'importance de la description de ce niveau.

- **L'application de cytologie**: dans cette application, une importante quantité d'information se décrit aux niveaux sémantique et physique: les modes d'acquisition et la scène sont bien connus de l'utilisateur et les objets sont prédictibles.

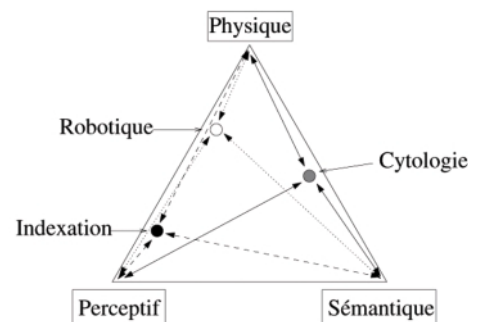


Figure 9. Trois types d'applications positionnés dans le référentiel de modélisation des classes d'images.

- **L'indexation**: dans les applications d'indexation et de recherche par l'exemple, très peu d'informations sont disponibles sur les effets de l'acquisition (que l'on ramène généralement à un bruit blanc gaussien) et les objets sont imprédictibles. Nous sommes alors limités à une description sur le niveau per-

ceptif en termes par exemple de points d'intérêt ou de régions homogènes.

- **La robotique**: les spécialistes de la robotique possèdent une bonne maîtrise de la chaîne d'acquisition (capteur, perturbations de l'environnement) mais les objets visualisés sont imprédictibles parce qu'ils sont trop nombreux et trop variés. Cette application se décrit essentiellement aux niveaux physique et perceptif: souvent une définition des contours pour décrire les objets par leur bord est utilisée et les connaissances sur le système d'acquisition renseignent, par exemple, le bruit contenu dans les images ou les distorsions géométriques de la caméra.

4. Le modèle de formulation au niveau utilisateur

La section précédente a permis d'identifier les informations nécessaires et suffisantes pour construire une application au niveau expert de traitement d'images. Dans cette section, nous proposons un modèle d'interaction permettant d'obtenir de l'utilisateur les informations qui définissent son application de traitement d'images.

4.1. Le modèle d'interaction

Le rôle du système proposé est d'aider l'utilisateur à fournir une formulation qui soit :

- **compacte**: réduite aux informations essentielles ;
- **représentative**: garde une forte correspondance avec les objets réels ;
- **discriminante**: contient assez d'informations pour discriminer les objets entre eux ;
- **précise**: permet une bonne identification des objets ;
- **dense**: caractérise tous les objets de la classe d'images et leur variabilité.

Nous considérons l'utilisateur d'un tel système en tant que novice du traitement d'images.

Le système devra donc proposer des interfaces amenant l'utilisateur à formuler son problème en utilisant des termes et des représentations facilement appréhendables.

En revanche, nous considérons aussi que l'utilisateur est un spécialiste de son domaine et qu'il est familier du système de formulation. À ce titre, il est donc capable de :

- choisir les tâches à réaliser dans une liste pour atteindre son objectif (en fonction de sa politique métier [1]) ;
- définir les contraintes associées aux tâches ;
- décrire les effets de la chaîne d'acquisition sur les images obtenues ;
- donner une description pertinente des images et des objets contenus ;

- évaluer les résultats du traitement des images.

C'est donc à lui qu'incombe la responsabilité de fournir une formulation qui soit à la fois **représentative** et **discriminante**.

En conséquence, le rôle du système d'interaction est d'aider à construire une formulation qui soit **compacte**, **précise** et **dense**.

Or nous savons que la formulation est essentiellement de nature qualitative puisqu'elle relève de choix subjectifs de la part de l'utilisateur, que ce soit pour le choix des éléments de spécification (tâches et descripteurs) aussi bien que pour leurs valeurs. Pour tenter d'objectiver au maximum le choix de ces éléments de spécification, nous proposons un modèle d'interaction qui s'appuie sur plusieurs principes. Les premiers permettent d'aider au choix des éléments de spécification :

- la spécification des objectifs de traitement d'images est initiée par les post-traitements.

- la définition de la classe d'images au niveau physique est guidée par la caractérisation de la chaîne d'acquisition.

- la définition de la classe d'images au niveau sémantique est basée sur la construction d'un arbre des objets.

