

# Modélisations orientées-objet en traitement d'images : exemples et évaluations

Marc Brelot, Gilles Privat

*marc.brelot@cnet.francetelecom.fr, gilles.privat@cnet.francetelecom.fr*  
France Telecom, CNET/DTL/ASR-Grenoble  
BP 98, 38243 Meylan Cedex

## RÉSUMÉ

## ABSTRACT

La modélisation algorithmique du traitement d'images doit évoluer pour permettre son intégration dans des systèmes de communication multimédias répartis ouverts. La modélisation orientée-objet, outre ses atouts de modularité, et de généralité, s'adapte parfaitement à des représentations basées sur les objets issus d'une analyse de l'image. Cet article évalue, à partir de l'exemple de représentations multirésolution, les compromis nécessaires sur la granularité temporelle et spatiale des objets image, compte tenu de la complexité additionnelle en mémoire et temps d'exécution introduite par leur encapsulation en modélisation objet.

Image processing algorithms should open up from their classical numerical-procedural background to become integrated in open distributed multimedia systems. Object-oriented modeling, besides its modularity and genericity benefits, is well matched to non-waveform object-based representations as obtained from image analysis. We assess, from the example of multiresolution representations, key trade-offs in the spatial and temporal granularity of objects, taking in to account the time and memory overhead resulting from their object-oriented encapsulation.

## 1 Introduction

Le traitement d'images reste marqué par l'influence de l'algorithmique numérique, qui lui a fourni ses principaux modèles de représentation aux niveaux mathématique aussi bien qu'informatique. Si ces modélisations conservent toute leur pertinence pour des traitements classiques pris isolément, les applications de communication d'images, fixes ou animées, qui utilisent ces traitements élémentaires, font apparaître de nouveaux besoins et de nouvelles contraintes par rapport aux modèles de représentation.

L'élargissement de la problématique est le suivant : les fonctions classiques du traitement d'image (analyse, restauration, compression, etc.) sont de type procédural *fermé*, c'est à dire qu'elles se ramènent, quelle que soit leur complexité, à des transformations closes d'un espace de représentation d'entrée vers un espace de sortie. Il s'agit de les intégrer dans des applications de type transactionnel *ouvert* où l'aspect temporel, interactif ou réactif, des traitements est irréductible<sup>1</sup>. Un exemple type pourrait être la *navigation interactive* dans des espaces visuels hétérogènes et physiquement répartis, qui peuvent être partiellement naturels ou synthétiques, enregistrés dans des serveurs ou acquis en temps-réel.

De ce point de vue, la modélisation orientée-objet est bien plus qu'une technique de génie logiciel destinée à améliorer la productivité des programmeurs. Elle est le nouveau paradigme qui permet le saut qualitatif d'expressivité nécessaire à la modélisation complète de systèmes en interaction avec leur environnement, qui

dépassent l'expressivité offerte par les modèles procéduraux[1].

Une division du travail classique permettrait de voir les choses ainsi : on modélise en "objets", pour une application de communication multimédia par exemple, tout ce qui est au dessus du niveau des images ou séquences d'images, qui peuvent ainsi bénéficier des facilités génériques d'un environnement multimédia réparti. A l'intérieur de ces contenus image élémentaires, vus comme des boîtes noires du point de vue de la modélisation objet, l'on reste libre d'utiliser une modélisation procédurale/numérique classique (matrice de pixels, signal temporel...).

Il est intéressant à plusieurs titres de chercher à généraliser l'utilisation de la représentation objet à tous les niveaux, ce qui revient à faire descendre l'utilisation des modèles objet à l'intérieur des images ou séquences d'images.

- la représentation objet peut être associée à des entités élémentaires de l'image, qui peuvent être la base de nouvelles fonctionnalités d'accès à l'information par le contenu (§2).
- ces objets élémentaires de niveau intra-image peuvent être de nature et de granularité très variées (§3)
- on peut bénéficier à ces différents niveaux de la souplesse du modèle objet, mais ces avantages doivent être évalués en regard des coûts additionnels induits par la représentation (§4 et 5).

