Institut Formation Doctorale

THESE DE DOCTORAT DE
L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité

INFORMATIQUE
(EDITE de Paris)

Présentée par

M. Nicolas BURRUS

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Apprentissage *a contrario* et architecture efficace pour la détection d'évènements visuels significatifs

Préparée au
Laboratoire Électronique et Informatique (UEI)
de l'École Nationale Supérieure de Techniques Avancées (ENSTA)
32 Boulevard Victor, 75739 Paris Cedex 15

soutenue le 11 décembre 2008

devant le jury composé de :

Dr. Thierry BERNARD	Directeur de thèse
Dr. Jacques BLANC-TALON	Examinateur
Dr. Bertrand COLLIN	Examinateur
Pr. Matthieu CORD	Examinateur
Pr. Jean-Michel JOLION	Directeur de thèse
Pr. Alain MERIGOT	Rapporteur
Pr. Lionel MOISAN	Rapporteur

Résumé

Pour assurer la robustesse d'un algorithme de détection, il est nécessaire de maîtriser son point de fonctionnement, et en particulier son taux de fausses alarmes. Cette tâche est particulièrement difficile en vision artificielle à cause de la grande variabilité des images naturelles, qui amène généralement à introduire des paramètres choisis a priori qui limitent la portée et la validité des algorithmes. Récemment, l'approche statistique *a contrario* a montré sa capacité à détecter des structures visuelles sans autre paramètre libre que le nombre moyen de fausses alarmes tolérées, en recherchant des entités dont certaines propriétés sont statistiquement trop improbables pour être le fruit du hasard.

Les applications existantes reposent toutefois sur un cadre purement analytique qui requiert un travail important de modélisation, rend difficile l'utilisation de caractéristiques multiples et limite l'utilisation d'heuristiques de recherche dirigées par les données. Nous proposons dans cette thèse d'assouplir ces restrictions en ayant recours à de l'apprentissage pour les quantités non calculables analytiquement. Nous illustrons l'intérêt de la démarche à travers trois applications : la détection de segments, la segmentation en régions homogènes et la détection d'objets à partir d'une base de photos. Pour les deux premières applications, nous montrons que des seuils de détection robustes peuvent être appris à partir d'images de bruit blanc. Pour la dernière, nous montrons que quelques exemples d'images naturelles ne contenant pas d'objets de la base suffisent pour obtenir un algorithme de détection fiable.

Enfin, nous remarquons que la monotonicité du raisonnement *a contrario* permet d'intégrer incrémentalement des informations partielles. Cette propriété nous conduit à proposer une architecture "anytime" pour la détection d'objets, c'est-à-dire capable de fournir des détections progressivement au cours de son exécution, en commençant par les objets les plus saillants. L'algorithme peut donc être stoppé à tout moment pour satisfaire à des contraintes temporelles.

Mots-clés : Vision par ordinateur, méthodes *a contrario*, apprentissage statistique, segmentation, reconnaissance d'objets, anytime

Abstract

To ensure the robustness of a detection algorithm, it is important to get a close control of the false alarms it may produce. Because of the great variability of natural images, this task is very difficult in computer vision, and most methods have to rely on *a priori* chosen parameters. This limits the validity and applicability of the resulting algorithms. Recently, by searching for structures for which some properties are very unlikely to be due to chance, the *a contrario* statistical approach has proved successful to provide parameterless detection algorithms with a bounded expected number of false alarms.

However, existing applications rely on a purely analytical framework that requires a big modeling effort, makes it difficult to use heterogeneous features and limits the use of data-driven search heuristics. In this thesis, we propose to overcome these restrictions by using statistical learning for quantities that cannot be computed analytically. The interest of this approach is demonstrated through three applications : segment detection, segmentation into homogeneous regions, and object matching from a database of pictures. For the two first ones, we show that robust decision thresholds can be learned from white noise images. For the last one, we show that only a few examples of natural images that do not contain the database objects are sufficient to learn accurate decision thresholds.

Finally, we notice that the monotonicity of *a contrario* reasoning enables an incremental integration of partial data. This property leads us to propose an architecture for object detection which has an "anytime" behavior : it provides results all along its execution, the most salient first, and thus can be constrained to run in limited time.

Mots-clés : Computer vision, *a contrario* reasoning, statistical learning, segmentation, object detection, anytime

Remerciements

Si un seul nom figure sur ce manuscrit, cette thèse est le résultat du soutien et de la participation de nombreuses personnes que je tiens à remercier ici.

En premier lieu, merci à Thierry Bernard, mon co-directeur de thèse, qui m'a encadré avec enthousiasme pendant ces trois années. Ses conseils scientifiques et rédactionnels m'ont beaucoup appris et ont joué un rôle essentiel tout au long de ce travail. Je le remercie également pour la grande liberté qu'il m'a laissé dans mes recherches, en privilégiant toujours mon propre intérêt. Enfin, ses qualités humaines ont toujours permis un dialogue direct, constructif, et dans la bonne humeur, ce qui est très utile dans la longue expérience que constitue une thèse. J'ai également eu la chance d'être co-dirigé par Jean-Michel Jolion, qui, malgré un agenda extrêmement chargé, a toujours réussi à me consacrer du temps et à tenir des délais souvent très (trop) courts. Son expertise, sa grande expérience, et ses remarques toujours très pertinentes m'ont été très précieuses.

Merci également aux membres du jury qui m'ont fait l'honneur de s'extraire de leurs nombreuses activités pour juger ce travail de thèse. En particulier, merci aux rapporteurs Alain Mérigot et Lionel Moisan qui se sont beaucoup investis et ont fourni des rapports très détaillés, avec de nombreuses remarques utiles et constructives. Merci à Matthieu Cord pour avoir accepté de présider ce jury, et merci à Bertrand Collin, qui a participé à l'élaboration du sujet initial.

Une thèse, c'est aussi un ensemble de moyens matériels. À travers Jacques Blanc-Talon, je remercie la DGA qui nous a fait confiance en finançant cette thèse. Je remercie également l'ENSTA, et en particulier l'UEI, pour m'avoir accueilli dans d'aussi bonnes conditions, à la fois matérielles et humaines. Merci donc à Alain Sibille, son directeur, ainsi qu'à l'ensemble des techniciens, agents administratifs, ingénieurs et enseignants-chercheurs. Un merci particulier à Antoine Manzanera, avec qui j'ai été très heureux de pouvoir échanger (et surtout recevoir) régulièrement, sur le plan scientifique, mais également humain.

Je tiens aussi à remercier les autres doctorants et stagiaires de l'UEI que j'ai pu côtoyer pendant cette thèse. Leur contact et leur soutien ont rendu le quotidien très agréable. Je remercie tout spécialement les membres du "bureau rétine", où l'ambiance a toujours été très joyeuse, amicale et stimulante.

Je suis particulièrement heureux d'avoir l'occasion ici de remercier ceux qui m'ont fait découvrir le monde de la recherche. Merci donc aux membres du LRDE avec qui j'ai eu la chance de travailler pendant deux années qui ont profondément influencé la suite, pour finalement conduire à soutenir cette thèse. Un merci tout particulier à Akim Demaille et à Thierry Géraud, deux personnalités hors normes auprès de qui j'ai énormément appris et qui se sont toujours investis sans compter pour leurs étudiants.

Un mot également pour les amis qui ne sont pas inclus dans les remerciements précédents, qui ont su être présents dans les bons et les moins bons moments, mais qui ont également su être

patients pendant les périodes récurrentes où les notions de soirées et de week-end devenaient très abstraites...

Enfin, mes derniers remerciements vont à mes parents, ma soeur et à Éric. Ils ont été mes plus fidèles supporters, ils ont toujours été présents pour me donner de l'énergie, de l'optimisme ou tout simplement pour me soutenir. Leur confiance, leur dévouement et leurs encouragements tout au long de ma vie ont été un moteur indispensable. Je leur dédie cette thèse, car elle n'existerait pas sans eux. Merci.

Table des matières

Introduction

En vision par ordinateur, la détection consiste à rechercher des structures particulières dans une image, par exemple des visages, des véhicules, ou bien des éléments plus génériques comme les contours des objets de la scène. Elle peut être utile par elle-même pour de nombreuses applications, par exemple en robotique pour interagir avec des objets connus ou se localiser, en imagerie médicale pour repérer des tumeurs ou des microcalcifications, en vidéo-surveillance pour détecter des personnes ou des véhicules, en photographie pour focaliser automatiquement sur un visage, etc.

Elle joue également un rôle essentiel pour atteindre l'objectif à long terme de la vision par ordinateur, à savoir d'être capable de donner un sens à une image en construisant une interprétation de son contenu, de façon analogue à un humain. Il est très difficile d'aborder ce problème en partant directement des pixels de l'image. La recherche en vision a en effet montré la nécessité d'étapes intermédiaires, capables de fournir progressivement des briques de plus en plus haut niveau, sur lesquelles on peut s'appuyer pour analyser l'image de façon globale. La détection joue alors un rôle de proposition, elle fournit des indices qui peuvent ensuite être combinés, confrontés les uns aux autres, et raffinés pour trouver la meilleure interprétation du contenu de l'image. Parmi les briques classiques dans la littérature, on retrouve la détection de contours, d'alignements, de régions homogènes, mais également des éléments plus spécifiques et de plus haut niveau comme du texte, des visages ou des objets particuliers.

