Une méthode autonome de ciblage de l'optimisation d'un système de détection d'objets par analyse de la responsabilité [1]

An autonomous method of optimization targeting for an object detection system based on responsability analysis

Rémi Landais [1], Laurent Vinet [2] et Jean-Michel Jolion [3]

[1] ENST-TSI, LTCI CNRS, 46 rue Barrault, 75013 Paris
landais@tsi.enst.fr
[2] DRE/DCA, INA, 4 Avenue de l'Europe, 94366 Bry-Sur-Marne Cedex
lvinet@ina.fr
[3] LIRIS, INSA Lyon, Bat J. Verne, 20 Avenue Albert Einstein 69621 Villeurbanne Cedex
Jean-Michel.Jolion@insa-lyon.fr

Manuscrit reçu le 5 juillet 2006

Résumé et mots clés

Les systèmes d'extraction d'objets sont mis à mal par la diversité de ces derniers. Leur adaptation est donc nécessaire pour maintenir des performances équivalentes quelle que soit la nature des objets sur lesquels ceux-ci sont appliqués. S'attachant plus particulièrement, dans l'optique de cette adaptation, à la tâche d'optimisation du paramétrage de ces systèmes, nous proposons dans cet article une méthode originale de ciblage de l'optimisation aux seuls paramètres des opérateurs du système estimés responsables des différentes catégories d'erreurs produites par le système. Cette méthode s'appuie alors sur deux analyses distinctes. La première porte sur les performances du système considéré et permet d'extraire les différentes catégories d'erreur déjà mentionnées. La seconde concerne le fonctionnement des différents opérateurs composant le système et donne lieu à la détermination d'un opérateur responsable pour chaque catégorie d'erreur. Une application de cette méthodologie à un système de détection de texte est par ailleurs détaillée.

Contrôle, optimisation, principe d'autonomie, diagnostic de responsabilité, détection d'objets, textes vidéo.

Abstract and key words

Object extraction systems performances are not homogeneous over different corpora because objects can take many different aspects within such sets. An adaptation of these systems is thus required in order to maintain equal performances over every kind of object the system may be applied on. Focusing on the issue of parameters optimization, a method has been developed to restrict optimization to parameters of operators which compose the system, responsible for the different categories of errors produced by the system. Two stages are involved in our method. The first one is dedicated to the analysis of the system performances and leads to the extraction of the different error categories already mentionned. The second one relates to the analysis of the behavior of the different operators, leading to extract a single operator responsible for each error category. Experiments have been carried out over a video text detection system.

Control, optimization, autonomy principle, responsibility diagnostic, object detection, video texts.

1. Ce travail a été réalisé dans le cadre d'une Bourse CIFRE entre l'Institut National de l'Audiovisuel et le laboratoire du LIRIS.

1. Introduction

Nombreuses sont les équipes de recherche qui reconnaissent la nécessité de faire face à l'accroissement continu du volume de documents audiovisuels numériques. Parmi l'ensemble des informations qu'il est possible d'extraire de la modalité visuelle pour aider à la documentation, la présence/absence d'objets particuliers (textes, visages, ...) constitue alors une source d'indices aisément exploitable.

La construction de systèmes capables d'extraire automatiquement de tels objets est complexe, principalement en raison de la diversité extrême des objets (du point de vue de leur forme, de leur couleur, de leurs caractéristiques texturales, ...). Puisqu'il est « *impossible* » de construire des systèmes d'extraction d'objets effectifs sur n'importe quel type d'objet, une solution consiste à les adapter relativement à chaque contexte d'application, soit à les modifier pour obtenir des résultats satisfaisants sur un corpus d'application particulier.

Un système d'extraction d'objets peut être vu comme une séquence de modules/d'opérateurs auxquels sont associés des paramètres. L'adaptation peut alors prendre deux formes selon que les modifications envisagées portent sur la nature des opérateurs composant la séquence ou sur les paramètres qui leur sont attachés. Dans cette étude, seul le second mode de modification sera envisagé.

L'enjeu de la méthodologie présentée dans cet article est alors de s'affranchir des limitations des systèmes d'optimisation existants qui prennent généralement en compte l'ensemble des paramètres du système, et ceci en déterminant automatiquement, préalablement à l'application de l'optimisation, la nature des paramètres qu'il convient d'ajuster.

La méthodologie développée se structure autour de deux phases distinctes. Le premier postulat de cette étude est ainsi qu'il existe non pas un unique, mais plusieurs paramétrages optimaux, le nombre de ces paramétrages dépendant alors du nombre de classes d'erreurs différentes qu'il est possible d'isoler. La première phase de la méthodologie relève donc de l'extraction de ces différentes classes d'erreur. La seconde phase a par la suite pour objectif de dresser un diagnostic relativement à chacune de ces classes (dites classes de comportements du système) permettant de déterminer pour chacune d'entre elles quel est l'opérateur responsable de l'erreur. Cette phase de diagnostic, ou de ciblage, permet ainsi *in fine* de restreindre l'optimisation aux seuls paramètres de cet opérateur responsable, ceci permettant notamment de réduire la complexité de cette phase d'optimisation (en termes d'espace de recherche à parcourir).

L'organisation de cet article est alors la suivante : après un survol rapide du domaine du contrôle des systèmes de vision, la méthodologie sera exposée en mettant en avant les deux phases mentionnées ci-dessus : l'analyse des comportements du système, et la phase de diagnostic. La dernière partie portera sur les expérimentations menées relativement à un objet particulièrement intéressant dans le contexte de l'indexation audiovisuelle, l'objet texte, en prenant pour système cible celui détaillé dans [17].

2. Contrôle des systèmes de vision : état de l'art et présentation de la méthodologie

Le contrôle des systèmes de vision comprend l'ensemble des tâches liées à la mise au point de ces systèmes. Un outil de contrôle peut alors être considéré comme un *méta-système*, comme la chaîne de montage et de maintenance d'un système de vision. Nous adoptons par la suite une représentation simplifiée du contrôle, celui-ci étant envisagé à chacune des étapes du cycle de vie d'un système : lors de sa conception, de son exécution ou encore dans le cadre de sa réparation dans le cas où les résultats produits par le système sont insuffisants.

La conception débute par une étape de « formulation » lors de laquelle une requête de l'utilisateur initie la construction du système en spécifiant la nature de la tâche à effectuer ainsi que les contraintes lui étant spécifiques (formes des images, contraintes relatives à l'évaluation, etc.). Les systèmes de vision sont composites : la réalisation de la tâche définie requiert d'appliquer en cascade un ensemble d'opérateurs de traitement d'images. La phase de planification s'applique alors à proposer le plan d'actions (*i.e* : une séquence d'opérateurs) adapté à l'objectif énoncé.

Notre méthodologie d'adaptation relève uniquement de la phase de « réparation » (ou « contrôle de l'exécution ») et l'état de l'art présenté ici se limitera ainsi à cette seule tâche.

2.1. Contrôle de l'exécution

Le contrôle de l'exécution implique d'une part l'évaluation des résultats du système conformément aux contraintes de performances établies lors de la formulation du problème ; et d'autre part la mise en oeuvre de modifications en vue de corriger les erreurs constatées.

Deux modes de modification du système sont alors envisageables : l'*optimisation* de ses paramètres ou la *re-planification*, qui consiste à remettre en cause la validité des choix effectués lors de la planification en substituant des opérateurs ou des séquences d'opérateurs par des éléments équivalents. On distingue par ailleurs deux « philosophies » de contrôle. Les méthodes dites « à base de connaissances » reposent sur une modélisation manuelle et intuitive des liens de causalité entre les connaissances « haut-niveau » relatives au contexte métier (l'évaluation de la qualité des résultats) et les connaissances liées au traitement, dites de « bas-niveau » (fonctionnement des opérateurs). D'autre part, les méthodes « autonomes » mettent l'accent sur l'accélération (et/ou l'automatisation) de l'optimisation. Concernant les méthodes du premier type, dans [15] les systèmes sont considérés comme étant capables de s'auto-ajuster

au contexte. Cet auto-ajustement repose sur la définition de règles d'évaluation et de règles d'ajustement. Le champ des modifications envisagées embrasse l'optimisation des paramètres tout comme la re-planification. La sélection de la modification à apporter au système en cas d'échec est conditionnée par un ensemble de règles de production. Certaines règles de production permettent par ailleurs la transmission de l'évaluation à des opérateurs précédents ou de niveau supérieur dans la hiérarchie [12].

Dans le système BORG [3], les sources de connaissances utilisées lors de la planification (comme, par exemple, la décomposition d'une tâche) sont qualifiées selon leur aptitude à résoudre un certain type de problème dans un certain contexte. Ces taux d'aptitude sont utilisés comme mesure de cohérence du système : leur évaluation détermine si le système a atteint une impasse. Dans cette situation, la construction du plan reprend depuis le niveau d'abstraction supérieur (le plan d'action construit est hiérarchique). Un autre mode d'évaluation entre en jeu dans le processus de contrôle : à chaque décomposition est attachée une règle d'évaluation. Si tous les critères d'évaluation des sous-tâches de chaque décomposition sont remplis, l'exécution est un succès. Dans le cas contraire, la décomposition entière est supprimée et une phase de re-planification est exécutée.

Dans le cadre du raisonnement par cas [8], l'adaptation consiste à affiner le plan d'action choisi initialement par proximité avec l'application envisagée. Cette adaptation consiste à rechercher récursivement, pour les sous-tâches du plan qui ne satisfont pas les critères retenus, d'autres décompositions plus adaptées dans la base de celles disponibles.

Les méthodes « *autonomes* » reposent sur une méthodologie « intelligente » d'optimisation. Dans [13, 16] par exemple, l'optimisation est limitée aux seuls paramètres auxquels le système est estimé être sensible (paramètres **influents**). Par ailleurs, dans [16], un plan d'expériences permet de réduire encore, lors de l'optimisation, le nombre de combinaisons à tester en vue de trouver le paramétrage optimal. L'objet de ces méthodes est donc essentiellement combinatoire : réduire la taille de l'espace des paramètres à parcourir en vue de déterminer le paramétrage optimal.