Les seconds principes permettent d'aider à la valuation des éléments de spécification :

- la détermination de valeurs repose sur le choix d'exemples sur des images ou de valeurs symboliques.

- les éléments de discrimination entre les objets sont mis en évidence par l'utilisation de l'arbre des objets.

Nous ajoutons à ce niveau des informations qui ne se retrouveront pas au niveau expert de traitement d'images mais qui sont utilisées pour aider à la formulation.

4.2. Les moyens d'interaction utilisés

Nous présentons maintenant dans le détail le modèle d'interaction qui conduit l'utilisateur dans la spécification de ses objectifs et dans la définition de sa classe d'images.

4.2.1. La spécification des objectifs

Nous proposons d'abord de spécifier les objectifs de traitement d'images à partir des informations sur les post-traitements envisagés. Les post-traitements fournissent des informations sur la façon dont les résultats de l'application vont être utilisés. C'est un point d'entrée assez naturel pour spécifier les tâches de traitement d'images puisqu'ils sont du genre : mesures de taille, de forme, de position, d'orientation, de radiométrie, de topologie, visualisation artistique, visualisation de détails, comptage.

L'utilisateur choisit, dans un premier temps, les post-traitements qui seront effectués à partir des résultats de l'application de traitement d'images. Il détermine ensuite les tâches qui doivent être effectuées pour atteindre son objectif. L'interface lui propose pour cela une liste des tâches du traitement d'images, d'assez haut niveau, classées suivant les catégories présentées dans la section 3.2.1. Chaque tâche est commentée afin d'expliquer à l'utilisateur ce qu'elle produit en sortie et dans quels cas elle peut être utilisée. L'utilisateur choisit ensuite les contraintes

associées aux tâches. Le système utilise alors les spécifications des post-traitements donnés par l'utilisateur pour proposer des valeurs par défaut pour ces contraintes. Sur notre exemple fil rouge, des mesures de surface sont envisagées par l'utilisateur sur les régions obtenues par la tâche d'extraction des noyaux. Le système propose alors la localisation des frontières comme critère à optimiser. L'utilisateur désire également effectuer des mesures radiométriques. Le système propose donc une erreur acceptable pour ce critère à optimiser qui est de placer la frontière à l'intérieur de la région pour éviter de prendre des pixels du fond. Les valeurs par défaut ne sont émises qu'en tant que propositions; c'est toujours à l'utilisateur de choisir les contraintes.

4.2.2. La définition de la classe d'images

Le système aide également l'utilisateur dans la définition de sa classe d'images. Au niveau physique, nous avons construit une base de données sur les systèmes d'acquisition classiques pour proposer des valeurs de bruit ou de défauts qui sont caractéristiques de ces systèmes. Nous savons par exemple que les images produites par des capteurs échographiques contiennent un bruit de type speckle. Une telle base de données peut facilement et régulièrement être mise à jour par des spécialistes du traitement d'images. Toutefois, l'utilisateur peut, s'il maîtrise la chaîne d'acquisition, fournir lui-même les effets induits sur les images capturées.

Au niveau perceptif, l'utilisateur choisit dans une liste de primitives visuelles contenues dans l'*ontologie utilisateur* et présentées par le système, celles qu'il observe dans ses images. Il est amené à fournir les caractéristiques visuelles invariantes de chacune de ces primitives en utilisant des descripteurs qui lui sont proposés par le système. Les éléments de description portent ici sur la caractérisation des régions, des lignes, des frontières ou du fond.

Au niveau sémantique, nous avons choisi de définir la classe d'images en nous appuyant sur la construction d'un arbre des objets. Ce choix repose sur l'hypothèse que les spécialistes d'un domaine partagent une taxonomie des objets de leur domaine.

L'utilisateur identifie tous les objets d'intérêt présents dans ses images et les organise hiérarchiquement. Il spécifie pour cela les relations hyperonymiques et méronymiques qui existent entre ces objets [Figure 10]:

- la relation hyperonymique 'sorte-de' permet à l'utilisateur de donner la taxonomie de son domaine qui est une classification basée sur les similarités. Cette structure est à la base de deux inférences élémentaires que nous faisons tous les jours: l'identification qui est notre capacité à reconnaître la classe d'un objet à partir de ses caractéristiques et la spécialisation qui est notre capacité à prendre en compte les catégories plus précises que celles demandées lors d'une recherche d'information.
- la relation méronymique 'partie-de' permet à l'utilisateur de décrire la composition de ces objets.