## 2 Représentations et fonctionnalités basées-objets en images.

Si l'extraction d'information de niveau structurel ou symbolique est depuis toujours l'objectif de l'analyse d'image, l'idée d'incorporer cette information structurelle

<sup>1</sup> Alors que l'aspect temporel éventuel des traitements classiques peut être "banalisé" en l'intégrant comme une dimension supplémentaire des espaces de représentations ; par exemple un traitement temps-réel effectué sur une séquence d'images peut être vu comme un mapping a-temporel dans un espace 3D (2D+T), et modélisé comme tel.

dans la *représentation* de contenus d'image naturelle animée, tels qu'ils peuvent être transmis ou stockés dans des applications de communication, est relativement nouvelle.

MPEG-4 [2] a ouvert la voie à cette idée qui doit permettre d'offrir des fonctionnalités de manipulation et de composition interactive de ces contenus, à l'instar de ce qui est possible sur de l'animation d'origine synthétique, et plus généralement aboutir, à terme, à la banalisation complète d'objets images de toute granularité dans des environnements répartis hétérogènes.

Les possibilités d'analyse de contenus naturels génériques imposent le plus souvent en pratique une limitation à des descriptions issues de segmentation dans le plan de l'imageur, par exemple basées sur une décomposition du contenu des images en termes de régions homogènes selon un certain critère (mouvement, texture, contours, etc.). On parle alors de codage par régions 2D. La possibilité de remonter au contenu 3D de la scène et à de véritables objets au sens géométrique reste impossible dans le cas général, quand on ne dispose pas d'information a priori sur le contenu.

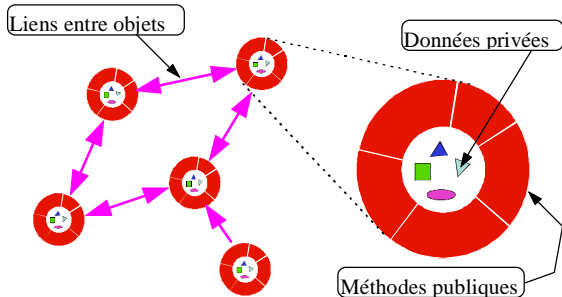


Figure 1 : Modélisation orientée-objet

Les fonctionnalités accessibles à partir de ce type de représentations peuvent être l'extension de ce qui est offert pour les objets de contenu multimédia qui les englobent, dans une application de communication. Ces fonctionnalités sont basées soit sur l'existence de *liens* entre les objets individuels, soit sur l'existence de *transformations* accessibles sur ces objets (figure 1).

La représentation adoptée devra permettre de rendre explicites et accessibles au niveau supérieur de l'interface utilisateur ou de l'application ces liens et ces transformations. La modélisation orientée-objet rend ceci assez facile : les transformations sont des méthodes publiques de l'objet correspondant, et les liens peuvent être des pointeurs vers d'autres objets.

Le paragraphe suivant donne différents exemples d'objets que l'on peut ainsi décrire séparément avec leurs inter-relations dans une image.

### 3 Exemples de représentations d'objets primitifs et de leurs relations.

#### 3.1 Modèle de vérification MPEG4

Une région élémentaire issue de segmentation peut être codée dans un objet primitif (Video Object Plane du modèle de vérification MPEG-4 [3]) de façon très classique avec un codage séparé des paramètres de texture (codage type DCT avec remplissage au rectangle englobant), forme (chain code, ou couche alpha compressée) et mouvement. Le

schéma de codage correspondant est alors assez proche d'un schéma de codage hybride classique, la différence résidant dans l'utilisation d'une estimation de mouvement basée-objet et d'un codage de forme.