2.1.1. Limitations des méthodes existantes

Concernant les méthodes dites « à base de connaissances », la principale restriction réside dans la difficulté d'acquérir les connaissances nécessaires et de choisir un formalisme de représentation de celles-ci suffisamment évolutif pour permettre d'ajouter facilement de nouvelles connaissances. Un premier objectif de notre méthodologie sera ainsi de circonscrire autant que possible le volume et la nature des connaissances *a priori* nécessaires (principe d'autonomie) et ceci en considérant uniquement les informations produites par le système lui-même (ses entrées/sorties).

Dans le cas des systèmes « *autonomes* », la nature des modifications envisagées relève de l'analyse de l'influence des différents paramètres. La principale limitation de ces systèmes relève du caractère « *aveugle* » de l'optimisation proposée : s'il existe un ciblage de l'optimisation aux paramètres influents, aucune analyse du lien entre le caractère influent d'un paramètre et la notion de **responsabilité** (relativement à l'erreur constatée) de l'opérateur de traitement auquel le paramètre est attaché, n'est proposée. En sus de l'autonomie, cette notion de **responsabilité** est tout aussi centrale dans notre méthodologie d'adaptation et nous verrons ainsi par la suite comment cette dernière s'articule autour de ces deux notions.

2.2. Présentation de la méthodologie

2.2.1. L'optimisation imposée par la contrainte d'autonomie

La méthodologie d'adaptation proposée se limite à l'optimisation (la re-planification n'est donc pas considérée) principalement en raison du coût élevé de la re-planification. En effet, pouvoir modifier la structure d'un système en substituant certains opérateurs ou séquences d'opérateurs par des opérateurs ou séquences équivalentes impose l'utilisation d'une base d'opérateurs (une bibliothèque typiquement) et l'existence d'un framework de programmation particulièrement efficace pour permettre les substitutions évoquées.

2.2.2. Représentation d'un système de détection d'objets

Une représentation **séquentielle** des systèmes de détection d'objets est adoptée : chaque système est considéré comme une suite de modules/opérateurs. Les cas des boucles, du parallélisme,... ne sont pas envisagés. Par ailleurs, nous imposons que les opérateurs entrant dans la séquence produisent des résultats de même nature que les résultats finaux du système (une image dans laquelle sont isolées les zones contenant un objet, un ensemble de boîtes englobantes ou tout du moins une image) ; et ceci pour satisfaire les contraintes liées à l'analyse du fonctionnement des différents opérateurs lors de l'établissement du diagnostic de responsabilité. Par ailleurs, et pour la même raison, nous supposons disponibles les sorties de ces opérateurs.

2.2.3. Organisation de la méthodologie

1. **L'analyse des comportements :** l'enjeu de cette phase est de parvenir à distinguer les différentes catégories d'erreurs produites par le système. Pour ce faire, l'idée est de définir un ensemble de **mesures d'évaluation** qui permettront, en comparant les résultats du système avec les vérités terrain, de produire des **vecteurs de performances**. Une étape de **clustering** relativement à ces vecteurs permettra finalement d'isoler les différents comportements.

2. **Le diagnostic de responsabilité :** le système de détection est représenté comme une séquence d'opérateurs. L'enjeu du diagnostic de responsabilité est alors de déterminer, pour chaque comportement extrait lors de l'analyse précédente, l'opérateur

responsable de l'erreur constatée. Un ensemble de caractéristiques visuelles (taille, forme, texture, ...) est alors défini. Les performances des différents opérateurs sont étudiées relativement à ces caractéristiques. La mise en relation des représentations des objets composant les différents comportements selon ces mêmes caractéristiques avec cette analyse des performances des opérateurs permet alors de calculer pour chaque opérateur un **indice de responsabilité** relativement à chaque caractéristique. Une méthode d'intégration des différents indices obtenus détermine finalement pour chaque comportement l'opérateur responsable.

3. Analyse des comportements

3.1. Motivations

L'*analyse des comportements* repose sur le postulat selon lequel il existe autant de paramétrages optimaux du système de détection que celui-ci produit, sur le corpus d'adaptation, de classes de comportement différentes (*cf.* définition 1).

Définition 1. *Une classe de comportement correspond à un ensemble d'objets sur lesquels un système produit des performances équivalentes.*

L'évaluation des systèmes de détection d'objets se limite généralement à comptabiliser les objets correctement détectés et ceux que le système n'est pas parvenu à extraire. Cette solution s'avère insuffisante dans la perspective de l'optimisation. En effet, lorsqu'une unique classe correspondant aux échecs est constituée, il n'existe pas nécessairement de cohérence en ce qui concerne la manifestation de l'erreur mesurée. Ceci va alors à l'encontre du postulat suivant.

Postulat 1. *La nature des paramètres à modifier pour optimiser un système dépend d'une qualification précise de l'erreur constatée.*

L'existence de ces différents comportements est illustrée dans le cas du système de détection de texte utilisé [17] dans les images de la figure 1, dans lesquelles apparaissent les boîtes englobantes des textes détectés par le système (les fausses alarmes qui, comme nous le verrons plus loin, font l'objet d'un traitement *a posteriori* n'apparaissent pas).
Il est ici montré que le système de détection se comporte différemment selon la nature des textes contenus dans les images à traiter. Sans prendre d'*a priori* sur le mode de fonctionnement du système, il est difficile de déterminer les classes d'objets correspondant aux différents comportements. Les images de la figure 2 donnent ainsi une illustration des risques encourus en assimilant les classes visuelles d'objets avec les classes de comportements : le même texte apparaissant dans deux images successives du flux peut aboutir à des résultats différents du système. En conséquence, une méthode d'extraction des comportements par clustering des vecteurs de performances apparaît comme la solution la plus fiable. Nous verrons par la suite comment ces vecteurs sont produits par comparaison des résultats du système avec une vérité terrain, en détaillant la construction de cette dernière, ainsi que les mesures d'évaluation développées.

3.2. Construction des vérités terrain

Les vérités terrain contiennent les positions, dans chaque image, de tous les objets recherchés. Une spécificité de l'objet texte réside dans les différentes granularités auxquelles il peut être détecté (lettres, mots, lignes ou blocs de texte). Un même système peut donc produire des résultats corrects à différents niveaux. Pour ne pas le pénaliser en choisissant un unique niveau, une hiérarchie de zones englobantes (bloc, lignes et mots) est conservée dans la vérité terrain pour l'ensemble des textes relevés.

3.3. Métrique d'évaluation d'un système de détection

3.3.1. Revue des métriques existantes

L'évaluation d'un système de détection d'objets revient à comparer, pour chaque image, l'ensemble des zones détectées par le

a) Fusion b) Un seul mot détecté c) Détection en mots d) Détection trop large

Figure 1. Des comportements différents produits par le système de détection de texte.

a) b)

Figure 2. Les résultats du système diffèrent sur deux images successives.

système avec celui conservé dans la vérité terrain. Le principe le plus simple consiste à comparer les taux de recouvrement mutuels entre les zones détectées par le système et les zones de la vérité terrain [2, 10, 13] :

$$\frac{A(G_j \cap D_i)}{A(D_i)} \quad \text{et} \quad \frac{A(G_j \cap D_i)}{A(G_j)} \qquad (1)$$

où D_i et G_j désignent respectivement une zone détectée et une zone de la vérité terrain. $A(Z)$ correspond à l'aire de la zone Z. En appliquant un seuil à ces deux critères, les fausses alarmes (éléments détectés par le système associés à aucun objet de la vérité terrain) et les oublis (éléments de la vérité terrain associés à aucun objet détecté par le système) sont extraits. Les mesures de précision, rappel et de moyenne harmonique sont alors utilisées.

Si un objet est repéré par plusieurs zones (dans [14] un visage est repéré par ses deux yeux), la *distance* entre la vérité terrain et les résultats du système est évaluée par un ensemble de quatre valeurs (3 rapports de distance et un angle). Chacune de ces composantes est évaluée séparément et les 4 mesures obtenues sont fusionnées par une combinaison linéaire.

Cette dernière étude préfigure les cas d'*associations multiples* traités dans plusieurs travaux [9, 11, 17]. La mesure décrite dans [17] s'appuie ainsi sur la définition de deux matrices associées aux taux de recouvrement décrits dans l'équation 1. Les mesures de précision et de rappel sont alors calculées à partir de ces matrices en assignant des coûts en fonction du cas observé : une (plusieurs) zone(s) de la vérité terrain est(sont) associé(es) à une(plusieurs) zone(s) détecté(es) par le système.

3.3.2. Définition des mesures d'évaluation adoptées

Il est nécessaire :

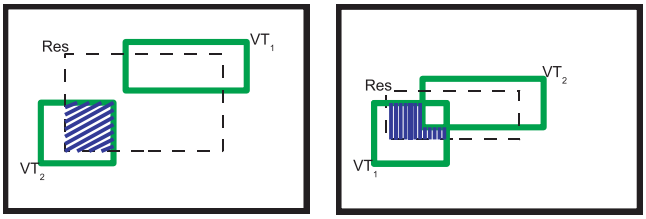
1. d'associer entre eux selon un seuil (en se basant sur l'équation 1), les objets de la vérité terrain et ceux détectés par le système, l'extraction des fausses alarmes et des oublis étant effectuée dans le même temps,
2. de proposer un vecteur de mesures de performances pour chaque objet de la vérité terrain qui n'a pas été « *oublié* » par le système, vecteur devant prendre en compte les cas de **fusion** (lorsque plusieurs zones de la vérité terrain sont associées à une même zone détectée par le système) ; et de **segmentation** (lorsque plusieurs zones détectées sont associées au même objet de la vérité terrain).