Dans un contexte de vision, un algorithme de détection comporte généralement deux composantes. La première définit comment les structures vont être recherchées dans l'image, quels vont être les "candidats" analysés. Il pourra par exemple s'agir de petites fenêtres dans l'image, ou bien d'ensembles de niveau pour l'approche morphologique. La seconde composante est un problème de décision, où, pour chaque candidat parcouru, il s'agit de décider s'il correspond à une structure à détecter ou non. C'est cet aspect sur lequel nous nous focalisons principalement dans cette thèse.

De nombreuses difficultés rendent cette tâche de décision complexe. Une image est en effet une capture en deux dimensions d'une scène tridimensionnelle. Une partie de l'information est donc perdue lors de la projection sur le capteur. Ce dernier n'est de plus jamais idéal, et l'image restituée contient toujours du bruit lié à l'optique utilisée et à la quantification numérique du signal lumineux. Au-delà de ces dégradations inhérentes au processus de capture, l'apparence des éléments d'une scène dans l'image peut fortement varier en fonction des conditions d'illumi-

nation ou du point de vue. Enfin, les éléments ont généralement une variabilité intrinsèque. Par exemple, l'apparence des visages changent d'une personne à l'autre, l'apparence des contours des objets varient en fonction de leur texture, etc.

Toutes ces variations empêchent le plus souvent de prendre des décisions avec certitude, et ont conduit de nombreux travaux à proposer une approche probabiliste de la vision. L'image est alors vue comme une réalisation d'un modèle de la scène, et la vision artificielle devient un problème d'inférence : il s'agit de retrouver le modèle de la scène à partir de l'image observée. Dans ce cadre, la décision pour un algorithme de détection consiste à déterminer, pour chaque candidat analysé dans l'image, s'il est associé à un modèle des éléments à détecter ou non.

Le choix de modèles appropriés reste cependant difficile, et la variabilité des apparences des éléments à détecter dans une image limite souvent la validité des modèles proposés à des environnements relativement restreints.

Ce constat a motivé l'approche statistique *a contrario* [DMM08], qui contourne cette difficulté en ne cherchant plus à modéliser explicitement l'apparence des éléments à détecter. Cette approche s'appuie sur le principe perceptuel de Helmholtz selon lequel plus un évènement a une probabilité faible d'apparaître par hasard, plus il est significatif, et plus nous le percevons. L'idée est alors d'identifier des mesures sur les candidats telles que plus leurs valeurs sont improbables par hasard, plus il y a de chances qu'un élément de la scène à détecter soit présent. Les éléments sont alors détectés *a contrario* en recherchant les candidats dont les mesures sont trop improbables pour être le résultat du hasard. Il est alors possible d'obtenir des algorithmes avec une portée relativement universelle, puisqu'ils se basent uniquement sur un modèle du hasard, qui peut généralement être défini de façon très générique.

Cette approche a été appliquée avec succès pour un certain nombre d'applications. Toutefois, la méthodologie utilisée dans les travaux existants, essentiellement analytique, induit un certain nombre de limitations et requiert un travail de modélisation mathématique conséquent. En particulier, nous montrons que cette approche souffre de deux limitations principales. Premièrement, elle rend difficile l'utilisation de mesures multiples et hétérogènes, qui sont pourtant très utiles pour augmenter le pouvoir discriminant du critère de décision et augmenter les taux de détection. Deuxièmement, elle limite l'utilisation d'heuristiques de parcours des candidats dirigées par les données, souvent nécessaires pour faire face à la masse d'information potentiellement analysable dans l'image.

Nous proposons dans cette thèse d'assouplir les conditions d'utilisation du cadre statistique *a contrario* en mixant calculs analytiques et apprentissage. Le principe est d'apprendre les quantités non estimables analytiquement à partir de réalisations du modèle de hasard, souvent faciles à générer. Nous illustrons à travers trois applications que les limitations du cadre purement analytique que nous avons mentionnées peuvent alors être levées, tout en préservant la portée et la fiabilité des algorithmes de détection obtenus.

Nous abordons également la question de l'implantation rapide d'algorithmes de détection *a contrario*. En remarquant que l'approche *a contrario* est naturellement capable d'intégrer progressivement des informations partielles, nous expérimentons une architecture permettant une

implantation “anytime” d’un algorithme de détection *a contrario*, c’est-à-dire capable de fournir des détections progressivement au cours de son exécution, en commençant par les éléments les plus saillants. L’algorithme cherche alors à maximiser le nombre et la qualité des détections en fonction du temps imparti, et peut être interrompu à tout moment pour satisfaire des contraintes de temps réel ou limité.

Plan de la thèse

Le chapitre 1 présente l’approche *a contrario* de [DMM08] et propose un tour d’horizon des applications existantes. Nous faisons ensuite ressortir certaines limitations de la méthodologie analytique utilisée dans ces travaux, notamment la difficulté à combiner plusieurs mesures discriminantes et les restrictions sur l’utilisation d’heuristiques de parcours des candidats dirigées par les données.

Le chapitre 2 étudie comment ces limitations peuvent être assouplies pour des primitives de bas niveau. Le modèle de hasard classique dans ce cas considère que les pixels sont indépendants et identiquement distribués, aussi nous proposons d’apprendre les quantités non calculables analytiquement par un apprentissage *a contrario* à partir d’images de bruit blanc. Deux applications illustrent la démarche. La première cherche à détecter les structures rectilignes de la scène. Elle s’appuie sur l’apprentissage pour estimer la distribution d’une mesure discriminante en présence d’une heuristique dirigée par les données. La seconde aborde la segmentation d’une image en régions homogènes. Nous montrons alors comment un apprentissage *a contrario* permet de combiner différentes mesures de distance entre régions au sein d’heuristiques de parcours complexes pour filtrer efficacement les fausses alarmes produites par les algorithmes de segmentation existants.

Le chapitre 3 expérimente l’apprentissage *a contrario* pour des applications de plus haut niveau à travers une application de détection d’instances d’objets connus et stockés dans une base de données. Le modèle de hasard choisi considère alors qu’aucun objet de la base n’est présent dans l’image. Nous montrons qu’un apprentissage à partir d’images naturelles ne contenant pas d’objets de la base, faciles à collecter, permet d’assouplir l’utilisation du cadre *a contrario*, en permettant de combiner des mesures discriminantes hétérogènes dont la distribution sous le modèle de hasard n’est pas calculable analytiquement. En s’appuyant cependant sur des calculs analytiques approximatifs, nous montrons que l’apprentissage peut se focaliser uniquement sur des phénomènes relativement génériques mais difficiles à prendre en compte par le calcul. Le choix des images d’apprentissage est alors peu sensible, et un apprentissage fiable est possible à partir d’une dizaines d’images seulement.

Le chapitre 4 propose finalement une implantation “anytime” de l’algorithme de détection d’objets du chapitre 3. Pour cela, nous introduisons une architecture à base d’agents autonomes

qui exploite la capacité de l'approche *a contrario* à intégrer progressivement les données de l'image et à évaluer la pertinence de chaque donnée. En privilégiant à tout moment les données les plus prometteuses, l'algorithme obtenu se montre capable de détecter les objets les plus saillants très tôt, et les taux de détections augmentent rapidement avec le temps alloué.

Chapitre 1

La détection *a contrario*

1.1 Introduction

La prise de décision dans un algorithme de détection d'objets (au sens large) peut être vue comme un problème de test statistique d'hypothèses : étant donné un candidat ou une observation w dans l'image, il faut décider s'il est le résultat de l'hypothèse H_0 où aucun objet n'est présent, ou bien s'il est le résultat de l'hypothèse H_1 de présence d'un objet. La méthode optimale qui minimise le risque d'erreur est connue, il s'agit de la classification bayésienne [DHS01].

Cette méthode conclut qu'un objet est présent si la probabilité *a posteriori* de H_1 est supérieure à un certain seuil : $P(H_1|w) > \delta$. Le seuil δ détermine le compromis entre le taux de fausses alarmes toléré et le taux de détection. On distingue généralement deux catégories de méthodes pour estimer cette probabilité *a posteriori* : les méthodes discriminantes et les méthodes génératives.

Les méthodes discriminantes tentent d'estimer directement $P(H_1|w)$, on y trouve essentiellement les techniques d'apprentissage statistique telles que les réseaux de neurones, les machines à vecteur support ou les approches à base de boosting (adaboost, etc.) [DHS01]. La principale limitation de ces approches est la difficulté à constituer des ensembles d'apprentissage pertinents : les exemples doivent être indépendants et distribués selon la probabilité *a priori* $P(w)$. De plus, un grand nombre d'exemples est généralement requis pour obtenir une bonne estimation.