Le système de formulation utilise l'arbre pour la description de chacun des objets en termes de primitives visuelles qui les composent, de relations topologiques et de relations spatiales (avec les relations topologiques et spatiales, l'arbre des objets devient un graphe). L'arbre constitué des objets liés par les relations hyperonymiques et méronymiques permet à l'utilisateur de mieux appréhender la discrimination des objets entre eux. Quatre principes fondamentaux déterminent la signification d'un nœud de l'arbre en fonction de son parent et de ses frères dans le cas d'une relation hyperonymique: le principe de communauté, c'est-à-dire de similarité ou de différence avec le père, le principe de différence et de similarité entre les frères. Les objets héritent donc des descriptions de leurs hyperonymes. La relation méronymique impose que chaque nouveau nœud créé dans l'arbre représente un objet qui ait une description différente des autres (pour au moins une valeur d'attribut). C'est le système qui impose à l'utilisateur le respect de ces principes par interaction. Ces informations obtenues auprès de l'utilisateur permettent de donner une part d'objectivité à sa formulation et pour le conduire vers une description **précise** et **dense** de ses objets.

4.2.3. L'ontologie utilisateur

Nous avons formalisé le modèle présenté précédemment sous forme d'une *ontologie de domaine*. Elle est basée sur la même

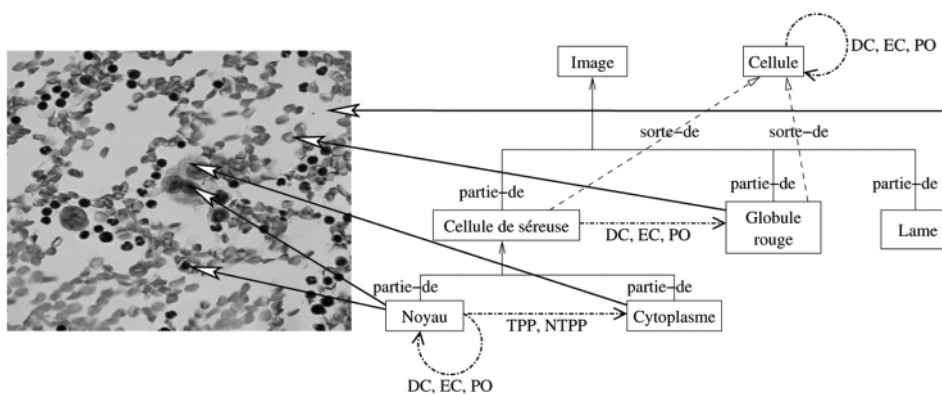


Figure 10. L'arbre des objets de l'application de cytologie et les relations hyperonymiques, méronymiques et topologiques (DC: déconnecté de, EC: connecté extérieurement à, PO: chevauche partiellement, TPP: inclut tangentiellement, NTPP: inclut non tangentiellement).

structure que l'*ontologie du traitement d'images* dont elle partage une partie des concepts (e.g. objectifs spécifiés par un ensemble de tâches et de contraintes, définition de la classe d'images sur trois niveaux), mais elle contient également des termes et concepts supplémentaires permettant de proposer des valeurs par défaut, et masque ceux qui sont vraiment spécifiques au traitement d'images (e.g. les points d'intérêt sont remplacés par les coins d'objets ou les jonctions entre lignes). Nous avons donc ajouté les concepts et relations nécessaires pour formaliser le modèle d'interaction proposé dans les sections précédentes (comme les post-traitements ou les relations hyperonymiques et méronymiques).

D'un point de vue conceptuel, les deux ontologies n'en forment qu'une seule globale qui présente deux points de vue sur la formulation : l'un utilisateur et l'autre expert de traitement d'images. L'ensemble des instances de l'ontologie de domaine créées par le système au cours de la formulation de l'utilisateur produit alors une ontologie d'application qui représente les connaissances métier du domaine considéré.