On notera par la suite $\{VT_i\}_i$ l'ensemble des zones de la vérité terrain dans une image et $\{Res_k\}_k$ l'ensemble des zones détectées par le système dans cette même image. On notera par ailleurs E_i^{Res} l'ensemble des éléments détectés par le système associés à l'élément VT_i. La mesure d'évaluation de la détection de chaque zone VT_i comporte alors quatre indices différents : deux d'entre eux correspondent aux taux de recouvrement mutuels de l'équation 1, un autre est lié aux positions respectives des zones associées et un dernier indice est lié à la fusion (tous ces indices sont moyennés en fonction du nombre d'éléments associés à VT_i, *i.e* : le nombre d'éléments de E_i^{Res}). La segmentation est prise en compte dans un des taux de recouvrement ($A(VT_i \cap Res_j)/A(VT_i)$ moyennés selon j) : plus le nombre d'éléments détectés entrant en compte (*i.e* : j) est important, plus cet indice diminue.

L'indice I_{fusion} est consacré à la fusion. Soit VT_I auquel est associée Res_k. Dans le cas de la fusion, Res_k est aussi associée avec un ensemble d'autres zones de la vérité terrain. Selon que les zones de la vérité terrain s'intersectent ou non (*cf.* figure 3), l'indice utilisé est plus ou moins restrictif :

$$I_{fusion}(VT_I, Res_k) = \frac{\sum_{i \neq I} A(Res_k \cap VT_i)}{A(Res_k \backslash (Res_k \cap VT_I))} \quad \textit{dans le premier cas ;}$$

$$= \frac{\sum_{i \neq I} A(Res_k \cap (VT_i \backslash VT_I))}{A(Res_k \backslash (Res_k \cap VT_I))} \quad \textit{sinon}$$



a) $VT_1 \cup VT_2 = \emptyset$ b) $VT_1 \cup VT_2 \neq \emptyset$

Figure 3. Les différents cas de figure de la fusion.

Spécificités liées au traitement de l'objet texte La vérité terrain des textes vidéos est constituée de mots, lignes et blocs stockés sous forme hiérarchique. La détection est alors évaluée à ces différents niveaux et le vecteur de performance finalement produit comporte 12 éléments (4 par niveaux). L'extraction des fausses alarmes et des oublis est par ailleurs adaptée pour tenir compte de cette hiérarchie.

3.4. Extraction des classes de comportements

L'application de la mesure d'évaluation adoptée produit un ensemble de vecteurs de performance qu'il convient de regrou-

per en classes de comportements homogènes. Parmi les comportements obtenus, l'enjeu sera alors d'extraire les comportements « insuffisants », pour lesquels une adaptation est nécessaire. Avant de détailler la méthode de clustering développée et celle d'extraction des comportements insuffisants, nous aborderons la question du traitement des comportements particuliers : les oublis et les fausses alarmes.

3.4.1. Traitement des oublis et des fausses alarmes

Les fausses alarmes et les oublis sont extraits à l'issue du seuillage concernant les taux de recouvrement entre les zones de la vérité terrain et les zones détectées par le système. Il n'est pas possible dans ces cas particuliers de calculer des vecteurs de performances et d'appliquer la suite de la méthodologie. En effet, cette méthodologie s'attache essentiellement à traiter les erreurs n'appartenant à aucune de ces deux catégories, erreurs généralement ignorées dans la littérature. Pour autant, une méthodologie similaire (non mise en oeuvre dans cet article) pourrait être appliquée aux oublis en mettant en oeuvre uniquement la phase de diagnostic de responsabilité.

Concernant les fausses alarmes, étant donné qu'il est difficile d'optimiser un système dans l'objectif « de ne pas détecter certaines zones », nous préférons construire des filtres qui seront appliqués *a posteriori*. L'objectif de ce filtrage est de supprimer les fausses alarmes tout en conservant les résultats corrects produits par le système. L'idée est alors de construire une signature de ces deux classes (appelées par la suite classe « fausses alarmes » et classe « résultats corrects ») et de supprimer un élément détecté si sa signature est plus proche de celle de la classe « fausses alarmes » que de celle de la classe « résultats corrects ».

Les caractéristiques utilisées pour créer ces signatures (vecteurs) sont les suivantes :

1. **Caractéristiques colorimétriques** : chaque zone est décomposée en RGB et HSL. Pour chaque canal, la moyenne des valeurs, tout comme les moments d'ordre 2 et 3 sont calculés (18 caractéristiques).
2. **Caractéristiques géométriques** : positions et tailles ainsi que les ratios $\frac{Y}{X}$ et $\frac{Largeur}{Hauteur}$ (6 caractéristiques).

Deux ensembles d'apprentissage de 2 000 éléments de chacune des deux classes sont constitués. Pour tenir compte de l'hétérogénéité de ces données (surtout pour la classe « fausses alarmes »), 4 sous-classes (ce chiffre est choisi expérimentalement : il correspond à la valeur permettant *in fine* de discriminer le plus facilement les deux classes) sont extraites de chacun de ces ensembles. Par ailleurs, une phase de sélection des caractéristiques par la méthode de Fisher est appliquée. Cette méthode produit un classement des caractéristiques en fonction de leur capacité à séparer les deux classes. En s'appuyant sur ce classement, plusieurs ensembles de caractéristiques sont testés (avec uniquement la première caractéristique, les deux premières, etc.). Pour chacun de ces ensembles, les 4 sous-classes « résultats corrects » et les 4 sous-classes « fausses alarmes » sont extraites et 8 signatures sont alors calculées (la signature correspond au centroïde de la classe). Ces différents ensembles de signatures (un ensemble de 8 signatures pour chaque ensemble de caractéristiques testé) sont alors évalués en termes de taux de suppression des deux classes, sachant que désormais, un élément détecté est supprimé si la distance euclidienne minimale entre sa signature et les huit signatures considérées est obtenue avec une signature d'une sous-classe de la classe « fausses alarmes ».

3.4.2. Choix d'une méthode de clustering et extraction des comportements insuffisants

La contrainte principale à prendre en compte concerne l'ignorance du nombre de classes à considérer. Une méthode simple, au paramétrage limité, a été développée dans ce sens. Celle-ci repose sur la « compétition » entre des algorithmes éprouvés : les *k*-moyennes et un algorithme *classique* de clustering hiérarchique dit de « *linkage* » [7]. Le principe est ainsi de choisir le meilleur résultat de clustering parmi un ensemble de résultats produits selon différents paramétrages de ces deux méthodes.

Dans un premier temps, le nombre de classes optimal (noté k_{opti}) est déterminé au sens du critère de Davies Bouldin (DB) [7] en s'appuyant sur une application itérative de l'algorithme des *k*-moyennes : $k_{opti} = \underset{k}{\text{argmax}}\{DB(k)\}$. Par la suite, un résultat optimal de l'algorithme des *k*-moyennes (noté $Kmoy_{opti}$) est produit pour cette valeur de *k*, en faisant varier la position des k_{opti} centroïdes initiaux.

Par ailleurs, un ensemble de représentations hiérarchiques des données (arbres) sont produites en appliquant l'algorithme de *linkage* selon différentes combinaisons des paramétrages suivants :

1. **Mode de normalisation des données** : normalisation $\sigma - \mu$ ou « min-max ».
2. **Mise en oeuvre (ou non) d'une réduction par ACP des données**,
3. **Distance utilisée pour mesurer l'écart entre les données** : L_1, L_2, L_3 et L_∞,
4. **Distance d'appartenance à un cluster** (*élément-cluster* ou *cluster-cluster*) : $d_{min}(\mathcal{C}_1,\mathcal{C}_2) = min_{(c_i\in\mathcal{C}_1, c_j\in\mathcal{C}_2)}(\|c_i - c_j\|_{L_2})$, $d_{max}(\mathcal{C}_1,\mathcal{C}_2) = max_{(c_i\in\mathcal{C}_1, c_j\in\mathcal{C}_2)}(\|c_i - c_j\|_{L_2})$ et $d_{moy}(\mathcal{C}_1,\mathcal{C}_2) = \|\overline{C}_1 - \overline{C}_2\|_{L_2}$

La meilleure de ces représentations (notée CHA_{opti}) est ensuite choisie selon la mesure de Cophenet (ou mesure de corrélation de Pearson). L'arbre obtenu est finalement coupé pour obtenir k_{opti} classes (le résultat est noté CHA^c_{opti}).

La dernière étape de la méthode consiste finalement à comparer les deux résultats de clustering obtenus ($Kmoy_{opti}$ et CHA^c_{opti}) selon l'indice de Davies-Bouldin.

Une fois les différentes classes de comportements extraites, leur « qualité » est mesurée par la norme de leur centroïde. Cette norme est alors comparée à un seuil et l'ensemble des classes dont la norme est inférieure à ce seuil sont considérées par la

suite comme les comportements « insuffisants » pour lesquels une adaptation est nécessaire. Le choix du seuil est alors laissé à l'initiative de l'utilisateur : plus ce seuil est élevé plus le nombre de classes à adapter sera important. La valeur du seuil est donc proportionnelle à l'effort d'adaptation envisagé.

4. Diagnostic de responsabilité

Une fois les comportements « insuffisants » extraits, l'enjeu est de déterminer pour chacun d'entre eux l'opérateur participant à la détection responsable de l'erreur à laquelle ils correspondent. La solution proposée s'appuie alors sur deux étapes distinctes :

1. Construction d'une nouvelle représentation des objets (zones extraites par le système de détection) contenus dans les classes correspondant aux comportements « insuffisants » basée sur un ensemble de caractéristiques visuelles (taille, texture, couleur, ...) prédéterminé,

2. Analyse du fonctionnement des opérateurs en fonction de ces caractéristiques, se concrétisant par le calcul d'un indice de responsabilité pour chaque caractéristique et chaque opérateur.