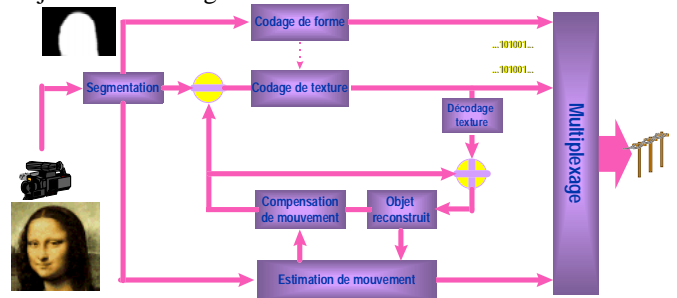


Figure 2 : Schéma de codage par régions 2D

#### 3.2 Relations entre objets

Une représentation de la structure de scène doit permettre de décrire toute la variété des relations entre objets identifiés dans la scène à coder, suivant leurs niveaux de description et de granularité.

Quelques exemples de telles relations, suivant les niveaux de représentation et de granularité des objets analysés sont les suivants :

- adjacence temporelle ou spatiale de régions (Figure 3)
- inclusion spatiale ou temporelle d'objets dans des objets composites
- recouvrement d'objets graphiques et bit-maps
- hiérarchies de description multirésolution (par exemple pyramide laplacienne)
- réseau sémantique d'interprétation d'une image

Il est utile de séparer la structure de composition des objets des objets eux-mêmes : cela a été proposé comme la description d'entrée d'un moteur de composition [4]: un plan graphe encapsule séparément cette structure, les liens sont décrits entre les objets noeuds de ce graphe et entre les noeuds et les objets contenu, et non entre les objets contenu eux-mêmes.

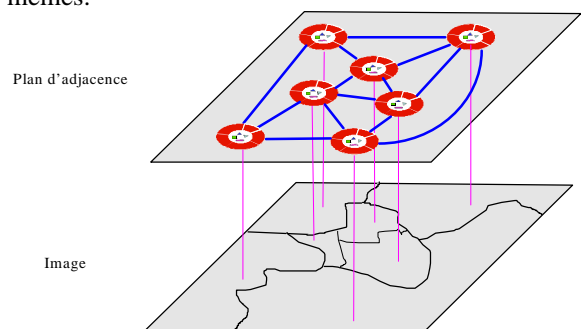


Figure 3 : Représentation objet d'adjacence de régions

#### 3.3 Exemples d'encapsulations objet

L'exemple de représentations multirésolutions de régions où chaque couche spatiale ou temporelle est représentée comme un objet séparé est évalué plus en détail ci-dessous (§5, cf. aussi [5]).

D'autres modèles de représentations moins classiques pourraient également être encapsulés en objets séparés, comme par exemple des codages fractals par IFS, si ils étaient calculés localement sur un objet.

Des objets de granularité encore plus fine pourraient par exemple correspondre aux triangles d'une triangulation de Delaunay, où à des blocs carrés de pixels "à la MPEG2", le cas limite étant celui où l'on modéliserait les pixels individuels en tant qu'objets! Il est clair que l'on perdrait alors la pertinence de la correspondance entre le niveau objet de la modélisation informatique et les objets de contenu qui sont la base des fonctionnalités pour les applications envisagées.

### 4. Evaluation quantitative des compromis dans le choix d'une représentation objet.

Le but est d'évaluer en fonction d'un certain nombre de paramètres le choix optimum du niveau de modélisation objet en fonction des caractéristiques de ce qui se trouve en dessous ("intra-objet") ou au dessus ("inter-objet") de ce niveau :

Niveau inter-objet	Niveau intra-objet
concurrence asynchrone	parallélisme synchrone
gros grain	grain fin
dégradation graduelle possible	contraintes temps-réel "dures"
optimisation tps moyen de calcul	optimisation "pire cas" du tps de calcul

Les 2 niveaux de modélisation permettent de tirer parti de ces différentes propriétés pour en tirer conjointement la performance optimale.

Le paramètre principal dont dépendra l'optimisation est la taille relative des objets élémentaires. Une quantification absolue de cette taille (par exemple en équivalent nombre de pixels), n'a pas forcément de sens dans la mesure où la résolution initiale de l'objet peut être modifiée à volonté. Le nombre total d'objets utilisés pour représenter la scène à un instant donné est une indication plus utile.