4.3. Traduction de la formulation utilisateur : le problème de l'ancrage

Une fois la formulation de l'utilisateur donnée, le système doit la traduire en une formulation pour le traitement d'images. Les termes employés par l'utilisateur dans la définition de la classe d'images doivent être associés à des données image ou à des termes du domaine du traitement d'images (concepts de l'*ontologie du traitement d'images*). Cette association est un problème à part entière appelé le problème de l'ancrage³. Pour le résoudre, il faut créer et maintenir une correspondance entre les symboles de l'utilisateur et les valeurs de traitement d'images [4] [5]. L'ancrage considéré ici doit être relativement direct. En effet, il faut garder une réelle correspondance entre les valeurs produites pour le traitement d'images et les valeurs fournies par l'utilisateur pour ne pas introduire de valeurs qui ne sont pas directement définies par celui-ci.

4.3.1. L'ancrage des termes définis par l'utilisateur

Encore une fois, le problème rencontré lorsque nous traitons de la description d'informations visuelles est la subjectivité de l'utilisateur. Une couleur par exemple n'est pas perçue de la même façon par différentes personnes : une même couleur peut être décrite comme étant plutôt verte ou plutôt bleue. Nous avons donc décidé que c'est l'utilisateur qui nomme ses couleurs (dans les termes qu'il emploie lorsqu'il parle des images de son domaine) puis qui renseigne les valeurs des descripteurs associés (par exemple teinte, luminance, saturation). Selon son niveau de compétence, il a la possibilité de donner des valeurs numériques directement, d'utiliser une palette de couleur, ou

encore d'utiliser des « patches » (des parties d'image) qu'il sélectionne à l'aide d'un outil graphique sur des images exemples qu'il aura choisies. Dans ce dernier cas, le système de formulation fournit au système de génération de programmes les « patches » qu'il utilisera pour calculer lui-même les paramètres qu'il considère intéressant pour la construction d'une solution (domaine de valeurs, seuils, histogrammes, moyennes, ...). Le même principe est applicable aux descripteurs de texture. Dans ce cas, le problème de l'ancrage n'est alors pas résolu par le système de formulation mais celui-ci donne au système de génération les moyens de le résoudre.

Nous pensons que cette méthode est la plus appropriée pour les caractéristiques dont la variabilité entre utilisateurs est trop importante. D'autres travaux proposent de choisir des termes dans une liste prédéfinie pour décrire l'apparence des régions comme la couleur ou la texture [5]. Ces termes sont associés directement à des valeurs numériques de descripteurs : une couleur orange sera associée à une teinte comprise entre [0.0,0.1] et une luminosité entre [0.5,1.0] (le domaine de valeurs de ces descripteurs étant compris entre 0 et 1). Nous pensons que la variabilité entre les utilisateurs est trop grande et qu'une telle approche n'est pas fiable dans notre cas. De plus, il est plus intéressant et efficace de garder dans la formulation la définition des valeurs de couleur ou de texture sous la forme de « patches » afin que le système de génération de programmes, ou un spécialiste du traitement d'images, puisse les utiliser pour extraire les valeurs dont il a besoin, plutôt que de calculer à partir de ces « patches » des valeurs numériques standard non réellement définies par l'utilisateur. Par ailleurs, ces « patches » donnent à l'utilisateur de la souplesse sur le choix des descripteurs puisqu'il peut décider de renseigner une propriété plus générale comme la radiométrie (qui comprend à la fois la colorimétrie et la texture) plutôt que de choisir explicitement les descripteurs de texture ou de couleur (choix qui peut s'avérer difficile pour un utilisateur novice du traitement d'images étant donné le peu de formalisation de la notion de texture). L'arborescence des descripteurs proposée dans l'ontologie permet une telle utilisation.

4.3.2. Les variables linguistiques

Les variables linguistiques permettent à l'utilisateur de garder une variabilité importante sur des caractéristiques de sa classe d'image (e.g. bruit, distance entre objets, ...) ou sur des propriétés qui sont par nature imprécises (e.g. circularité, compacité, ...) [5]. L'utilisateur choisit alors un terme dans une liste prédéfinie (e.g. très faible, faible, moyen, fort, très fort) pour donner une valeur à un descripteur.

L'utilisateur peut également utiliser des comparatifs. Ceci est très utile pour exprimer qu'il existe des caractéristiques qui peuvent être utilisées directement pour différencier des objets, sans donner de valeur quantitative. Deux types de comparaison sont possibles :

- une comparaison par rapport à un ou plusieurs objets (e.g. les cytoplasmes ont une surface plus grande que les noyaux).