L'analyse du fonctionnement des opérateurs équivaut ici à quantifier la variation des résultats de ces derniers en fonction des caractéristiques. Deux méthodes différentes, toutes deux basées sur les deux étapes présentées ci-dessus, ont été expérimentées. La première repose sur la quantification de la sensibilité des opérateurs aux caractéristiques. La seconde s'appuie sur l'analyse des courbes d'évolution des résultats des différents opérateurs en fonction des caractéristiques. En référence au principe d'autonomie adopté, on soulignera ici l'usage, dans ces deux méthodes, des seules connaissances disponibles : les entrées des opérateurs, sous la forme de la représentation des objets contenus dans les classes de comportements ; ainsi que les sorties de tous les opérateurs.
Ces deux méthodes nécessitent de constituer un ensemble de caractéristiques visuelles auxquelles sont associées des bases d'objets dont les variations de chaque caractéristique sont contrôlées. Nous aborderons ainsi dans un premier temps la question du choix des caractéristiques et de la construction des bases leur étant dédiées.

4.1. Choix des caractéristiques et construction des bases dédiées

Le choix des caractéristiques est guidé par le mode de construction des bases qui leur sont associées. En effet, ces bases doivent contenir des objets dont une unique caractéristique varie et cette variation doit être maîtrisée. En conséquence, elles doivent être construites manuellement : si elles l'étaient de façon automatique en se basant sur des images « réelles » extraites d'un flux vidéo, des variations parasites les rendraient inexploitables. Par ailleurs, les caractéristiques utilisées (au nombre de 12) pour représenter le texte sont choisies en fonction de celles sur lesquelles reposent généralement les systèmes de la littérature.
Les méthodes d'estimation de ces caractéristiques ne seront pas présentées ici pour ne pas obscurcir le propos central de l'étude. En voici néanmoins la nature ainsi que les noms associés, utilisés par la suite :

1. **Couleur du texte et du fond** (*couleur 1, couleur 2*)

2. **Contraste** (*Contraste*)

3. **Complexité du texte *CT* et de son fond *CF*** : quantifie la densité de contours horizontaux et verticaux dans la zone du texte et, pour le fond, dans une couronne autour de cette zone. Cette couronne est par ailleurs utilisée pour le calcul de toute caractéristique relative au fond. (*CT, CFV, CFH*)

4. **Orientation des contours du fond** (OC_F)

5. **Position et dimensions de la zone de texte** (X_{pos}, Y_{pos}, L et H)

6. **Orientation du texte** : correspond à l'orientation du texte en termes de rotation autour de l'axe de la caméra. (O_T)

Si la construction de certaines bases est immédiate (par exemple celles dédiées à la position, au contraste, etc.), il convient de donner quelques précisions pour les bases les plus *problématiques* :
- **Complexité du texte** : utilisation de différentes polices en vue de simuler la variation de la complexité en termes de densité de contours,
- **Complexité du fond** : on simule la complexité horizontale et verticale en créant des images dont le fond est strié par des lignes (verticales ou horizontales respectivement). Lorsque la fréquence de ces lignes augmente, la complexité associée augmente dans le même temps.
- **Orientation des contours** : la base se compose d'images dont le fond est strié par des lignes dont l'angle est contrôlé (le nombre en est fixe).
Le nombre d'images contenues dans chacune de ces bases dépend de la caractéristique à laquelle celles-ci sont associées (de 18 éléments pour la base dédiée à la caractéristique *CFV*, à 192 éléments pour celle associée au contraste).

4.2. Une méthodologie d'analyse du fonctionnement basée sur la sensibilité

4.2.1. Principe et fonctionnement

Par la suite la définition suivante de la sensibilité est adoptée :

Définition 2. *Un opérateur est dit sensible à une caractéristique si l'application de celui-ci sur des objets pour lesquels cette caractéristique varie aboutit à une variation importante de ses résultats.*

La quantification des variations des résultats est complexe : il est difficile de définir un seuil à partir duquel un opérateur est

déclaré sensible à une caractéristique. Nous avons en conséquence défini une méthode originale de quantification basée sur les travaux exposés dans [5], s'inspirant du principe d'Helmholtz selon lequel « des structures remarquables dans une image peuvent être vues comme des exceptions à l'aléatoire ». L'objectif de ces travaux est de détecter les évènements saillants dans une image (limités à un ensemble de formes géométriques simples) en supposant qu'un tel évènement se caractérise par une faible probabilité d'observation, relativement à une distribution des pixels dans l'image de la forme d'un bruit uniforme. Cette approche est alors transposée à notre problématique en s'appuyant sur une nouvelle définition de la sensibilité :

Définition 3. *Un opérateur est dit* ***sensible*** *aux variations d'une caractéristique f si ses résultats varient significativement en fonction des valeurs prises par f. Les variations constatées sont considérées comme significatives si elles sont statistiquement équivalentes (ou supérieures) à celles obtenues par un opérateur stochastique de même nature.*

Le problème clé est alors de définir clairement, pour chaque opérateur composant la séquence de la détection, un opérateur stochastique équivalent. Une première méthodologie consiste à appliquer strictement le principe exposé dans [5]. Dans ce cas, la définition de l'opérateur stochastique ne prend pas en compte la nature des modifications apportées par l'opérateur considéré (détection d'un contour, augmentation du contraste, ...). Seule la nature des données produites par celui-ci est considérée et l'opérateur stochastique équivalent est alors défini selon une distribution uniforme. Tous les opérateurs travaillant dans des espaces identiques (niveaux de gris par exemple) se voient ainsi associer des opérateurs stochastiques identiques. Dans le cas d'un détecteur de contours produisant des images en niveaux de gris, l'opérateur stochastique équivalent consiste ainsi à assigner à chaque pixel une valeur en niveau de gris, sachant que chaque valeur est équiprobable.

4.2.2. Mise en application

Considérons un système composé de trois opérateurs, dont la sensibilité à une caractéristique C_k^{visu} doit être étudiée ($m = op_{sys}^0 \succ op_{sys}^1 \succ op_{sys}^2$, où l'indice *sys* signifie que l'opérateur considéré correspond à sa version originelle et non à son équivalent stochastique op_{stoch}). Le comportement de tout opérateur est évalué dans le contexte de la séquence à laquelle il appartient : les modifications apportées par l'opérateur le précédant doivent être prises en compte. En conséquence, si l'on admet qu'une base d'images $DB_{C_k^{visu}}$ dédiée à C_k^{visu} existe, l'étude de la sensibilité de l'opérateur op_{sys}^1 relativement à la caractéristique C_k^{visu} impose la mise en application des étapes suivantes :

1. Application de l'ensemble de la séquence « $op_{sys}^0 \succ op_{sys}^1$ » à chacune des images de $DB_{C_k^{visu}}$,

2. Application de la séquence stochastique « $op_{sys}^0 \succ op_{stoch}^1$ » autant de fois que $DB_{C_k^{visu}}$ contient d'images,

3. Calcul de la variance des résultats obtenus dans les deux cas,

4. Comparaison des variances obtenues selon un test d'hypothèse bilatéral.

Supposant que l'ensemble des résultats produits sont de type « image », le calcul de la variance repose sur la norme de Frobenius (équivalent matriciel à la norme euclidienne pour les vecteurs), adaptée au calcul de distances entre images. En ce qui concerne le test bilatéral de comparaison entre les deux variances calculées, une table de Fisher-Snédécor permet de rejeter ou d'accepter (auquel cas l'opérateur est déclaré sensible) l'hypothèse H_0 selon laquelle celles-ci sont comparables. Les résultats obtenus sur trois opérateurs (deux détecteurs de contours et un opérateur de réhaussement de contraste), relativement à trois caractéristiques (bruit, luminosité et contraste) ont abouti à la conclusion d'une unique sensibilité : celle de l'opérateur de réhaussement relativement à la luminosité. Or il est évident que les opérateurs de détection de contours dépendent du contraste de l'image. Ces premiers résultats se sont donc avérés insuffisants et nous ont amenés à envisager une modification, consistant à adopter une représentation plus précise du comportement des opérateurs pour la création de leur équivalent stochastique et ceci grâce à une estimation de la distribution de ce dernier. Plus précisément, l'idée consiste à calculer $P(X^+ = x | X^- = y)$, représentant la probabilité d'obtenir au pixel X, après application de l'opérateur considéré, la valeur x, sachant que ce pixel avait, avant application, la valeur y. Malheureusement, de telles estimations s'avèrent trop difficiles pour des opérateurs complexes. Bien qu'originale cette méthode a donc été finalement abandonnée au profit de la seconde méthode « analytique » basée sur l'analyse des courbes d'évolution des résultats des opérateurs.

4.3. Une méthodologie « *analytique* » d'analyse du fonctionnement

4.3.1. Principe et validation

La construction des courbes déjà mentionnées repose sur la comparaison entre les sorties des différents opérateurs et la vérité terrain de la sortie finale du système de détection (*cf.* figure 4).

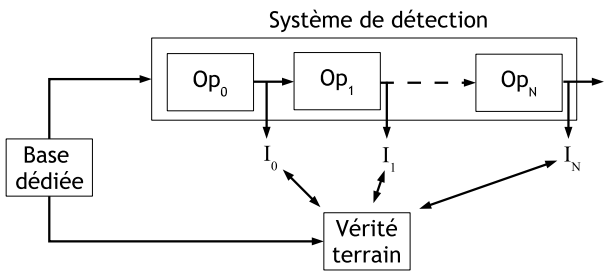


Figure 4. Les performances des opérateurs sont évaluées relativement à la sortie finale du système.