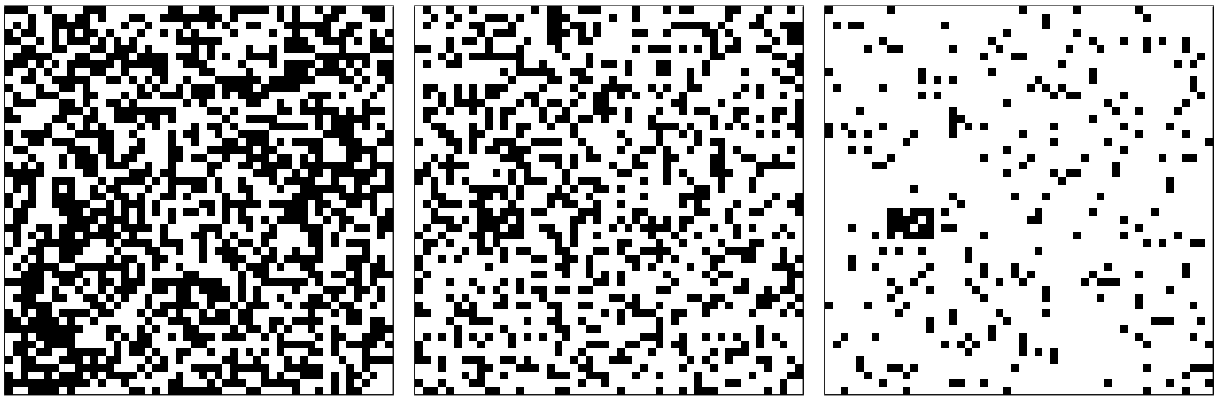
Les méthodes génératives s'appuient quant à elles sur la règle de Bayes pour estimer $P(H_1|w)$:

$$\begin{aligned}
P(H_1|w) > \delta \quad &\Leftrightarrow \quad \frac{P(w|H_1)P(H_1)}{P(w)} > \delta \\
&\Leftrightarrow \quad \frac{P(w|H_1)P(H_1)}{P(w|H_1)P(H_1) + P(w|H_0)P(H_0)} > \delta \\
&\Leftrightarrow \quad 1 + \frac{P(w|H_0)P(H_0)}{P(w|H_1)P(H_1)} < \frac{1}{\delta} \\
&\Leftrightarrow \quad \frac{P(w|H_0)}{P(w|H_1)} < (\frac{1}{\delta} - 1)\frac{P(H_1)}{P(H_0)} \\
&\Leftrightarrow \quad \frac{P(w|H_0)}{P(w|H_1)} < \delta' \\
&\Leftrightarrow \quad \frac{P_{H_0}(w)}{P_{H_1}(w)} < \delta'
\end{aligned}$$

Ce calcul classique montre que décider en fonction de $P(H_1|w)$ est équivalent à décider en fonction du ratio des vraisemblances de H_0 et H_1 pour l'observation w. Les méthodes génératives fournissent des modèles pour H_0 et H_1 permettant de calculer $P_{H_0}(w)$ et $P_{H_1}(w)$, c'est-à-dire d'expliquer comment les observations sont statistiquement *générées* sous chacune des deux hypothèses.

Dans les deux cas, il est nécessaire de spécifier explicitement et quantitativement l'apparence des objets à détecter dans une image, par des exemples pour les méthodes discriminantes, ou bien par un modèle pour les méthodes génératives. Cette tâche est souvent très difficile en vision par ordinateur, car les images naturelles sont très variables et les changements d'illumination ou d'environnement peuvent radicalement modifier l'apparence des objets. Les modèles ou les exemples proposés sont alors uniquement pertinents pour certains types d'images, et des paramètres ont souvent besoin d'être ajustés d'une image à l'autre.

Pour obtenir des algorithmes de détection génériques adaptés à tous les types d'images, Agnès Desolneux, Lionel Moisan et Jean-Michel Morel ont proposé la méthodologie *a contrario* [DMM00b], qui ne cherche pas à estimer explicitement l'apparence des objets dans l'image. L'objectif initial était de donner une formalisation mathématique à la théorie de la Gestalt [Des00], et les premiers travaux se sont attachés à détecter des groupements géométriques correspondants à des gestalts partielles, comme des alignements ou des contours. Cette méthodologie se base sur un principe énoncé par Helmholtz, selon lequel plus une structure a une probabilité faible d'être le résultat du hasard, plus elle est perceptible par notre système visuel. Les gestalts que nous percevons seraient alors des groupements non accidentels dans une image, comme l'illustre la figure 1.1.



(a) (b) (c)

FIG. 1.1 – Images de bruit où les pixels sont indépendants et identiquement distribués. Dans chaque image, des pixels noirs ont été ajoutés dans une zone rectangulaire de 6×4 pixels à la même position. Plus la proportion globale de pixels noirs P_N est faible, plus la probabilité d'obtenir une forte concentration locale de pixels noirs par hasard est faible, et plus le rectangle devient perceptuellement significatif. On peut calculer combien on s'attend à trouver de rectangles avec une densité de pixels noirs aussi forte par hasard dans chaque cas. Mathématiquement, il s'agit de l'espérance d'une variable aléatoire, appelée *NFA* (voir section 1.2). Plus cette espérance est faible, et plus le rectangle est significatif. Cette formulation permet de quantifier de façon intuitive le caractère significatif du rectangle.

Outre ces motivations phénoménologiques, le raisonnement *a contrario* s'est montré utile par la suite pour des applications dépassant le cadre initial de la théorie de la Gestalt (voir la section 1.3). Appliquée à la détection d'objets au sens large, la méthodologie consiste tout d'abord à identifier une ou plusieurs mesures discriminantes dont on suppose *a priori* que plus leurs valeurs sont grandes, plus il y a de chances qu'un objet soit présent. L'information perceptuelle portée par les mesures est ensuite quantifiée par le principe de Helmholtz en calculant la probabilité d'obtenir des valeurs aussi grandes par hasard. Plus cette probabilité est faible, et plus l'objet est perceptuellement saillant. Les objets peuvent alors être détectés en recherchant les candidats dont les mesures sont statistiquement trop élevées pour être accidentelles. Ainsi, seul un modèle du hasard également appelé modèle *a contrario* est nécessaire pour quantifier statistiquement la confiance dans la présence d'un objet.

Nous continuons ce chapitre par une formalisation mathématique plus précise de la méthodologie *a contrario*, qui reprend les concepts de l'ouvrage de référence [DMM08], mais avec une formulation parfois différente. Nous ferons ensuite un tour d'horizon des travaux *a contrario* existants, qui sont tous basés sur le cadre purement analytique établit par [DMM00b]. Nous montrerons cependant que les calculs purement analytiques permettent difficilement de combiner plusieurs mesures discriminantes ou d'utiliser des heuristiques de recherche de candidats dirigées par les données. Ces limitations motiveront le développement d'approches mixtes combinant calculs analytiques et apprentissage dans les chapitres suivants.

1.2 Formalisation mathématique

1.2.1 Notion de *PFA*

Plusieurs éléments sont nécessaires pour raisonner *a contrario* :
- Un ensemble de mesures discriminantes, représentées par des variables aléatoires. Une variable est dite discriminante si plus elle est grande, plus il y a de chance qu'un objet soit présent.
- Éventuellement, un ensemble de mesures non discriminantes, représentées également par des variables aléatoires. Nous les appellerons par la suite variables conditionnantes, car elles vont servir à prendre en compte le contexte pour évaluer le degré de confiance statistique associé aux variables discriminantes.
- Un modèle *a contrario* permettant d'estimer la distribution des variables sous l'hypothèse H_0 où leurs valeurs sont le résultat du hasard.

Exemple En s'inspirant de la figure 1.1, nous illustrons les concepts de cette section par une application dont le but est de détecter des taches noires rectangulaires dans une image. Pour cet exemple, il est naturel de prendre comme variable discriminante le nombre K de pixels noirs dans un rectangle : plus il est grand, plus il y a de chances pour qu'une tache noire soit présente. La significativité perceptuelle du nombre de pixels noirs d'un rectangle donné dépend de la densité globale P_N de pixels noirs sur l'image et de la taille L du rectangle, nous prenons donc ces deux mesures comme variables conditionnantes. Nous considérons enfin comme modèle *a contrario* un modèle où les pixels sont spatialement indépendants et identiquement distribués. Ainsi, les taches noires seront détectées à partir du moment où la concentration de pixels noirs est trop forte pour être le résultat d'un arrangement spatial accidentel de pixels.

Il est supposé *a priori* que plus les variables discriminantes sont grandes, plus un objet a de chances d'être présent. En s'appuyant sur le principe de Helmholtz, la significativité perceptuelle de l'objet pour une observation candidate w peut alors être estimée en calculant la probabilité que les variables discriminantes soient aussi grandes que celles de w par hasard.

Définition 1 (Probabilité de fausse alarme)**.** *Soient H_0 l'hypothèse de "hasard", $\mathbb{X} = \{X_1, \ldots, X_i\}$ un vecteur de variables aléatoires discriminantes, $\mathbb{Y} = \{Y_1, \ldots, Y_j\}$ un vecteur de variables aléatoires conditionnantes et w une observation candidate. On note $\mathbb{X} \geq \mathbb{X}(w)$ l'évènement $\{X_1 \geq X_1(w), X_2 \geq X_2(w), \ldots, X_i \geq X_i(w)\}$. La probabilité de fausse alarme associée à l'observation w est définie par :*

$$\mathit{PFA}(w) = P_{H_0}(\mathbb{X} \geq \mathbb{X}(w) \mid \mathbb{Y} = \mathbb{Y}(w))$$

La probabilité de fausse alarme d'une observation w mesure donc à quel point il est probable d'observer des valeurs discriminantes aussi grandes que celles de w par hasard, étant donné ses variables conditionnantes. Cette probabilité est calculée à l'aide du modèle *a contrario* de hasard choisi *a priori*.