3. *anchoring problem* ou *symbol grounding problem* en anglais.

- une comparaison par rapport à tous les autres objets (*e.g.* les cytoplasmes ont la plus grande surface).

Ces valeurs symboliques sont conservées dans la formulation au niveau expert de traitement d'images puisqu'elles se suffisent à elles-mêmes pour orienter les choix de conception de la solution.

4.3.3. Les inférences sur les termes des ontologies

Afin de réduire le fossé sémantique entre l'utilisateur et le système, des termes ont été fixés dans l'*ontologie utilisateur* pour décrire certaines propriétés comme la nature des frontières des régions. Ces termes sont traduits directement en termes du traitement d'images par des règles de production du premier ordre : par exemple, si les frontières ne sont pas marquées par une ligne ou un trait alors, au niveau expert de traitement d'images, le type de contour est « rampe ».

5. Conclusion

Le modèle de formulation d'applications de traitement d'images présenté dans cet article identifie et organise les informations nécessaires et suffisantes pour qu'un spécialiste ou un système de résolution automatique puisse concevoir une solution qui corresponde aux besoins d'un utilisateur.

Ce modèle a été formalisé sous forme d'une ontologie afin qu'il soit partagé par le système de formulation et le système de génération de programmes. Cette ontologie fixe alors la grammaire et le vocabulaire de la formulation qui est fournie au système de génération de programmes. Nous avons également défini des moyens d'interaction qui permettent de guider l'utilisateur dans sa formulation en tentant, d'une part, de réduire le fossé sémantique entre les connaissances de son domaine et les connaissances de traitement d'images nécessaires à la construction d'une solution et, d'autre part, d'objectiver le plus possible sa formulation. Pour cela le modèle s'appuie sur les post-traitements pour la spécification des objectifs, et sur la construction d'un arbre des objets et la caractérisation de la chaîne d'acquisition pour la définition de la classe d'images. La valuation des descripteurs fait appel à l'utilisation de « patches » et de variables linguistiques pour éviter les écueils dus à la quantification. À partir de là, le problème de l'ancrage est résolu en associant aux symboles définis par l'utilisateur des données image sans toutefois tout quantifier. Ce rôle est quelque fois laissé au système de génération de programmes qui choisit lui-même les descripteurs à calculer à partir des données de la formulation.

Le système de formulation est implémenté en Java et les ontologies en OWL DL, formalisme basé sur la logique de description. Elles comptent environ 300 et 350 concepts respectivement, une vingtaine de rôles et plus de 150 restrictions et sont implémentées avec le logiciel Protégé. Elles sont utilisées par le système par le biais d'une API OWL qui permet de les interroger et de les instancier. L'indépendance entre l'interface du sys-

tème de formulation et les ontologies permet une mise à jour automatique de ses différentes parties qui sont construites dynamiquement à partir des concepts de l'ontologie de domaine. L'interface permettant de valuer les descripteurs présente par exemple des outils graphiques adaptés aux descripteurs choisis grâce à une information stockée dans l'ontologie de domaine. L'indépendance entre l'interface et les ontologies est une qualité intéressante pour la maintenance et l'évolution du système. Par exemple, l'ajout d'un nouveau descripteur ou d'une nouvelle tâche consiste simplement en la création de nouveaux concepts dans les ontologies déjà existantes (par spécialisation des concepts déjà définis) et en la définition de restrictions sur leurs propriétés héritées.

Ce modèle et sa formalisation ont été éprouvés par des expérimentations de rétro-ingénierie de 14 applications qui couvrent des domaines et objectifs variés (*e.g.* reconnaissance de plaques d'immatriculation, restauration de vidéos, étude diachronique de l'évolution des sols, étude de la structure morphologique de céramiques, amélioration d'images astronomiques), et par ingénierie de nouvelles applications en biologie et robotique. La rétro-ingénierie nous a permis de montrer que la variété des informations supportées par le modèle permet de rendre compte de la variabilité des applications en évitant le biais qui aurait pu être introduit par une formulation *a priori* avec notre modèle. L'ontologie de traitement d'images proposée a également été directement utilisée dans le cadre d'une plate-forme destinée à des spécialistes de l'écologie [22]. Cette plateforme doit leur permettre de faire eux-mêmes l'analyse des vidéos capturées par des capteurs répartis sur un parc naturel. Ces travaux visent à automatiser la construction des applications dont ces scientifiques ont besoin pour leurs expérimentations.