Comme nous l'avons déjà évoqué, une des contraintes d'application de la méthodologie est que les sorties de chaque opérateur soient du même type que la sortie finale. Cette contrainte permet ici de faciliter la comparaison de ces sorties intermédiaires avec la sortie finale attendue (la vérité terrain). Dans le cas de la détection par exemple, cette vérité terrain est une image binaire dans laquelle l'objet est délimité par une zone blanche et tous les résultats produits par les opérateurs sont donc de type « image ». Les différentes mesures obtenues (une nouvelle fois en utilisant la norme de Frobenius) permettent alors de construire les courbes qui serviront de base à l'établissement du diagnostic de responsabilité, en se basant sur le postulat suivant :

Postulat 2. *Chaque opérateur entrant dans la composition de la séquence du système de détection, tend à améliorer les résultats obtenus par l'opérateur le précédant dans la séquence.*

Une première validation de ce postulat relève de son utilisation dans le contexte de la planification des systèmes de traitement d'images ([6, 4]). Dans ces travaux, le choix des opérateurs constituant la séquence de traitement est ainsi guidé par la capacité des opérateurs à faire évoluer les données transitant dans la séquence vers le résultat final attendu.
Une expérimentation menée pour l'objet texte nous permet de donner une seconde validation de l'adoption de ce postulat. Nous avons ainsi vérifié que pour des textes correctement détectés, le comportement des modules validait le postulat. Comparant la sortie des 3 modules du système utilisé ([17]), appliqués sur 7 images dans lesquelles les textes sont correctement détectés et dont les vérités terrains sont construites (*cf.* figure 6), la progression séquentielle (au fur et à mesure de l'application des modules de la séquence) des résultats a ainsi été vérifiée (*cf.* figure 5). Ce « rapprochement » avec la vérité terrain est d'ailleurs illustré visuellement dans les images de la figure 6.

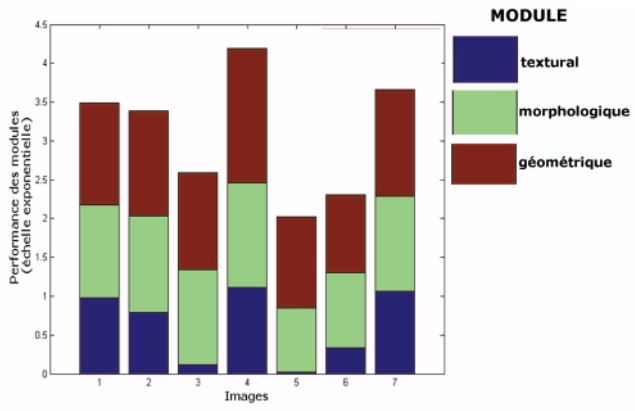


Figure 5. Illustration de la progression séquentielle des performances.

4.3.2. Calcul de l'indice de responsabilité

Le diagnostic repose sur l'identification de l'opérateur dont le comportement local va le plus à l'encontre du postulat 2. Pour chaque caractéristique, l'analyse des performances locales de chaque opérateur (sur la plage de variation de cette caractéristique mesurée pour le comportement considéré) permet de lui assigner un indice de responsabilité. Cet indice est établi en termes de dégradation locale des performances au regard de celles obtenues par l'opérateur le précédant dans la séquence.
La mesure des différences de performances entre deux opérateurs successifs sur la plage de variation d'une caractéristique est normalisée par la différence maximale de performances calculée sur l'ensemble des plages de même largeur pour rendre compte du comportement local des opérateurs. Nous rappelerons ici que la performance d'un opérateur (notée par la suite $0p_j$ pour un opérateur j) correspond à la distance de Frobenius

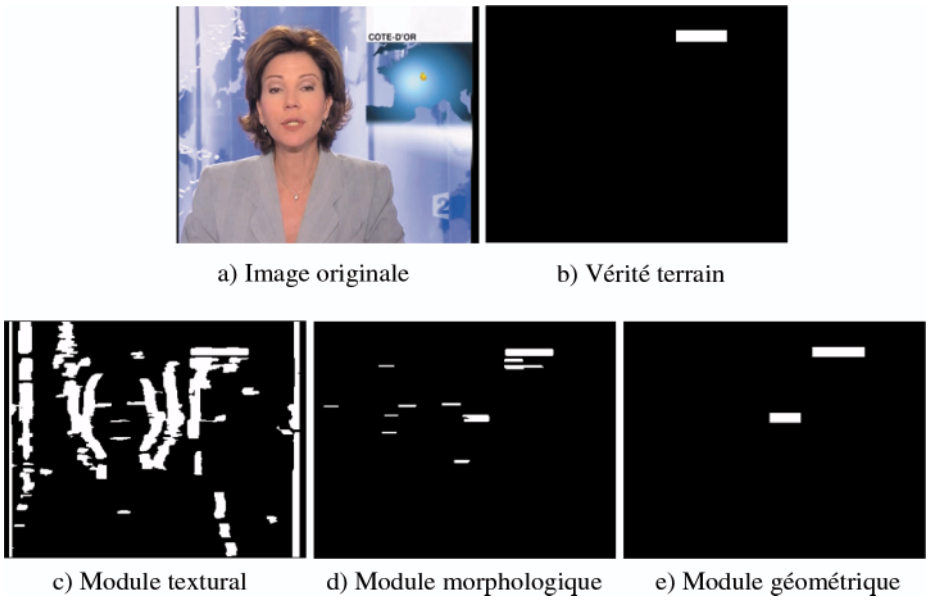


Figure 6. Les images prises en compte lors du calcul des performances.

entre l'image produite par cet opérateur et l'image représentant le résultat attendu en fin de chaîne, après application de la séquence des opérateurs en son ensemble. Plus précisément, si on note C_i la caractéristique, l'indice de responsabilité $\mathcal{I}(Op_j)$ attribué à l'opérateur Op_j est le suivant :

$$\mathcal{I}(0p_j) = argmin_{(x,y)} \left(\frac{\int_{C_{min}}^{C_{max}} (0p_j - 0p_{j-1})\, dC_i}{\int_x^y (0p_j - 0p_{j-1})\, dC_i} \right) \quad avec\ y - x = C_{max} - C_{min},\ \forall j \geq 1 \qquad (2)$$

où C_{min} et C_{max} délimitent la plage de variation de C_i sur le comportement.
Pour le premier opérateur Op_0, la formule est ajustée en quantifiant uniquement la singularité du comportement observé sur la plage $[C_{min}, C_{max}]$, selon l'équation suivante :

$$\mathcal{I}(Op_0) = argmin_{x,y} \left(\frac{\int_{C_{min}}^{C_{max}} Op_0\, dC_i}{\int_x^y Op_0\, dC_i} \right) \quad avec\ y - x = C_{max} - C_{min} \qquad (3)$$

L'indice de responsabilité varie sur l'intervalle $]-\infty, 1]$. Pour chaque caractéristique, l'opérateur obtenant l'indice minimal est considéré comme *responsable*. On obtient donc, pour chaque comportement, un opérateur *responsable* par caractéristique. Ces diagnostics sont ensuite fusionnés par un vote à la majorité : l'opérateur désigné comme responsable pour un maximum de caractéristiques est choisi. Lorsque ce vote aboutit à une ambiguïté, c'est l'ordre de mobilisation des opérateurs dans la séquence représentant le système de détection qui détermine le diagnostic final : le premier opérateur à être appliqué est choisi.
Bien entendu, si la responsabilité se limite ici à un unique opérateur, l'ensemble des indices de responsabilité pourrait être exploité par l'utilisateur sans aucun fusion ultérieure, ce dernier établissant alors lui même la rigueur de son optimisation en prenant en compte un ensemble plus ou moins large d'opérateurs responsables. Cette extension constitue ainsi une première alternative aux choix effectués dans notre méthodologie.

5. Récapitulatif

Les deux phases de la méthodologie (ainsi que celle relativement autonome de suppression des fausses alarmes) nécessitent l'utilisation de nombreux critères, distances, données, ..., rappelés ici brièvement préalablement à la partie sur les expérimentations :

1. **Analyse des comportements**
   (a) Evaluation des résultats sur le corpus d'adaptation : vérité terrain, mesures d'évaluation
   (b) Constitution des classes de comportement : méthode de clustering
   (c) Extraction des classes de comportements insuffisants : seuil relatif aux normes des centroïdes des classes produites lors du clustering.
2. **Diagnostic de responsabilité**
   (a) Choix de caractéristiques de représentation des objets contenus dans les classes (dans notre cas des zones extraites par le système de détection de texte)
   (b) Constitution de bases composées d'objets dont les variations de ces caractéristiques sont maîtrisées.
   (c) Analyse du fonctionnement des opérateurs en fonction de ces caractéristiques : mesure de la performance d'un opérateur, indice de responsabilité, méthode de fusion des diagnostics (par vote).
3. **Suppression des fausses alarmes**
   (a) Choix de caractéristiques pour distinguer les fausses alarmes des zones de textes
   (b) Construction d'une signature des textes et des fausses alarmes.

6. Expérimentations : application de la méthodologie à l'objet texte

6.1. Système de détection utilisé

Le système de détection de texte utilisé est détaillé dans [17]. Ce système est composé de trois modules (un module textural suivi d'un module morphologique et d'un module géométrique), eux même décomposés en opérateurs (3 pour le module textural, 6 pour le module morphologique et 3 pour le module géométrique). Cette décomposition du système autour de deux niveaux permet d'appréhender ce dernier selon deux granularités : celle des tâches (le premier niveau) et celle des opérateurs (le second niveau). Cette décomposition fait écho aux représentations hiérarchiques adoptées en planification des systèmes de traitement d'images (comme dans [3] par exemple).
Donnons ici quelques précisions sur le fonctionnement des modules dont l'enchaînement successif a pour objectif d'affiner progressivement, en mettant en oeuvre des caractéristiques différentes concernant le texte, la position des zones de texte détectées dans les images :

1. **Module textural** : il est supposé ici que les zones de texte présentent une forte réponse à un opérateur de Sobel horizontal, étant composées d'alternances horizontales de gradient (passages caractère/fond, fond/caractère). Une accumulation hori-

zontale et une binarisation permettent d'extraire les zones de texte candidates.
2. **Module morphologique** : l'application de plusieurs opérateurs morphologiques permet d'affiner la délimitation des zones de texte.
3. **Module géométrique** : les boîtes englobantes des composantes connexes sont créées et filtrées selon des contraintes géométriques. Finalement, les boîtes s'intersectant sont fusionnées selon des critères de recouvrement.