Par ailleurs, les paramètres annexes dont dépend l'optimisation sont

- coût additionnel ("overhead") en mémoire de l'encapsulation objet
- "overhead" temporel de l'encapsulation objet
- degré maximal de parallélisme inter-objet (nombre de processeurs).
- degré maximal de parallélisme intra-objet
- "overhead" de communication inter-objet

La mise en place d'une première plate-forme de tests a permis de mesurer l'overhead mémoire ainsi que l'overhead d'exécution introduit par une représentation objet.

Pour tous les types de représentations, il est possible alors d'évaluer une granularité minimum des objets (en termes de taille des données, et de temps d'exécution) en dessous de laquelle l'encapsulation objet entraîne un surcoût intolérable dans un contexte temps réel et/ou contraint en mémoire.

### 5. Evaluation dans le cas de représentations multi-résolutions

Dans cette exemple, un schéma de codage multirésolution très primitif (le but n'étant pas d'évaluer le schéma de codage lui-même), de type pyramide laplacienne, a été utilisé comme base d'évaluation. La représentation choisie outre le fait de séparer explicitement la forme et la texture, encapsule chaque niveau de résolution dans un objet séparé, comme le montre la figure ci-dessous :

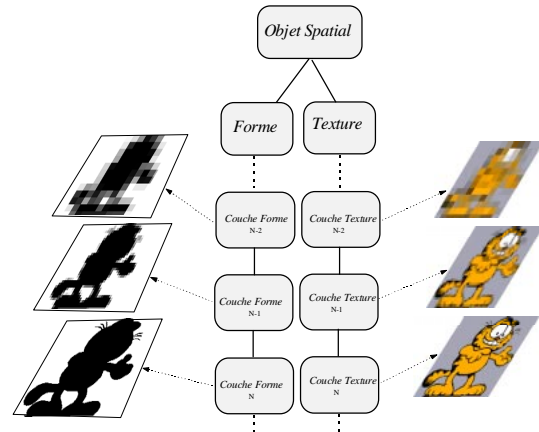


Figure 4 : Objets correspondant à différentes couches de résolution spatiale

De plus les instances temporelles de cette objet peuvent être également encapsulées séparément comme le montre la figure ci-dessous où tous les instants ont été décomposés.

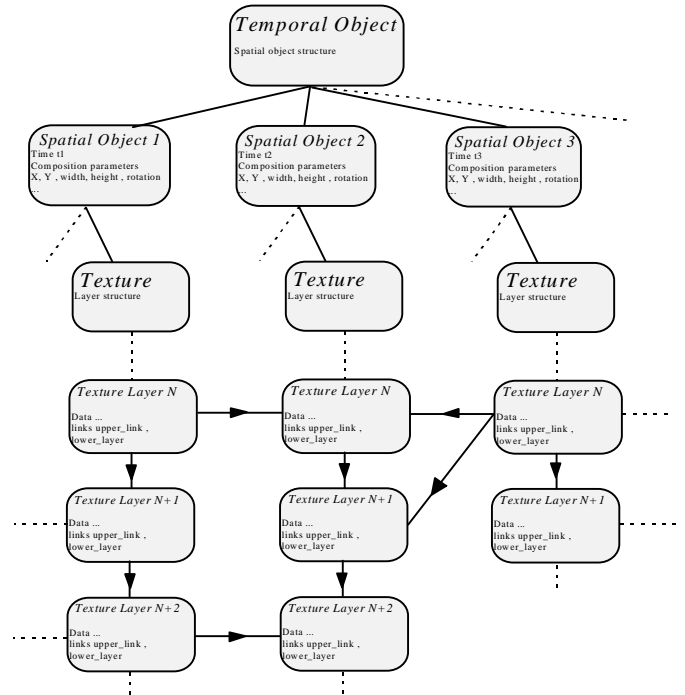


Figure 5 : Décomposition spatio-temporelle d'un objet

On cherche à travers cet exemple à soulever les problèmes de compromis à faire entre le choix du niveau de granularité des objets aussi bien dans le domaine spatial que

temporel par rapport aux fonctionnalités que ce type de représentation permettent implicitement.