Plus *PFA*(w) est faible, moins les variables de l'observation w sont susceptibles d'être aussi grandes par hasard, et donc, *a contrario*, plus elles sont susceptibles d'être associées à un objet à détecter. La probabilité de fausse alarme permet de classer les observations par degré de confiance : on dira qu'une observation w_1 est plus significative, et donc plus probablement associée à un objet qu'une observation w_2 si *PFA*$(w_1) <$ *PFA*(w_2). Le rôle du modèle *a contrario* est donc de servir de référence statistique pour évaluer la confiance dans la présence d'un objet pour chacune des observations, en fonction des mesures discriminantes et du contexte.

Remarque Par souci de simplicité, ce chapitre se focalise sur des variables discriminantes telles que plus leurs valeurs sont grandes, plus un objet a de chances d'être présent. Il est bien entendu possible de raisonner de façon opposée avec des variables telles que plus leurs valeurs sont petites, plus un objet a de chances d'être présent. Les évènements considérés par la probabilité de fausse alarme seraient alors de type $X_i \leq X_i(w)$. De tels évènements seront utilisés dans le chapitre 3.

Exemple Pour la détection de taches noires, le vecteur discriminant pour un rectangle R contient une seule variable K, le nombre de pixels noirs dans le rectangle. Le vecteur conditionnant contient les deux variables P_N et L, respectivement la densité globale de pixels noirs sur l'image et la taille de R. La *PFA* de R est donc donnée par :

$$\textit{PFA}(R) = P_{H_0}(K \geq K(R) \mid L = L(R), P_N)$$

Le modèle *a contrario* choisi pour H_0 considère que les pixels sont indépendants et identiquement distribués. Chacun des pixels de R peut donc être vu comme une variable de Bernouilli Z_i qui vaut 1 (noir) avec une probabilité P_N et 0 (blanc) avec une probabilité $1 - P_N$. Le nombre de pixels noirs K dans le rectangle correspond alors à la somme des $L(R)$ variables Z_i, sa loi est donc binomiale, de paramètres $L(R)$ et P_N :

$$P_{H_0}(K \geq K(R) \mid L = L(R), P_N) = \mathcal{B}_{\geq}(K(R), L(R), P_N)$$

avec $\mathcal{B}_{\geq}(k, n, p) = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$ la queue de la loi binomiale de paramètres n et p appliquée en k.

1.2.2 Notion d'algorithme ε-fiable

Il faut maintenant déterminer un seuil à partir duquel les observations candidates sont jugées suffisamment significatives et doivent être détectées. Une *PFA* est difficile à seuiller dans l'absolu, car elle ne correspond pas à une quantité physique intuitive. De plus, dans une image on analyse rarement une seule observation candidate, mais plutôt un ensemble. Par exemple, pour la détection de taches noires, si l'on ne sait pas à l'avance où peuvent se situer les taches, il faudra analyser tous les rectangles de l'image.

Il est donc plus naturel de s'intéresser à la probabilité de produire une fausse alarme sur l'ensemble des candidats testés. Mais comme cela a été mis en évidence dans [DMM08], cette probabilité est très difficile à estimer, du moins analytiquement, car les observations ne sont généralement pas indépendantes entre elles. Toujours dans l'exemple des taches noires, les rectangles possibles dans l'image se superposent parfois et ne sont donc pas tous indépendants. Aussi l'approche de [DMM08] se fixe comme objectif de borner l'espérance du nombre de fausses alarmes produit par un algorithme de détection sur une image, beaucoup plus simple à calculer grâce à la linéarité de l'espérance.

Nous traduisons cela par la notion suivante, qui englobe les définitions que l'on peut trouver dans la plupart des travaux précédents, le plus souvent dédiées à une application précise.

Définition 2 (Algorithme ε-fiable)**.** *Un algorithme de détection produit une fausse alarme s'il considère qu'une observation candidate est associée à un objet alors qu'elle est le résultat de l'hypothèse de hasard H_0, selon le modèle a contrario retenu. Nous dirons qu'un algorithme de détection est ε-fiable si l'espérance du nombre de fausses alarmes produit par cet algorithme sur une image est inférieure à ε.*

L'ε-fiabilité d'un algorithme est donc une garantie de robustesse. Elle assure qu'en moyenne, il y aura moins de ε fausses détections par image, c'est-à-dire moins de ε objets détectés alors que leurs propriétés sont le résultat du hasard. Classiquement, ε est fixé à 1 [DMM08], ce qui correspond à une moyenne de moins d'une fausse alarme par image. Le choix d'ε n'est toutefois pas très sensible, car il n'aura généralement qu'une influence logarithmique sur les seuils finaux appliqués aux variables discriminantes.

L'ε-fiabilité fournit donc un objectif à atteindre facilement interprétable, et va diriger le choix des seuils de signifiance sur la *PFA* dans les algorithmes de détection.

Remarque La notion d'algorithme ε-fiable est très proche de la définition générale de *NFA* (Nombre de Fausses Alarmes) qui a été proposée dans [GM06]. Cette dernière assure qu'un algorithme ne fera pas plus de ε détection, en moyenne, dans un ensemble où *toutes* les observations sont le résultat de l'hypothèse H_0. Avec notre définition, nous cherchons à borner le nombre de fausses alarmes de façon plus générale sur une image arbitraire, pouvant contenir à la fois des observations accidentelles associées à l'hypothèse H_0, et des observations associées à la présence d'un objet. En pratique, l'utilisation d'un *NFA* de [GM06] mène quasi-systématiquement à un algorithme ε-fiable, mais nous verrons dans la section 2.3.8 (chapitre 2) qu'il est parfois nécessaire d'introduire des conditions supplémentaires en présence d'heuristiques de parcours de l'image dirigées par les données.

1.2.3 Processus *a contrario* classique

L'algorithme 1 résume le processus classique de détection *a contrario*. C'est cet algorithme que nous allons chercher à rendre ε-fiable pour chaque application.

Algorithme 1 : Algorithme de détection *a contrario*.

(1) Extraire de l'image un ensemble d'observations $W = \{w_1, w_2, \ldots, w_N\}$;
(2) Conserver les observations w_i telles que $\mathit{PFA}(w_i) < \varepsilon_{\text{pfa}}$;
(3) (Optionnel) Si des observations sont incluses dans d'autres, ne conserver que les plus significatives (principe de maximalité) ;

$\Rightarrow$ Les objets détectés correspondent aux observations conservées.

Le nombre de fausses alarmes produites par l'algorithme 1 repose donc sur le seuil ε_{pfa}. Nous illustrons maintenant comment ce seuil peut être obtenu de façon automatique en cherchant à rendre l'algorithme de détection de taches noires ε-fiable.

1.2.4 Application à la détection de taches noires

Le but de l'exemple tiré de la figure 1.1 est de détecter les taches noires dans une image. Nous avons déjà proposé dans la section 1.2.1 une probabilité de fausse alarme pour évaluer à quel point la concentration de pixels noirs d'un rectangle est significativement forte. Pour rechercher tous les endroits de l'image où une tache noire est présente, nous analysons tous les rectangles possibles dans l'image. Ceci constitue l'ensemble W de l'algorithme 1. Pour chaque rectangle candidat R, une tache sera détectée si $\mathit{PFA}(R)$ est inférieure à un seuil ε_{pfa} (étape 2 de l'algorithme 1).

Pour chaque tache détectée, la probabilité qu'il s'agisse d'une fausse alarme (au sens de la définition 2) est donc inférieure à ε_{pfa}. On peut alors prouver analytiquement (voir section 1.4) que l'espérance du nombre de fausses alarmes est nécessairement inférieure à $N_r \times \varepsilon_{\text{pfa}}$, avec N_r le nombre total de rectangle analysés dans l'image. Pour garantir que l'algorithme produise en moyenne moins de ε fausses alarmes par image, il suffit alors de fixer $\varepsilon_{\text{pfa}} = \frac{\varepsilon}{N_r}$.

Le nombre de rectangles N_r dépend de la taille de l'image. Pour une image de taille $N \times M$, il vaut :

$$N_r = \frac{N(N+1)}{2} \times \frac{M(M+1)}{2}$$

En raisonnant *a contrario*, la détection de taches noires peut donc être effectuée de façon purement analytique et sans autre paramètre libre que le nombre moyen de fausses alarmes statistiquement toléré ε.