Ce système compte 18 paramètres différents (détaillés dans le tableau 1 qui précise dans le même temps la nature des opérateurs mis en jeu), auxquels sont associés des plages de variation quelques fois continues. La taille de l'espace des paramètres motive ainsi le ciblage de l'optimisation proposée dans notre méthodologie.

Tableau 1. Paramètres des différents opérateurs.

Module	Opérateur	Paramètres
Textural	Sobel Op_0	0
	Accumulation Op_1	1 *(S)*
	Binarisation Op_2	1 (α)
Morphologique	Fermeture Op_3	1 (N_F)
	Suppression des ponts Op_4	2 (N_{SP}, Th_1)
	Dilatation conditionnelle Op_5	3 (N_{DC}, Th_2, Th_3)
	Érosion conditionnelle Op_6	1 (N_{EC})
	Érosion horizontale Op_7	1 (N_{EH})
	Dilatation horizontale Op_8	1 (N_{DH})
Géométrique Op_{10}	Création des boîtes Op_9	2 (δ_X, δ_Y)
	Filtrage	2 (Th_4, Th_5)
	Fusion Op_{11}	3 (Th_6, Th_7, Th_8)

Voici quelques précisions concernant la nature de ces paramètres :

• S correspond à la longueur de la fenêtre d'accumulation horizontale,

• N_F, N_{SP}, N_{DC}, N_{EC}, N_{EH} et N_{DH} désignent le nombre d'itérations de l'opérateur morphologique auquel ils sont attachés,

• Th_1, Th_2, Th_3, Th_4, Th_5, Th_6, Th_7 et Th_8 correspondent à des seuils relatifs respectivement à :

1. la hauteur minimale d'une composante connexe,
2. les différences de position et de taille entre deux composantes en deçà desquelles un pixel placé entre ces deux composantes est dilaté lors de la dilatation conditionnelle,
3. le rapport $\frac{largeur}{hauteur}$ minimal d'une zone détectée,
4. les taux de recouvrement mutuels minimums entre deux zones détectées pour effectuer leur fusion.

• δ_X et δ_Y désignent la taille de l'agrandissement dans les directions x et y des zones détectées lors de leur création pour compenser la différence entre les effets de l'érosion horizontale et ceux de la dilatation horizontale.

6.2. Corpus d'adaptation

Les expérimentations sont menées sur un journal télévisé d'une durée de 37 minutes environ. On appellera par la suite « famille » l'ensemble des zones conservées dans la vérité terrain pour délimiter un texte dans une image (bloc, lignes et mots). La vérité terrain contient alors 21 514 familles différentes. Le module de détection produit 461 035 zones de textes. Ce nombre paraît ici extrêmement défavorable. Pour autant, le système développé dans [17] propose à la suite de la détection une phase de suivi qui permet de supprimer un grand nombre de fausses alarmes en s'appuyant sur des critères temporels (typiquement, une zone dont la détection est trop « instable » est considérée comme une fausse alarme). Par ailleurs, le document choisi comporte de nombreuses scènes de manifestations extrêmement texturées provoquant ainsi de nombreuses fausses alarmes.

6.3. Extraction des oublis et des fausses alarmes

Comme nous l'avons déjà expliqué, le choix du seuil utilisé pour isoler les oublis et les fausses alarmes relève de la volonté de l'utilisateur de pratiquer une adaptation plus ou moins importante. Le seuil de 0.2 (pour lequel les taux d'oublis et de fausses alarmes sont respectivement de 14 % et 95 %) est alors utilisé (on soulignera une nouvelle fois que ce taux très élevé de fausses alarmes n'est pas le reflet des résultats finaux obtenus par le système après application de la phase de suivi).

6.4. Traitement des fausses alarmes

La méthode de filtrage des fausses alarmes (*cf.* partie 3.4.1) nous amène à construire plusieurs ensembles de signatures (8 signatures par ensemble de caractéristiques considéré). Les filtres associés sont évalués sur un ensemble de test (différent de l'ensemble d'apprentissage) composé de 2 000 éléments de type fausse alarme et 2 000 éléments de type résultat correct, en termes de taux de suppression des fausses alarmes et des résultats corrects. Le meilleur résultat est obtenu en utilisant l'ensemble des 24 caractéristiques (ce qui tend donc à prouver que l'ensemble des caractéristiques considéré jusqu'à présent pourrait gagner à être étendu) : suppression de 67 % des fausses alarmes et de 22 % des résultats corrects.
Les images de la figure 7 illustrent trois résultats corrects classés parmi les fausses alarmes. Ces textes sont complexes au regard de leur orientation, des cas d'occlusion subis ou encore de leur résolution. Ils correspondent ainsi à des cas assez rares relativement à l'ensemble des résultats corrects qui concerne en

a) b) c)

Figure 7. Trois résultats corrects de l'algorithme classés parmi les fausses alarmes.

majorité des textes artificiels. La faible représentation de tels textes dans l'ensemble d'apprentissage des résultats corrects peut donc expliquer ces erreurs de classification. Malgré quelques erreurs, ces premiers résultats sont encourageants. Il sera à l'avenir envisagé de les améliorer en utilisant des caractéristiques supplémentaires (comme par exemple des moments statistiques liés à la caractérisation des textures) et en prenant en compte des ensembles d'apprentissage « mieux » construits en termes de représentativité des différentes catégories de résultats corrects.

Pour évaluer la robustesse des signatures calculées, une évaluation « quantitative » de cette méthode sera par ailleurs effectuée à l'avenir en l'appliquant sur les zones détectées par le système sur un document d'un type différent de celui considéré ici (du point de vue des textes qu'il contient).

6.5. Extraction des comportements

Une fois extraits les comportements « *extrêmes* » (oublis et fausses alarmes), des vecteurs de performances sont associés à l'ensemble des éléments de la vérité terrain qui n'ont pas été oubliés (soit 18 568 vecteurs (21 514-2 946 oublis)). L'algorithme de clustering produit alors 6 classes dont certaines sont illustrées dans les figures 8, 9 et 10.

Les classes sont obtenues automatiquement par l'algorithme de clustering. Chaque classe contient des zones sur lesquelles le système de détection de texte a produit des résultats similaires au regard des vecteurs de performances utilisés. Contrairement à une tâche de classification, la seule évaluation possible d'un tel résultat de clustering est manuelle et consiste à analyser la cohérence des classes obtenues en inspectant visuellement leur contenu. Cette tâche est difficile, d'autant que les vecteurs utilisés comme base du clustering sont de dimension trop élevée pour qu'une interprétation immédiate soit possible. Sur certaines classes pourtant, l'analyse se revèle assez simple. Nous présentons ainsi ici uniquement un descriptif de celles-ci.

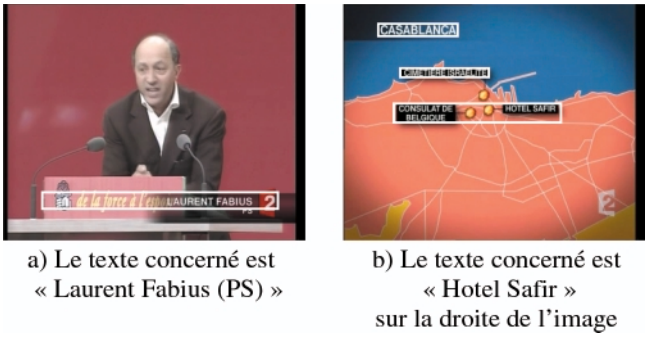
1. **La classe 1**, qui contient par ailleurs la très grande majorité des éléments, correspond aux cas pour lesquels la détection se déroule correctement. Les images extraites de cette classe (*cf.* figure 8) montrent alors que le module de détection produit des résultats satisfaisants sur des textes de scène (« sapeurs

a) b) c)

Figure 8. Images extraites de la classe 1.

a) b) c)

Figure 9. Images extraites de la classe 3



a) Le texte concerné est « Laurent Fabius (PS) »

b) Le texte concerné est « Hotel Safir » sur la droite de l'image

Figure 10. Images extraites de la classe 4.

pompiers » sur l'image a), sur des textes artificiels au niveau des lignes (image b) ou du bloc (image c).

2. **La classe 3** contient les cas pour lesquels la zone détectée par le système peut être simplement qualifiée de « trop large ». De tels résultats interviennent lorsqu'il existe à proximité des textes, des zones présentant des propriétés texturales similaires à celles supposées des zones de textes (zones de textures verticales).

3. **La classe 4** correspond à des cas de fusion.

Le tableau 2, qui montre les normes des centroïdes des classes obtenus confirme en partie l'analyse visuelle des classes puisque la classe 1 présente une norme élevée comparativement aux autres classes. Comme nous l'avons déjà expliqué, le seuil en deçà duquel une classe est considérée comme relevant d'un comportement insuffisant est choisi manuellement selon la qualité de l'adaptation envisagée par la suite. Pour illustrer la suite de la méthodologie sur un maximum de cas différents, on considérera ici les classes 2 à 6 comme des comportements insuffisants et ces classes deviennent ainsi les *cibles* de la phase de diagnostic.

Tableau 2. Analyse des classes de comportements.