Les perspectives de ce travail se déduisent des limites du système actuel. Ces limites concernent :

- **L'ergonomie.** L'ergonomie du système n'est pas assez développée pour ouvrir le système à des utilisateurs totalement novices du traitement d'images. Certains concepts restent difficiles à appréhender tels que. Toutefois, les premières expérimentations menées ont montré que la formulation faite avec le système n'induisait pas de surcharge cognitive importante pour l'utilisateur et qu'au contraire cela lui permettait de mieux appréhender la manifestation visualisation des concepts de son domaine et de revenir sur la production des images pour améliorer le rendu visuel de ces concepts.

- **Le cycle de formulation.** Le système ne met pas en œuvre véritablement un cycle de formulation itératif et incrémental par un couplage intime entre le système de formulation et le système de génération de programmes. Pourtant, l'obtention assez rapide de résultats et la visualisation des effets des choix opérés permettrait une meilleure appréhension des éléments de formulation par l'utilisateur. En particulier, le système laisse l'utilisateur seul pour produire une formulation qui est représentative et discriminante de la classe d'image. Il doit déterminer les caractéristiques qu'il juge d'une part représentatives de sa classe d'image, c'est-à-dire invariantes et exprimables par des valeurs quantitatives ou qualitatives, des intervalles de valeurs ou des

patches, et d'autre part discriminantes, c'est-à-dire que les caractéristiques suffisent à identifier l'information ciblée. Un cycle de formulation itératif apporterait pour cela une aide majeure.

- **L'ouverture.** Le système n'offre pas de mécanisme d'extension qui permettrait à l'utilisateur d'ajouter dynamiquement de nouvelles caractéristiques pour la description de sa classe d'image. Pour l'instant, cela nécessite d'ajouter la caractéristique dans Protégé puis de relancer le système où cette nouvelle caractéristique est néanmoins automatiquement intégrée.

- **La validation.** La validation du système ne s'est faite que sur quelques applications (non nécessairement simples) prises dans la littérature, ne couvrant que quelques domaines d'application et que quelques tâches. Or, ce nombre d'expérimentations n'est pas encore assez important pour garantir que le pouvoir de formulation de notre modèle est suffisamment général et représentatif pour tout domaine d'application.