Remarque Détecter les rectangles R tels que $\mathit{PFA}(R) < \frac{\varepsilon}{N_r}$ est équivalent à détecter les rectangles R tels que $\mathit{NFA}(R) = N_r \times \mathit{PFA}(R) < \varepsilon$. Cette quantité *NFA* est appelée nombre de fausses alarmes dans les travaux *a contrario*. Dans cet exemple, $\mathit{NFA}(R)$ correspond à l'espérance du nombre de rectangles au moins aussi significatifs que R dans une image issue du modèle *a contrario* et permet d'interpréter quantitativement le degré de saillance perceptuelle d'une tache noire. Dans la figure 1.1(a), le *NFA* associé au rectangle artificiellement ajouté est

d'environ 10^3, ce qui signifie que dans une image 50×50 où les pixels sont indépendants et identiquement distribués (i.i.d.), on s'attend à obtenir un millier de rectangles aussi significativement noirs que celui-là. Dans la figure 1.1(b), le *NFA* associé au rectangle tombe à 2×10^{-1}, et le rectangle commence à devenir perceptuellement significatif. Dans la figure 1.1(c), le *NFA* est de l'ordre de 10^{-10}, ce qui signifie qu'il faudrait générer plus de 100 milliards d'images i.i.d. pour espérer observer une concentration aussi forte de pixels noirs dans un rectangle. L'évènement est donc très significatif, et nous le percevons immédiatement. Nous ne reposons pas explicitement sur cette notion de *NFA* dans ce chapitre car elle est difficile à définir de façon générique, et ne prend tout son sens que dans des cas concrets. C'est pourquoi nous avons choisi d'introduire à la place la notion d'ε-fiabilité.

1.3 Applications existantes

Nous faisons maintenant un tour d'horizon des travaux *a contrario* existants en se focalisant sur leur façon de modéliser *a contrario* le problème de détection. Pour les applications les plus représentatives, nous donnons le type d'éléments analysés (l'ensemble W de candidats de l'algorithme 1), les variables discriminantes et les variables conditionnantes proposées. Ce niveau de détail permet de faire ressortir les points communs entre les travaux existants, qui sont presque tous basés sur des calculs purement analytiques. Ceci donne une intuition sur les limites du cadre purement analytique, qui seront discutées plus formellement dans la section 1.4.

Détection d'alignements [DMM00a], figure 1.2 Le but est de détecter des segments dans l'image dont la proportion de points partageant la même orientation locale est significativement grande. Les candidats sont ici tous les segments de droite de l'image, qui peuvent être définis par leur point de départ et d'arrivée. Dans le modèle *a contrario* les pixels sont spatialement indépendants, et les orientations locales sont uniformément distribuées. Pour un segment S la variable discriminante est K le nombre de points alignés avec le segment, et les variables conditionnantes sont L la longueur du segment, et p la précision pour la quantification des orientations ($\frac{1}{16}$). La *PFA* déduite est :

$$PFA(S) = P_{H_0}(K \geq K(S) \mid L = L(S), p) = \mathcal{B}_{\geq}(K(S), L(S), p)$$

Une version beaucoup plus rapide a été proposée dans [GJMR08], en utilisant l'heuristique de [BHR86] pour analyser un ensemble de candidats réduit.

Détection de points de fuite [ADV03] Un point de fuite est détecté si une concentration significative de droites s'intersectent dans une région de l'espace (extérieure ou intérieure à l'image). Les droites sont données par le détecteur d'alignements précédent. L'espace est échantillonné en régions disjointes couvrant le plan de l'image. Ces régions forment l'ensemble des candidats. Un point de fuite est déclaré présent dans une région si le nombre d'intersections de droites dans

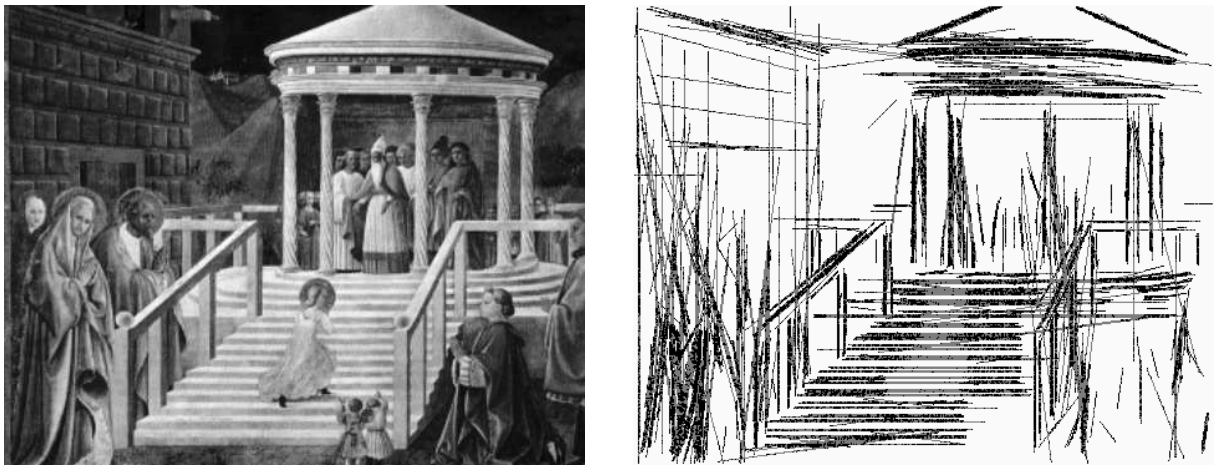


FIG. 1.2 – Détection *a contrario* d'alignements [DMM00a].

cette région est significatif. Le modèle *a contrario* suppose que les droites ont des coordonnées polaires uniformément distribuées. La variable discriminante choisie pour une région R est K le nombre d'intersections dans la région, et les variables conditionnantes sont N_d le nombre de droites total dans l'image, et P_r la probabilité *a priori* qu'une droite passe par la région R. Les régions sont choisies de façon à ce que P_r soit constante, nous ne détaillons pas le calcul ici. On en déduit la *PFA* associée à une région R :

$$\mathit{PFA}(R) = P_{H_0}(K \geq K(R) \mid N_d, P_r) = \mathcal{B}_{\geq}(K(R), N_d, P_r)$$

Détection de contours [DMM01], figure 1.3 Un contour est défini comme une ligne de niveau dont le contraste est significativement élevé. Les candidats sont l'ensemble des lignes de niveau d'une image. Dans le modèle *a contrario* les intensités de gradient sont indépendantes le long de chaque ligne de niveau. La variable discriminante choisie pour une ligne de niveau L est μ l'intensité de gradient minimale le long de la ligne. Les variables conditionnantes sont K la longueur de la ligne de niveau et $g[X \geq x]$ la distribution globale des intensités de gradient, estimée empiriquement sur l'image. On en déduit la *PFA* suivante :

$$\mathit{PFA}(L) = P_{H_0}(\mu \geq \mu(L) \mid K = K(L), g) = (g[\mu(L)])^{K(L)}$$

Détection de mouvement [VCB06] Une région est dite en mouvement si le nombre de points de la région dépassant un certain seuil de déplacement est significativement grand. Nous considérons ici une version simplifiée avec un seul seuil de mouvement choisi *a priori*. L'ensemble des régions analysées peut être fourni, par exemple, par la détection de contours fermés précédente, ce qui donne l'ensemble des candidats. Pour une région R, la variable discriminante est K le nombre de pixels pour lesquels le mouvement est supérieur au seuil, et les variables conditionnantes sont N la taille de la région et P_δ la probabilité globale sur l'image que le

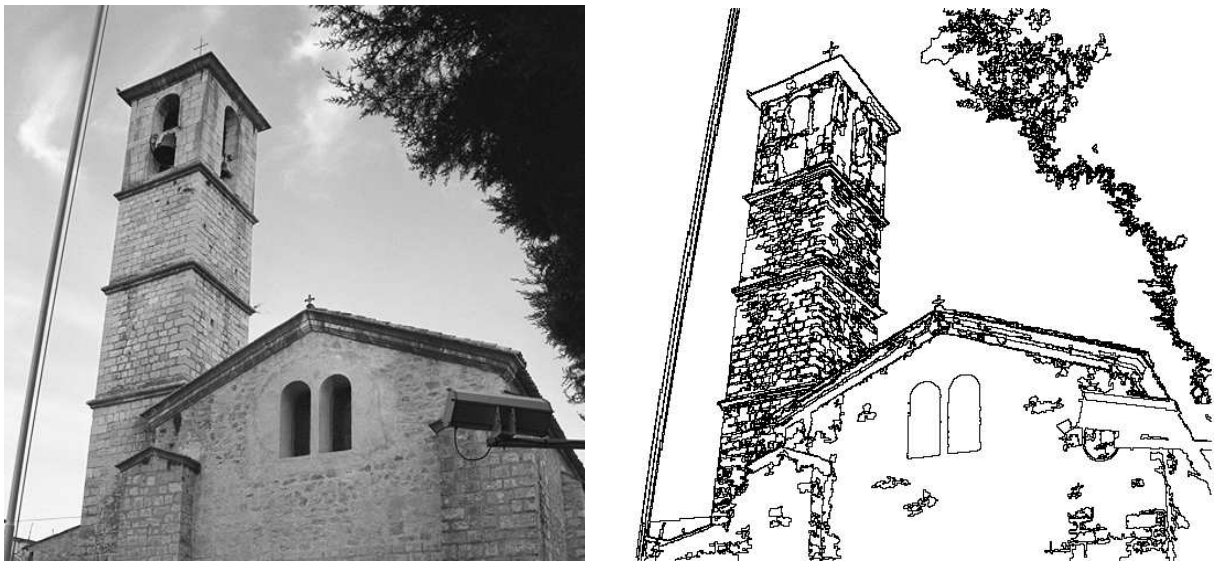


FIG. 1.3 – Détection *a contrario* de lignes de niveau significativement contrastées [DMM01].

mouvement d'un pixel soit supérieur au seuil. On en déduit la *PFA* :

$$\mathit{PFA}(R) = P_{H_0}(K \geq K(R) \mid N = N(R), P_\delta) = \mathcal{B}_{\geq}(K(R), N(R), P_\delta)$$

Modes d'un histogramme [DMM03] Un mode d'histogramme est défini comme un intervalle sur lequel il y a une densité significativement grande de points. Les candidats sont tous les intervalles possibles dans un histogramme. Pour chaque intervalle I, la variable discriminante est K le nombre de points dans I, et les variables conditionnantes sont le nombre total M de points et L la taille de I. Sous l'hypothèse *a contrario*, les points sont uniformément répartis, et la probabilité qu'un point tombe dans un intervalle I est donc le rapport entre la longueur $L(I)$ de I et la longueur totale L_t de l'histogramme. On en déduit la *PFA* d'un intervalle I :

$$\mathit{PFA}(I) = P_{H_0}(K \geq K(I) \mid L = L(I), M) = \mathcal{B}_{\geq}(K(I), M, \frac{L(I)}{L_t})$$

Basée sur le même principe, une version plus élaborée de segmentation d'histogramme *a contrario* est proposée dans [DDLP07].