Classe	*1*	*2*	*3*	*4*	*5*	*6*
$\|\|\tilde{C}\|\|$	2.60	1.82	1.58	1.24	1.61	0.74

6.6. Établissement du diagnostic de responsabilité

6.6.1. Analyse des courbes de résultats

Chacun des 12 opérateurs composant la séquence du système de détection est appliqué sur l'ensemble des images de chaque base pour produire les courbes de variation des performances (pour améliorer la précision de ces courbes, une interpolation selon des splines cubiques est effectuée). Etant donné que 12 caractéristiques sont considérées, on obtient donc un ensemble de 132 courbes. La figure 11 en montre quelques unes (les performances « négatives » observées dans l'image c correspondent à des fausses alarmes).
Les courbes de la figure 11 rendent compte des *a priori* sur le comportement des opérateurs, validant ainsi leur « allure » :

1. **Image a** : les performances de l'opérateur de Sobel sont proportionnelles au contraste du texte avec le fond (plus les contours du texte sont marqués, plus la réponse à cet opérateur est forte). Par ailleurs, on remarque aussi que l'application de l'opérateur d'accumulation permet de s'approcher de la vérité terrain dans la mesure où la réponse à l'opérateur de Sobel est étendue selon la fenêtre d'accumulation utilisée.

2. **Image b** : le fonctionnement de l'opérateur de dilation conditionnelle est invariant à la couleur du texte. Etant donné que la couleur n'est pas utilisée comme caractéristique prédominante dans le modèle du texte adopté, ce résultat paraît logique.

3. **Image c** : les résultats de l'opérateur de Sobel varient en fonction de l'orientation du fond. Un pic est obtenu pour 90 degrés, c'est-à-dire lorsque l'orientation du fond est proche de l'horizontal, auquel cas le fond présente une réponse moindre à l'opérateur de Sobel horizontal. Au contraire, une dégradation des performances est constatée autour de 0 et 180 degrés (lorsque l'orientation du fond est proche de la verticale), valeur pour lesquelles la réponse des pixels du fond à l'opérateur de Sobel horizontal est importante, provoquant ainsi des fausses alarmes. On constate par ailleurs une symétrie autour de 90 degrés.

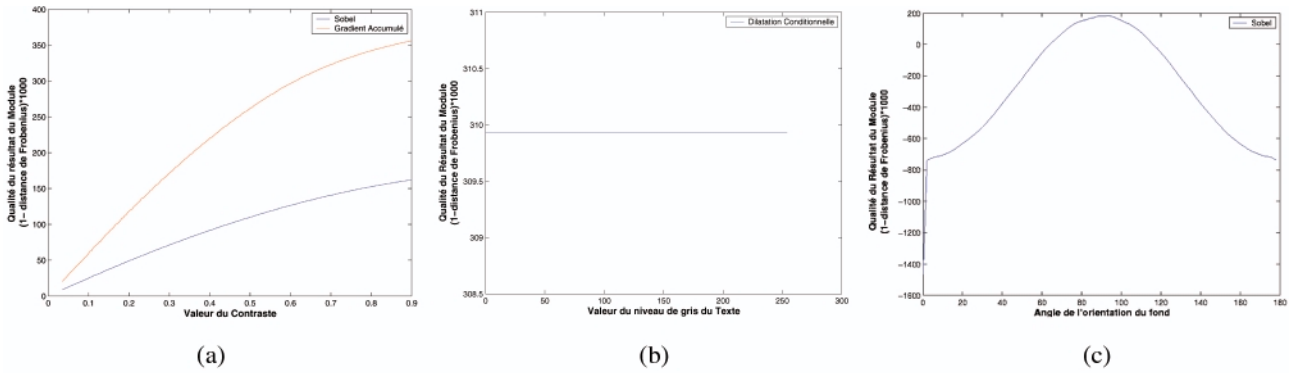


Figure 11. Courbes de variation de la qualité des résultats de certains opérateurs selon différentes caractéristiques.

6.6.2. Détermination des opérateurs responsables

Les courbes produites, associées au calcul, pour chaque classe de comportement, des plages de variations des différentes caractéristiques, permettent de calculer pour chaque opérateur et chaque caractéristique un indice de responsabilité. Une première fusion (sélection de l'opérateur avec l'indice minimal) permet de déterminer un opérateur responsable par caractéristique. La fusion finale selon un vote à la majorité permet ensuite de déterminer un unique opérateur responsable par comportement insuffisant. Ces différents diagnostics sont résumés dans le tableau 3.
L'analyse de la figure 3 donne alors lieu aux remarques suivantes :

• le diagnostic associé à certaines caractéristiques est le même quelle que soit la classe considérée : c'est le cas pour la position en Y de la zone de texte ainsi que pour l'angle de rotation du texte. Deux facteurs peuvent expliquer ce résultat : d'une part, la similarité des plages de variation des caractéristiques pour les différentes classes, et d'autre part le fait que ces plages soient relativement larges au regard de la plage de la base dédiée (ce qui réduit la variabilité des indices pouvant leur être associés).

• pour chaque classe, certains opérateurs ne sont jamais déclarés responsables (5 opérateurs pour la classe 2 ; 4 pour les classes 3, 4 et 5 ; 3 pour la classe 6), ce qui réduit ainsi dans l'absolu le nombre de paramètres à prendre en compte pour l'optimisation (8 paramètres en moins pour la classe 2 ; 7 pour les classes 3, 4 et 5 ; 6 pour la classe 6). Le fait que certains opérateurs ne soit jamais responsables pour une classe donnée signifie que les variations des caractéristiques des zones incluses dans cette classe appartiennent à des plages sur lesquelles les indices de responsabilité de ces opérateurs sont faibles, c'est à dire des plages sur lesquelles leur fonctionnement correspond au fonctionnement attendu. On remarquera par ailleurs que les opérateurs Op_4 et Op_{11} ne sont jamais responsables pour aucune caractéristique ni aucune classe ce qui tend à montrer que le paramétrage inital de ces opérateurs est suffisant pour s'adapter à l'ensemble des classes d'objets extraites sur le corpus considéré.

• l'assignation de la responsabilité à l'opérateur de Sobel (Op_0) pour la classe 4 soulève la question d'une remise en cause de la structure du système de détection. En effet, cet opérateur ne possède pas de paramètres. En conséquence, le fait qu'il soit déclaré responsable implique de changer d'opérateur. On pourrait penser par exemple à introduire un opérateur équivalent paramétré, comme le Sobel généralisé dont la taille du voisinage utilisé est variable [1]. On constate ici que la méthodologie permet de mettre en avant des limitations du système autres que celles uniquement liées au paramétrage.

Une validation expérimentale des diagnostics consiste à pratiquer la phase d'optimisation pour vérifier que la modification des paramètres des opérateurs diagnostiqués responsables permet une amélioration des résultats. Nous limitons ici cette validation aux classes 3 et 5. Notre but est d'illustrer l'optimisation sur deux opérateurs différents aux paramétrages limités (au contraire de Op_5 qui comporte 3 paramètres) pour faciliter la lecture et l'interprétation des résultats. Dans le cas de la classe

Tableau 3. Diagnostics de responsabilité par classe et par caractéristique.

		Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Diagnostic	*couleur1*	Op_2	Op_2	Op_0	Op_0	Op_0
	couleur2	Op_2	Op_2	Op_2	Op_2	Op_2
	contraste	Op_0	Op_8	Op_0	Op_9	Op_9
	CT	Op_{10}	Op_{10}	Op_7	Op_{10}	Op_{10}
	CF_{Horiz}	Op_3	Op_2	Op_2	Op_3	Op_3
	CF_{Verti}	Op_3	Op_3	Op_3	Op_{10}	Op_5
	OC_F	Op_0	Op_0	Op_0	Op_1	Op_{10}
	X_{pos}	Op_3	Op_3	Op_5	Op_5	Op_5
	Y_{pos}	Op_1	Op_1	Op_1	Op_1	Op_1
	L	Op_5	Op_8	Op_5	Op_5	Op_5
	H	Op_2	Op_2	Op_2	Op_7	Op_7
	O_T	Op_6	Op_6	Op_6	Op_6	Op_6
	VOTE	Op_2 ou Op_3	Op_2	Op_0 ou Op_2	Op_1, Op_5 ou Op_{10}	Op_5
	FINAL	**Op_2**	**Op_2**	**Op_0**	**Op_1**	**Op_5**

3, l'opérateur déclaré responsable est celui de binarisation (Op_2) auquel est attaché le paramètre α, réglant un seuil. Pour ce qui est de la classe 5, c'est l'opérateur d'accumulation (Op_1) qui est désigné responsable. Cet opérateur comporte lui aussi un unique paramètre, *S*, qui désigne la taille de la fenêtre d'accumulation. Les valeurs initiales de ces deux paramètres sont respectivement 0.87 et 13.

Les figures 12 et 13 montrent alors les résultats obtenus sur des images issues respectivement de la classe 3 et de la classe 5 pour différentes valeurs des paramètres concernés, ainsi que les résultats obtenus après filtrage des fausses alarmes.

L'analyse des résultats produits pour les deux images issues de la classe 3 (figure 12) montre que la modification du paramètre lié à la binarisation permet une amélioration des performances : en choisissant pour α une valeur proche de 0.96/0.97 on observe une réduction de l'erreur obtenue pour le paramétrage initial, à savoir une zone détectée trop grande. Dans le cas de la première image la zone détectée correspondant à la seconde ligne de texte incluait, avant modification (image (a)), une partie du chapeau de l'homme présent, chapeau dont la texture peut être « confondue » par le système avec une texture de texte. On constate que la modification du paramètre α permet de scinder cette grande zone en deux, une des deux correspondant au texte, et l'autre au chapeau (image (b)). L'application du filtre (image (c)) concernant les fausses alarmes permet finalement d'éliminer cette zone. Dans le cas de la seconde image (images (d) à (e)), la même séparation de la zone en deux zones est effectuée. Malheureusement la phase de suppression des fausses alarmes ne parvient pas à supprimer la zone erronée dont la texture est vraisemblablement trop similaire à celle du texte.

Concernant la classe 5 (figure 13), on observe une amélioration en fixant la taille de la fenêtre d'accumulation S à 28. Cette valeur permet de détecter le mot « VOUS » situé à droite sur la banderole, tout en améliorant la précision de la détection des autres mots. Concernant le filtrage des fausses alarmes, on remarque qu'il permet une suppression efficace de ces erreurs sans pour autant supprimer les résultats corrects correspondant pourtant à des textes complexes dans le cas de cette image de la classe 5.