En effet cette représentation apporte entre autres les fonctionnalités suivantes :

- Adaptation des informations codées à la résolution du décodeur
- Adaptation du débit de transmission des informations à la bande passante disponible.
- Auto-adaptation du décodage à la puissance du décodeur.

L'adaptation du décodage à la puissance du décodeur est réalisée de façon implicite de par le graphe de dépendance entre les différents objets couches, si toutefois on dispose de possibilités d'exécution concurrente.

Un exemple d'adaptation du débit à la bande passante ou à la capacité de traitement du décodeur est donné par les figures ci-dessous, où la région correspondant au sujet principal ne subit pas de dégradation, seul le fond étant décodé à une résolution inférieure pour s'adapter à la limitation de ressources. Dans cet exemple, des priorités ont été assignées sur chaque objet, et le décodage a tenu compte de ces priorités pour l'adaptation.



Figure 6a : Décodage à résolution uniforme



Figure 6b : Décodage à résolution adaptée par régions

Notre première plate-forme de tests a permis de mesurer le coût additionnel ("overhead") mémoire ainsi que l'overhead d'exécution introduits par cette représentation.

Par exemple pour une représentation où chaque instance temporelle d'un objet est encapsulée dans 7 objets distincts correspondant à des couches de résolution spatiale (les

fonctionnalités fournies par cette représentation étant intéressantes pour  $N > 3$ ) la granularité minimum pour une couche de données doit être supérieure à 50 pixels si on ne veut pas dépasser un overhead temporel de 10%.

Ceci signifie également que si le décodeur est capable de traiter 25 images secondes sans encapsulation objet, il pourrait traiter 22 images / s ( chute de 10 % ), en encapsulant dans 7 couches les instances temporelles des différents objets

Il est clair que si les 25 instances par secondes sont gardées, le décodeur devant respecter des contraintes de temps réel ne pourrait effectuer le calcul de la meilleure résolution pour chaque image.

Ainsi avec 7 couches par objets et par instances temporelles, ce type de représentation donne accès à un niveau de fonctionnalités intéressant sans pour autant coûter trop cher en mémoire et en temps.

Bien sur ces résultats n'ont qu'une valeur relative puisqu'ils dépendent directement de toutes les caractéristiques de la plate-forme sur laquelle on fait les mesures (système d'exploitation, langage de programmation, machine virtuelle éventuelle) et aussi des algorithmes particuliers de décodage implémentés dans les objets couches eux-mêmes.

## 6 Conclusion

Les ordres de grandeur fournis par ces évaluations indiquent clairement que l'utilisation de modélisations objet en traitement d'images n'est pas excessivement pénalisante en termes d'efficacité de mise en oeuvre. Si l'on conserve une granularité moyenne des objets, le niveau intra-objet permet encore d'exploiter un parallélisme/pipeline explicite à grain fin de manière classique, tandis que le niveau inter-objet permet laisser jouer des mécanismes de concurrence implicite plus génériques et adaptables. C'est cependant sur le plan des fonctionnalités que les services génériques apportés par les couches de gestion d'objets sous-jacentes sont le plus déterminants, et permettront d'intégrer véritablement le traitement d'images de manière transparente et délocalisée dans les systèmes multimédia répartis.

## Références

- [1] Peter Wegner "Interactive Foundations of Object-based Programming", *Computer*, vol. 28 n° 10, Oct. 1995
- [2] [http://drogo.cselt.it/mpeg/mpeg\\_4.htm](http://drogo.cselt.it/mpeg/mpeg_4.htm)
- [3] Video Verification Model Version 7.0", ISO/IEC JTC1/SC29/WG11 document N1642, 39th MPEG meeting, April 1997
- [4] Marc Brelot, Gilles Privat "Représentation basée-objet de structures de scènes pour la composition d'images en représentation 2.5D", 3<sup>e</sup> colloque "Compression et Représentation des Signaux Audiovisuels, CORESA'97, Mars 1997
- [5] M. Brelot G. Privat "Object encapsulation of multiscale image region representations", 2<sup>nd</sup> European Conference on Multimedia Applications, Services and Techniques, *ECMAST'97*, Milan, 21-23 Mai 1997