Détection de groupements [CDD$^+$07], figure 1.4 Le but est de détecter des groupes de points dans l'espace dont la concentration est anormalement élevée. Il s'agit d'une version évoluée et générique de l'exemple de la figure 1.1, adaptée aux espaces continus à n dimensions. Étant donné une région R de l'espace, la variable discriminante choisie est le nombre K de points dans R, et les variables conditionnantes sont le nombre total M de points dans l'espace et π, la probabilité *a priori* qu'un point tombe dans une région (dans un espace fini à deux dimensions, un exemple de $\pi(R)$ est le rapport entre l'aire de R et l'aire totale de l'espace). La *PFA* qui en

découle est :

$$\mathit{PFA}(R) = P_{H_0}(K \geq K(R) \mid M, \pi) = \mathcal{B}_{\geq}(K(R), M, \pi(R))$$

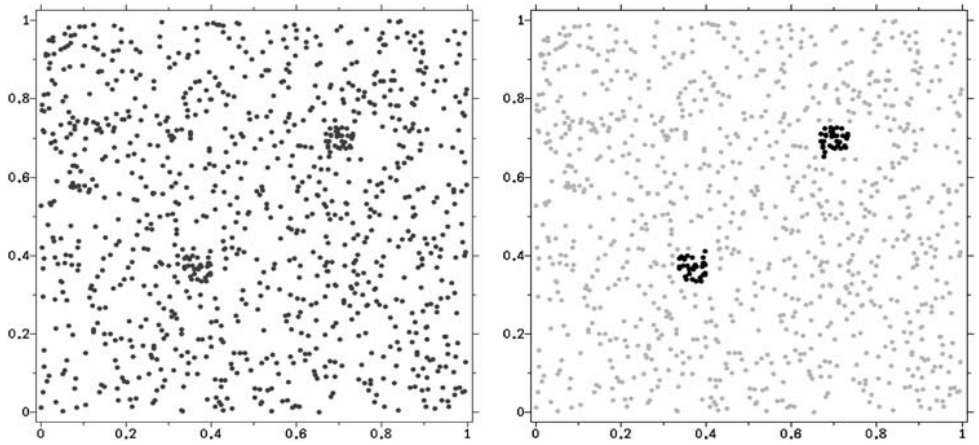


FIG. 1.4 – Détection *a contrario* de groupements [CDD+07].

Mise en correspondance d'objets [MSC+06, CDD+07], figure 1.5 L'objectif est de déterminer si une ligne de niveau extraite d'une image correspond à une des lignes de niveau d'une base de données. Une application directe est la mise en correspondance de deux images, où l'on cherche, pour chacune des lignes de niveau de la première image, une correspondance dans l'ensemble des lignes de niveau de la deuxième image. Pour évaluer la similarité entre deux lignes de niveau S et S', six mesures de distance $\{D_i(S, S')\}_{i \in 1 \ldots 6}$ indépendantes et normalisées sont proposées. La variable discriminante choisie est finalement $D(S, S') = \max_i D_i(S, S')$ et la *PFA* qui en découle est :

$$\mathit{PFA}(S, S') = P_{H_0}(D \geq D(S, S'))$$

La *PFA* est estimée empiriquement à partir de la base de lignes de niveau, en calculant les distances des couples de lignes de niveau dans la base.

Autres travaux Parmi les autres travaux *a contrario*, on notera la détection de taches appliquée à l'analyse de mammographies [GM06], l'estimation de profondeur à partir de paires d'images stéréo [IPG+07], la détection de changements dans des images satellitaires [RMlHM07], l'estimation de la matrice fondamentale en vision stéréo [MS04], ou bien la mise en correspondance de points d'intérêt [RGD07]. Leurs modélisations *a contrario* sont similaires à celles des exemples détaillés précédemment.

Dans tous ces travaux, mis à part la reconstruction tridimensionnelle de [IPG+07] qui sera traité explicitement dans la section 1.4, il est possible déterminer automatiquement le seuil ε_{pfa} qui assure l'ε-fiabilité de la détection. Pour une image donnée ce seuil de fiabilité est le plus souvent $\varepsilon_{\text{pfa}} = \frac{\varepsilon}{\#W}$, $\#W$ étant le nombre de candidats analysés dans l'image. Ce résultat n'est

FIG. 1.5 – Mise en correspondance de lignes de niveaux *a contrario* [MSC+06].

cependant pas universel et résulte des propriétés communes à ces travaux, que nous dégageons dans la section 1.4.

Remarque Certains travaux [RMlHM07, GM06] ont montré qu'il est parfois souhaitable d'introduire des seuils multiples sur la *PFA*, qui dépendent de certaines propriétés des observations. Ceci permet de seuiller la *PFA* différemment en fonction de la catégorie à laquelle appartient l'observation, par exemple pour prendre en compte le nombre de candidats dans la même catégorie. Nous en verrons un exemple dans la section 2.3 (chapitre 2). Nous ne considérons cependant dans ce chapitre que le cas plus simple où un seuil unique est utilisé, car l'utilisation de seuils multiples ne change pas les conditions d'applications du cadre *a contrario* purement analytique.

1.4 Applicabilité du cadre *a contrario* purement analytique

1.4.1 Proposition fondatrice

Les preuves d'ε-fiabilité des travaux *a contrario* de la section 1.3 reposent toutes sur les mêmes principes, que nous regroupons de façon générique à travers la proposition suivante.

Proposition 1. *Soit $W = \{w_1, w_2, \ldots, w_N\}$ l'ensemble des candidats analysés dans une image par l'algorithme 1. Soit H_0 l'hypothèse nulle selon laquelle un évènement est le résultat du hasard. Soit X une variable aléatoire réelle et $\mathbb{Y}$ un vecteur de variables aléatoires réelles. On note $\#\Omega$ le cardinal d'un ensemble Ω. Si la PFA associée à une observation est définie par :*

$$PFA(w_i) = P_{H_0}(X \geq X(w_i) \mid \mathbb{Y} = \mathbb{Y}(w_i))$$

alors le seuil $\varepsilon_{pfa} = \frac{\varepsilon}{\#W}$ assure l'ε-fiabilité de l'algorithme de détection. 1.

Démonstration. Nous rappelons d'abord le lemme suivant :

Lemme 1. *Soit X une variable aléatoire réelle, et $F(x) = P(X \geq x)$ sa fonction de répartition complémentaire. Alors :*

$$\forall t \in [0,1], \quad P(F(X) \leq t) \leq t$$

Remarque : l'égalité est obtenue si $F(x)$ est continue. Une démonstration de ce lemme est donnée dans [GM06]. Il signifie que la probabilité d'obtenir une valeur x pour X telle que $P(X \geq x)$ soit inférieure à un réel t est elle-même inférieure à t.

Ce lemme peut être étendu à des distributions conditionnelles.