Références

- [1] R. CLOUARD, « Une méthode de développement d'applications de traitement d'images », *Traitement du signal*, vol. 21, no. 4, pp. 277–293, 2004.
- [2] O. LEZORAY and H. CARDOT, “Cooperation of color pixel classification schemes and color watershed : a study for microscopic images,” *IEEE Trans. on Image Processing*, vol. 11, no. 7, pp. 783–789, 2000.
- [3] R. CLOUARD, A. ELMOATAZ, C. PORQUET, and M. REVENU, “Borg : A knowledge-based system for automatic generation of image processing programs,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 2, pp. 128–144, 1999.
- [4] S. CORADESCHI and A. SAFFIOTTI, “An introduction to the anchoring problem,” *Robotics and Autonomous Systems*, vol. 43, no. 2-3, pp. 85–96, 2003.
- [5] C. HUDELOT, N. MAILLOT, and M. THONNAT, “Symbol grounding for semantic image interpretation: from image data to semantics,” in *Proceedings of the Workshop on Semantic Knowledge in Computer Vision*, ICCV, Beijing, China, 2005.
- [6] T. MATSUYAMA, “Expert systems for image processing : knowledge-based composition of image analysis processes,” *Comput. Vision Graph. Image Process.*, vol. 48, no. 1, pp. 22–49, 1989.
- [7] C. LIEDTKE and A. BLÖMER, “Architecture of the Knowledge Based Configuration System for Image Analysis "Conny",” in *ICPR'92*, Den Haag, Netherlands, 1992, pp. 375–378.
- [8] V. CLÉMENT and M. THONNAT, “A knowledge-based approach to integration of image procedures processing,” *CVGIP : Image Understanding*, vol. 57, no. 2, pp. 166–184, 1993.
- [9] S. CHIEN and H. MORTENSEN, “Automating image processing for scientific data analysis of a large image database,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 854–859, 1996.
- [10] J. HASEGAWA, H. KUBOTA, and J. TORIWAKI, “Automated construction of image processing procedures by sample-figure presentation,” *ICPR*, vol. 86, pp. 586–588, 1986.
- [11] B. A. DRAPER, J. BINS, and K. BAEK, “ADORE : Adaptive object recognition,” in *ICVS*, 1999, pp. 522–537. [Online]. Available : cite-seer.ist.psu.edu/drapper99adore.html
- [12] A. NOUVEL, « Description de concepts par un langage visuel pour un système d'aide à la conception d'applications de traitement d'images », Ph.D. dissertation, Université Paul Sabatier – Toulouse III, 2002.
- [13] N. MAILLOT, M. THONNAT, and A. BOUCHER, “Towards Ontology Based Cognitive Vision,” *Machine Vision and Applications*, vol. 16, no. 1, pp. 33–40, december 2004.
- [14] V. BOMBARDIER, P. LHOSTE, and C. MAZAUD, « Modélisation et intégration de connaissances métier pour l'identification de défauts par règles linguistiques floues », *Traitement du Signal*, vol. 21, no. 3, p. 227–247, 2004.
- [15] C. TOWN, “Ontological inference for image and video analysis,” *Mach. Vision Appl.*, vol. 17, no. 2, pp. 94–115, 2006.
- [16] S. BLOEHDORN, K. PETRIDIS, C. SAATHO, N. SIMOU, V. TZOUVARAS, Y. AVRITHIS, S. HANDSCHUH, Y. KOMPATSIARIS, S. STAAAB, and M. G. STRINTZIS, “Semantic annotation of images and videos for multimedia analysis,” in *ESWC*, ser. LNCS, vol. 3532. Springer, 2005, pp. 592–607.
- [17] Y. SAIDALI, É. TRUPIN, J. LABICHE, N. BAUDOIN, and M. HOLZEM, “Incremental modelling and acquisition of image processing knowledge,” in *ICWI. IADIS*, 2002, pp. 184–190.
- [18] A. NOUVEL and P. DALLE, « Une approche interactive de définition d'ontologies image, » in *13^{ème} Congrès RFIA*, Angers, France, 2002, pp. 1023–1031.
- [19] M. JOLY, *Image et les signes : approche sémiotique de l'image fixe*. Nathan, Paris, 1994.
- [20] A. RENOUF, « Modélisation de la formulation d'applications de traitement d'images, » Ph.D. dissertation, Université de Caen/Basse-Normandie, SIMEM, septembre 2007.
- [21] G. SCHREIBER, B. WIELINGA, H. AKKERMANS, W. VAN DE VELDE, and A. ANJEWIERDEN, “CML : The CommonKADS Conceptual Modelling Language,” in *EKAW 94*, ser. LNCS, vol. 867. Springer, 1994, pp. 1–25.
- [22] D. RANDELL, Z. CUI, and A. COHN, “A spatial logic based on regions and connection,” in *KR'92. Principles of Knowledge Representation and Reasoning : Proc. of the 3th International Conference*, San Mateo, California, 1992, pp. 165–176.



Arnaud **Renouf**

Arnaud Renouf est docteur de l'Université de Caen. Ses travaux de recherche au sein de l'équipe Image du GREYC portent sur la modélisation de la formulation d'applications de traitement d'images et sur la construction d'un système de formulation d'applications dédié à des utilisateurs novices.



Régis **Clouard**

Régis Clouard est docteur de l'Université de Caen. Depuis 1995, il est maître de conférences de l'école d'ingénieurs ENSICAEN. Ses travaux de recherche au sein de l'équipe Image du GREYC portent sur la modélisation des connaissances en traitement d'images en vue de leur capitalisation et de leur opérationnalisation dans des systèmes à base de connaissances.



Marinette **Revenu**

Marinette Revenu est ingénieure de l'ENSI de Caen et docteur en informatique de l'Université de Paris VI. Professeure à l'ENSICAEN, elle est responsable de la spécialité informatique de l'ENSICAEN et de l'équipe Image du GREYC.

Le fil conducteur de son activité de recherche est attaché à l'acquisition, la modélisation et l'exploitation de connaissances, appliquées principalement à l'imagerie numérique. Son but est d'extraire automatiquement des connaissances numériques a priori pour les injecter dans le processus de segmentation ou d'interprétation d'images.