Figure 12. Optimisation sur la classe 3 : les résultats (a) et (d) sont obtenus avec le paramétrage initial de α ; les résultats (b) et (e) sont produits pour un paramétrage différent ; les images (c) et (f) montrent les résultats du filtrage des fausses alarmes.

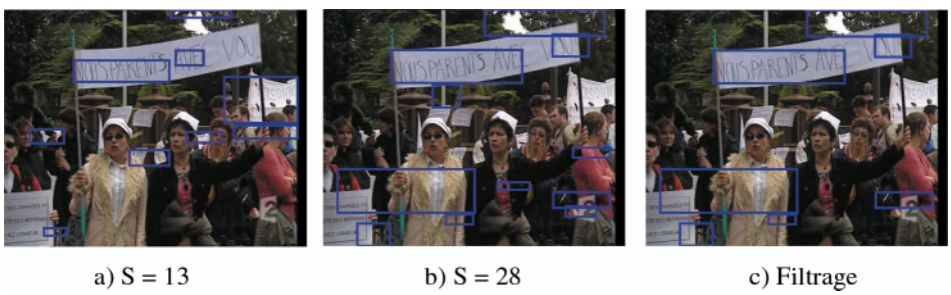


Figure 13. Optimisation sur la classe 5 : le résultat (a) est obtenu avec le paramétrage initial de S, le résultat (b) est produit pour un paramétrage différent ; l'image (c) correspond au résultat du filtrage des fausses alarmes.

Ces résultats montrent sur ces quelques exemples que la modification des paramètres des opérateurs déclarés responsables permet une amélioration des performances. Ils montrent par ailleurs la relative efficacité du filtre concernant les fausses alarmes. Bien entendu, cette évaluation n'est pas suffisamment quantitative pour valider entièrement la méthodologie et, comme il sera souligné en conclusion, il sera envisagé à l'avenir de mettre en oeuvre un protocole d'évaluation plus précis.

7. Conclusion

Nous avons présenté dans cet article une méthodologie originale de préparation à l'optimisation des systèmes de détection d'objets. Cette méthodologie est basée sur deux analyses innovantes : d'une part la détermination automatique des différents comportements d'un système, et d'autre part une phase de diagnostic de responsabilité, s'appuyant sur une analyse du fonctionnement des opérateurs composant le système.

L'apport majeur de la méthode réside dans l'application de la contrainte d'autonomie nous ayant amené à définir les principes d'une méthode de compréhension des systèmes par la seule analyse de leurs entrées/sorties. L'accent porté sur la notion de responsabilité des opérateurs constitue par ailleurs un autre contribution importante. On notera par ailleurs les nécessaires développements satellites effectués, comme la définition de mesures d'évaluation ou encore la construction d'une méthode de clustering adaptée à nos usages.

Les résultats obtenus, tant au niveau du diagnostic de responsabilité que de la suppression des fausses alarmes sont encourageants. À l'avenir, il sera intéressant de compléter l'évaluation de ces diagnostics en essayant notamment de quantifier leur apport relativement à une optimisation « standard » prenant en compte l'ensemble des paramètres.

La construction automatique des bases utilisées lors de l'analyse du fonctionnement des opérateurs, en utilisant des images du flux pour s'absoudre de la contrainte concernant le caractère synthétisable des caractéristiques, apparaît comme une autre perspective intéressante. Cette modification nous obligerait alors à prendre en compte les variations parasites précitées, variations dont on pourrait tenir compte lors de la fusion des diagnostics selon les différentes caractéristiques, en attribuant à chacun d'entre eux un degré de fiabilité relatif à la « qualité » de la base lui étant associée.

À plus long terme, un autre enjeu réside dans la capitalisation des connaissances acquises sur le mode de fonctionnement des systèmes, connaissances pouvant prendre la forme de plages de fonctionnement optimal en fonction des caractéristiques visuelles. Enfin, si nous avons adopté jusqu'ici une représentation séquentielle des systèmes (séquence de modules/d'opérateurs), il sera intéressant de réfléchir par la suite à la prise en compte de systèmes ne fonctionnant pas nécessairement sur ce mode (typiquement en parallèle).

Références

[1] L. ASFAR, Recherche des contours dans une image landsat. In *RFIA*, pages 237-248, Nancy, 1981.

[2] D. CHEN, J-M. ODOBEZ and H. BOURLARD, Text detection and recognition in images and video frames. *Pattern Recognition*, (37): 595-608, 2004.

[3] R. CLOUARD, A. ELMOATAZ, C. PORQUET and M. REVENU, Borg: a knowledge-based system for automatic generation of image processing programs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2): 128-144, 1999.

[4] P. DALLE and P. DEJEAN, Planification en traitement d'image : approche basée sur les données. In *Congrès RFIA*, pages 75-84, Clermont-Ferrand, France, 1998.

[5] A. DESOLNEUX, L. MOISAN and J-M. MOREL, Edge detection by helmholtz principle. *Journal of Mathematical Imaging and Vision*, 14(3): 271-284, 2001.

[6] B.A. DRAPER, J. BINS and K. BAEK, Adore: Adaptative object recognition. *VIDERE*, 1(4): 86-99, 2000.

[7] R.O. DUDA, P.E. HART and D.G. STORK, *Pattern Classification, second edition*. New York: Wiley And Sons, 2001. 654 p.

[8] V. FICET-CAUCHARD, C. PORQUET and M. REVENU, An interactive case-based reasoning system for the development of image processing applications. In *European Workshop on Case Base Reasoning (EWCBR)*, pages 437-447, Dublin, Irlande, 1998.

[9] X-S. HUA, L. WENYIN, and H-J. ZHANG, An automatic performance evaluation protocol for video text detection algorithms. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4): 498-507, 2004.

[10] R. LIENHART and A. WERNIKE, Localizing and segmenting text in images, videos and web pages. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(4): 256-268, 2002.

[11] V. Y. MARIANO, J. MIN, J-H. PARK, R. KASTURI, D. MIHALCIK, H. LI and D. DOERMANN, Performance evaluation of object detection algorithms. In *International Conference on Pattern Recognition*, pages 965-969, Québec, Canada, 2002.

[12] S. MOISAN, R. VINCENT, J. VAN DEN ELST and F. VAN HARMELEN, Towards an intelligent failure handling mechanism in program supervision. In *1st International Workshop on Knowledge Based systems for the (re)Use of Program Libraries*, pages 109-118, Sophia Antipolis, France, Nov 1995.

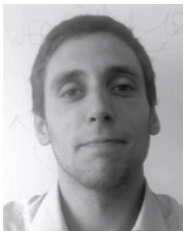
[13] V. PARAMESWARAN, P. BURLINA and R. CHELLAPPA, Performance analysis and learning approaches for vehicle detection and counting in aerial images. In *ICASSP*, pages 2753-2756, Munich, Allemagne, 1997.

[14] V. POPOVICI, Y. RODRIGUEZ, J.-P. THIRAN and S. MARCEL, On performance evaluation of face detection and localization algorithms. In *17th International Conference on Pattern Recognition*, volume 1, pages 313-317, Cambridge, Angleterre, 2004.

[15] C. SHEKHAR, S. MOISAN, R. VINCENT, P. BURLINA, and R. CHELLAPPA, Knowledge-based control of vision systems. *Image and Vision Computing*, 17: 667-683, 1998.

[16] S. TREUILLET, D. DRIOUCHI and P. RIBEREAU, Ajustement des paramètres d'une chaîne de traitements d'images par un plan d'expérience factoriel fractionnaire 2^{k-p}. *Traitement du Signal*, 21(2): 141-156, 2004.

[17] C. WOLF, *Détection de textes dans les images issues d'un flux vidéo pour l'indexation sémantique*. PhD thesis, Lyon : INSA de Lyon, 2003. 205 p.



Rémi **Landais**

Rémi Landais a obtenu son doctorat de l'INSA de Lyon spécialité informatique en 2006 sur la thématique du contrôle des systèmes de vision (convention CIFRE avec l'Institut National de l'Audiovisuel). En post-doctorat au sein du département TSI à l'ENST Paris depuis lors, sa recherche porte désormais sur l'analyse de la vidéo en se focalisant sur les indices relatifs aux visages (détection, suivi, reconnaissance, localisation audiovisuelle du locuteur, ...). Il participe à différents projets, tel que le projet Infom@gic du pôle de compétitivité Cap Digital de la région Ile-de-France et le réseau d'Excellence européen K-SPACE.

Laurent **Vinet**

Laurent Vinet a obtenu son doctorat en informatique et analyse d'image à l'Université Paris 9/INRIA en 1991. Il a été chercheur à Elf-Aquitaine pendant deux ans puis pour la société Eclimed pendant quatre ans où il a développé des systèmes d'analyse d'images médicales. Il est depuis 1996 chercheur dans le domaine de l'analyse des contenus à l'Institut National de l'Audiovisuel (INA) et ses travaux portent sur la description des documents audiovisuels, l'indexation vidéo et le design orienté objet. Il a été impliqué dans plusieurs projets français et européens (K-Space, ARGOS, ECHO, EuroDelphes, anim2000).

Jean-Michel **Jolion**

Jean-Michel Jolion est diplômé de l'INSA de Lyon (ingénieur en 1984 et doctorat en 1987, spécialité informatique). Séjour à l'Université du Maryland (Washington, USA) en 1987-1988. Maître de Conférences à l'Université Lyon I de 1998 à 1994 et Professeur à l'INSA de lyon de 1994 à 2007. Il est depuis délégué général de l'Université de Lyon. Il effectue sa recherche au laboratoire LIRIS au sein de l'équipe M2Disco sur les thèmes de la reconnaissance de formes statistiques appliquées aux structures discrètes.