Lemme 2. *Soit X une variable aléatoire réelle, $\mathbb{Y}$ un vecteur aléatoire réel, et $F(x,y) = P(X \geq x \mid \mathbb{Y} = y) = P_y(X \geq x)$ la fonction de répartition complémentaire de X conditionnellement à $\mathbb{Y}$. Alors :*

$$\forall t \in [0,1], \quad P(F(X,\mathbb{Y}) \leq t) \leq t$$

Démonstration. Nous développons le cas où X est discrète, le cas continu est similaire et se déduit immédiatement du Lemme 2 de [GM06] :

$$\begin{array}{rcl} P(F(X,\mathbb{Y}) \leq t) & = & \sum_y P(\mathbb{Y} = y) \times P(F(X,y) \leq t \mid \mathbb{Y} = y) \\ & = & \sum_y P(\mathbb{Y} = y) \times P_y(F(X,y) \leq t) \end{array}$$

P_y est une mesure de probabilité, et l'on peut donc appliquer le Lemme 1 :

$$\begin{array}{rcl} P(F(X,\mathbb{Y}) \leq t) & \leq & \sum_y P(\mathbb{Y} = y) \times t \\ & \leq & t \end{array}$$

□

Nous avons maintenant les outils pour démontrer la proposition 1. Soit Z_i la variable de Bernouilli qui vaut 1 si l'observation w_i est une fausse alarme, c'est-à-dire si on a $\mathit{PFA}(w_i) < \varepsilon_{\text{pfa}}$ alors que l'hypothèse sous-jacente pour cette observation est en réalité H_0. L'espérance $\mathbb{E}_{\#\text{FA}}$ du nombre de fausses alarmes produites par l'algorithme 1 s'écrit :

$$
\begin{aligned}
\mathbb{E}_{\#\text{FA}} &= \mathbb{E}\left(\#\left\{w_i \in W; Z_i = 1\right\}\right) \\
&= \mathbb{E}\left(\sum_{i=1}^{\#W} Z_i\right) \\
&= \sum_{i=1}^{\#W} \mathbb{E}\left(Z_i\right) && \text{(linéarité de l'espérance)} \\
&= \sum_{i=1}^{\#W} P(Z_i = 1) \\
&= \sum_{i=1}^{\#W} P(\textit{PFA}(w_i) < \varepsilon_{\text{pfa}}, H_0) \\
&= \sum_{i=1}^{\#W} P(\textit{PFA}(w_i) < \varepsilon_{\text{pfa}} \mid H_0) \times P(H_0) \\
&\leq \sum_{i=1}^{\#W} P(\textit{PFA}(w_i) < \varepsilon_{\text{pfa}} \mid H_0) && (P(H_0) \leq 1) \\
&\leq \sum_{i=1}^{\#W} P_{H_0}(P_{H_0}(X \geq X(w_i) \mid \mathbb{Y} = \mathbb{Y}(w_i)) < \varepsilon_{\text{pfa}}) \\
&\leq \sum_{i=1}^{\#W} \varepsilon_{\text{pfa}} && \text{(Lemme 2)} \\
&\leq \#W \times \varepsilon_{\text{pfa}}
\end{aligned}
$$

En choisissant $\varepsilon_{\text{pfa}} = \frac{\varepsilon}{\#W}$, on a bien une espérance du nombre de fausses alarmes inférieure à ε, ce qui assure l'ε-fiabilité de l'algorithme 1. □

Cette proposition permet de choisir analytiquement le seuil de décision *a contrario*. Nous analysons maintenant plus précisément deux de ses conditions d'application.

1.4.2 Une seule variable discriminante

Le lemme 1 n'est valable que pour une seule variable discriminante, il n'existe pas de méthode systématique et simple pour estimer $P(P(X_1 \geq x_1, X_2 \geq x_2) < \varepsilon_{\text{pfa}})$. Ceci explique probablement qu'il n'existe qu'une seule publication relative à une application *a contrario* analytique utilisant deux variables discriminantes, la seconde partie de la détection de groupements significatifs de [CDD$^+$07]. Dans ce cas précis, les deux variables sont discrètes et identiquement distribuées, et donc une borne a pu être trouvée. Si les variables sont normalisées et de même nature, elles peuvent également être regroupées par un opérateur de type min ou moyenne. Mais dans le cas général, le calcul est complexe, même pour deux variables.

1.4.3 Distribution de la variable discriminante estimable analytiquement

Pour effectuer la détection d'objets de façon totalement analytique, il est nécessaire de pouvoir calculer la probabilité de fausse alarme associée à la variable discriminante de façon analytique. C'est le cas dans tous les travaux précédents à l'exception de [MSC$^+$06] et [RGD07]. Les probabilités de fausses alarmes ont été estimées empiriquement à partir de la base des formes à reconnaître dans [MSC$^+$06] et à partir de la base de descripteurs à reconnaître dans [RGD07]. En ce sens, ces travaux sont les premiers exemples de méthodes mixant calculs analytiques et estimations empiriques.

1.4.4 Candidats choisis indépendamment de la variable discriminante

Une condition implicite de la proposition 1 est l'indépendance de la variable discriminante vis-à-vis du choix des candidats. Cette condition est vérifiée dans la quasi-totalité des travaux *a contrario*, car tous les candidats possibles dans l'image sont analysés. Plusieurs exceptions sont à noter : l'estimation de la matrice fondamentale de [MS04], la détection de changements de [RMlHM07], la détection de segments de [GJMR08] et l'analyse stéréo de [IPG$^+$07]. Dans ces travaux, des heuristiques dirigées par les données sont utilisées pour ne considérer que les candidats les plus susceptibles d'être significatifs et accélérer les temps de calcul. L'utilisation d'heuristique amène l'algorithme de détection 2, légère variation de l'algorithme 1.

Algorithme 2 : Algorithme de détection *a contrario* avec heuristique.

(1) Déterminer un ensemble d'observations $W = \{w_1, w_2, \ldots, w_N\}$ dans l'image ;
(2) Utiliser l'heuristique $\mathcal{H}$ pour extraire le sous-ensemble $\mathcal{H}_W \subset W$ à analyser ;
(3) Conserver les observations w_i de $\mathcal{H}_W$ telles que $\mathit{PFA}(w_i) < \varepsilon_{\text{pfa}}$;
(4) (Optionnel) Si des observations sont incluses dans d'autres, ne conserver que les plus significatives (principe de maximalité) ;

$\Rightarrow$ Les objets détectés correspondent aux observations conservées.

Une étape a été ajoutée dans l'algorithme 2 pour sélectionner les candidats à analyser grâce à une heuristique $\mathcal{H}$. En pratique, cette heuristique est choisie de manière à diminuer au maximum le nombre de candidats, tout en gardant ceux qui sont le plus susceptibles d'être significatifs.

Considérons d'abord le choix de [GJMR08], [MS04] et [RMlHM07] à travers un exemple de détection *a contrario* avec une variable discriminante X et donc $\mathit{PFA}(w) = P_{H_0}(X \geq X(w))$. Estimons le nombre de fausses alarmes effectué par l'algorithme 2 en utilisant une heuristique $\mathcal{H}$. Pour cela, on note par $\mathcal{H}(w)$ le fait qu'un candidat w soit sélectionné par $\mathcal{H}$. L'espérance du nombre de fausses alarmes est l'espérance du nombre de candidats sélectionnés par $\mathcal{H}$ et considérés significatifs alors qu'ils sont en réalité le résultat de l'hypothèse de hasard H_0 :

$$
\begin{aligned}
\mathbb{E}_{\#\text{FA}} &= \sum_{w \in W} P(\mathcal{H}(w), \textit{PFA}(w) < \varepsilon_{\text{pfa}}, H_0) \\
&= \sum_{w \in W} P(\mathcal{H}(w) \mid \textit{PFA}(w) < \varepsilon_{\text{pfa}}, H_0) \times P(\textit{PFA}(w) < \varepsilon_{\text{pfa}} \mid H_0) \times P(H_0) \\
&\leq \sum_{w \in W} P(\mathcal{H}(w) \mid \textit{PFA}(w) < \varepsilon_{\text{pfa}}, H_0) \times P(\textit{PFA}(w) < \varepsilon_{\text{pfa}} \mid H_0) \\
&\leq \sum_{w \in W} P(\mathcal{H}(w) \mid \textit{PFA}(w) < \varepsilon_{\text{pfa}}, H_0) \times P_{H_0}(P_{H_0}(X \geq X(w)) < \varepsilon_{\text{pfa}}) \\
&< \sum_{w \in W} \varepsilon_{\text{pfa}} \times P_{H_0}(\mathcal{H}(w) \mid \textit{PFA}(w) < \varepsilon_{\text{pfa}}) \qquad \text{(Lemme 1)} \\
&< \sum_{w \in W} \varepsilon_{\text{pfa}} \\
&< \#W \times \varepsilon_{\text{pfa}}
\end{aligned}
$$

Ce calcul montre que, bien évidemment, choisir $\varepsilon_{\text{pfa}} = \frac{\varepsilon}{\#W}$ assure toujours l'ε-fiabilité de l'algorithme, car $P_{H_0}(\mathcal{H}(w) \mid \textit{PFA}(w) < \varepsilon_{\text{pfa}})$ est inférieure ou égale à 1. Ce facteur correspond à la probabilité que l'heuristique sélectionne un candidat w sachant qu'il est accidentellement significatif. Cette probabilité vaut 1 si l'heuristique sélectionne tous les candidats, ce qui revient au cas de l'algorithme 1. Elle est également très proche de 1 si l'heuristique ne rate que très peu de candidats avec une *PFA* faible. C'est ce qui se passe dans la détection de segments de [GJMR08], qui peut donc choisir $\varepsilon_{\text{pfa}} = \frac{\varepsilon}{\#W}$ et conserver un taux de détection proche de l'optimal. La seule limitation dans ce cas est la calculabilité de $\#W$, qui peut s'avérer complexe, comme nous le verrons dans la section 2.2. Pour [MS04] et [RMlHM07], un algorithme de type RANSAC [FB81] est utilisé. Grâce à un nombre d'itérations suffisant, la probabilité de rater un candidat significatif est faible. Ces approches peuvent donc également considérer que l'heuristique est quasi-optimale, et utiliser un seuil $\varepsilon_{\text{pfa}} = \frac{\varepsilon}{\#W}$.

L'autre extrême consiste à ignorer l'influence de l'heuristique sur les propriétés des candidats. Estimons à nouveau l'espérance du nombre de fausses alarmes $\mathbb{E}_{\#\text{FA}}$, mais avec un autre

développement :

$$\begin{aligned}
\mathbb{E}_{\#\text{FA}} &= \sum_{w \in W} P(\mathcal{H}(w), \textit{PFA}(w) < \varepsilon_{\text{pfa}}, H_0) \\
&= \sum_{w \in W} P(\mathcal{H}(w), \textit{PFA}(w) < \varepsilon_{\text{pfa}}, H_0) \\
&= \sum_{w \in W} P(\textit{PFA}(w) < \varepsilon_{\text{pfa}} \mid \mathcal{H}(w), H_0) \times P(\mathcal{H}(w), H_0) \\
&= \sum_{w \in W} P_{H_0}(\textit{PFA}(w) < \varepsilon_{\text{pfa}} \mid \mathcal{H}(w)) \times P(\mathcal{H}(w)) \times P(H_0 | \mathcal{H}(w)) \\
&\leq \sum_{w \in W} P_{H_0}(\textit{PFA}(w) < \varepsilon_{\text{pfa}} \mid \mathcal{H}(w)) \times P(\mathcal{H}(w))
\end{aligned}$$

Soit $\mathcal{H}_W \subset W$ le sous-ensemble de candidats extrait de W par $\mathcal{H}$. La probabilité *a priori* qu'un candidat soit sélectionné par $\mathcal{H}$ est simplement :

$$P(\mathcal{H}(w)) = \frac{\#\mathcal{H}_W}{\#W}$$

Si on considère maintenant que les candidats sont choisis par $\mathcal{H}$ indépendamment de la variable discriminante X, on a :

$$P_{H_0}(\textit{PFA}(w) < \varepsilon_{\text{pfa}} \mid \mathcal{H}(w)) = P_{H_0}(\textit{PFA}(w) < \varepsilon_{\text{pfa}}) < \varepsilon_{\text{pfa}} \qquad \text{(Lemme 1)}$$

On obtient finalement :

$$\begin{aligned}
\mathbb{E}_{\#\text{FA}} &\leq \sum_{w \in W} P_{H_0}(\textit{PFA}(w) < \varepsilon_{\text{pfa}} \mid \mathcal{H}(w)) \times P(\mathcal{H}(w)) \\
&\leq \sum_{w \in W} \varepsilon_{\text{pfa}} \times \frac{\#\mathcal{H}_W}{\#W} \\
&\leq \#W \times \varepsilon_{\text{pfa}} \times \tfrac{\#\mathcal{H}_W}{\#W} \\
&\leq \#\mathcal{H}_W \times \varepsilon_{\text{pfa}}
\end{aligned}$$

Dans ce cas, en choisissant $\varepsilon_{\text{pfa}} = \frac{\varepsilon}{\#\mathcal{H}_W}$ on assure également l'ε-fiabilité de la détection. Il faut bien noter la différence avec le cas précédent, ici on divise ε par le nombre de candidats analysés par l'heuristique, alors que dans le cas précédent on divisait ε par le nombre total de candidats possibles dans l'image, souvent bien plus grand.

Cette analyse repose donc sur l'hypothèse que l'heuristique choisit ses candidats indépendamment de la variable discriminante et donc indépendamment de leur significativité. C'est par exemple le cas si les candidats sont échantillonnés de façon purement aléatoire. Cependant, le but de l'heuristique est le plus souvent de choisir efficacement les quelques candidats qui ont le plus de chance d'être significatifs, et cette hypothèse d'indépendance n'est alors plus vérifiée. Dans [IPG+07], le calcul de significativité est uniquement utilisé de façon relative pour décider si deux régions sont plus significatives séparées ou regroupées. Un seuil absolu de significativité n'est donc pas nécessaire dans ce cas, et l'hypothèse d'indépendance donne de bons résultats. Mais dans la plupart des applications, il faudra tenir compte de la dépendance de l'heuristique par rapport aux variables discriminantes.

En dehors de ces deux cas extrêmes, il est difficile d'estimer analytiquement un seuil de significativité en présence d'une heuristique car elles sont le plus souvent définies de façon purement algorithmique, et non mathématique.

1.4.5 Conclusion

Pour détecter des objets *a contrario* de façon purement analytique, nous avons identifié certaines conditions :

- Une seule variable discriminante, dont la distribution sous le modèle *a contrario* est connue analytiquement ;
- Des structures géométriques choisies *a priori*, c'est-à-dire indépendamment des données. L'utilisation d'heuristique dirigée par les données a été limitée au cas extrême où l'heuristique peut être considérée parfaite (aucune structure significative n'est ignorée). Il est également nécessaire de pouvoir dénombrer analytiquement le nombre de candidats implicitement ou explicitement analysés.

[MSC+06] et [RGD07] ont effectué un premier pas vers le relâchement de ces contraintes en montrant que les probabilités de fausses alarmes pouvaient parfois être estimées empiriquement. Nous allons voir dans les chapitres 2 et 3 comment il est possible d'aller plus loin en mixant calculs analytiques et apprentissage statistique à partir d'exemples d'images issues du modèle *a contrario*.

Chapitre 2

Apprentissage *a contrario* bas niveau à partir d'images de bruit blanc

2.1 Introduction

Dans ce chapitre, nous allons montrer comment il est possible de combiner calculs analytiques et apprentissage pour des primitives de bas niveau. Ces primitives ont comme point commun de travailler directement au plus près de l'image en cherchant des groupes de pixels ayant certaines propriétés. Le modèle *a contrario* classique dans ce cas consiste à supposer que les pixels sont indépendants et identiquement distribués, et à chercher les groupes de pixels dont les propriétés sont suffisamment improbables sous ces hypothèses. Notre objectif est de montrer que lorsqu'un raisonnement totalement analytique n'est pas envisageable, une alternative peut être d'apprendre empiriquement, à base d'images de bruit blanc, les propriétés statistiques des groupes de pixels accidentels. Autrement dit, nous chercherons à apprendre ce que le hasard peut générer comme déviations. L'objectif qui reste inchangé est l'ε-fiabilité des algorithmes obtenus. Cette propriété permet de garantir la robustesse des détections, quels que soient les assouplissements apportés au cadre purement analytique.

Deux applications illustrent la démarche. Dans la section 2.2, nous chercherons à détecter des segments significatifs en se basant notamment sur un critère de longueur. Une heuristique dirigée par les données sera utilisée pour extraire les segments candidats. Cette heuristique est optimale, dans le sens où elle ne parcourt que les meilleurs candidats, mais le nombre total de candidats implicitement analysés n'est pas calculable analytiquement. Nous montrerons alors que la distribution des longueurs sous l'hypothèse *a contrario* peut être estimée de façon fiable à partir de simulations dans des images de bruit blanc.

Dans la section 2.3, le problème du seuillage automatique d'un algorithme de segmentation d'une image en régions sera abordé. Dans ce cas, une première difficulté viendra de la combinaison de plusieurs variables discriminantes de nature différente. Une deuxième difficulté viendra de l'utilisation d'heuristiques d'exploration (par exemple de regroupement itératif de régions)

dirigées par les données mais largement sous-optimales (de nombreuses régions significatives peuvent être ignorées). Nous montrerons alors que les seuils de décision peuvent également être déduits automatiquement à partir d'images de bruit blanc.

2.2 Détection de segments significatifs

2.2.1 Introduction

Les structures rectilignes ou alignements ont fait l'objet d'un intérêt constant en vision par ordinateur. Elles correspondent généralement à des bords d'objets et sont présentes dans la plupart des images, en particulier en environnement urbain (bâtiments, objets manufacturés). Elles ont donc un intérêt applicatif pour des tâches de plus haut niveau comme la reconstruction stéréo [JZ88, TK95], l'analyse d'images satellitaires [Jak07], ou encore la reconnaissance d'objets [FFJS08].

Il y a deux difficultés principales dans la détection de segments : la définition des structures candidates, et le choix d'un critère de décision qui permette de discriminer les segments correspondants à un phénomène physique dans la scène des alignements accidentels dus au bruit où aux zones très texturées. Parmi les algorithmes standards, on trouve les approches à base de transformée de Hough [IK88] et les méthodes à base de chainage de morceaux de contours [FDM$^+$92, Ete92]. Ces travaux ont comme principal défaut la présence d'un grand nombre de fausses alarmes dues à l'utilisation de critères de décision ad-hoc. Ce constat a motivé l'utilisation d'un raisonnement *a contrario* dans [DMM00b], qui s'est avéré capable de filtrer efficacement les fausses alarmes, mais tendait systématiquement à regrouper des segments proches qui correspondaient cependant à des objets disjoints. Ce défaut a été corrigé avec l'algorithme de découpage de [Jak07]. Dans ces deux travaux, les structures candidates à analyser sont très nombreuses et coûteuses à extraire, les temps de calcul obtenus sont donc très élevés. Ceci a conduits [GJMR08] à proposer de combiner l'heuristique d'extraction de structures candidates de [BHR86], très rapide, et l'approche *a contrario* pour l'étape de décision.

Poursuivant la démarche *a contrario* ainsi initiée, nous étudions ici une nouvelle définition des structures candidates et de nouvelles variables discriminantes. Initialement motivés par l'implantabilité d'un algorithme de détection de segments *a contrario* sur des architectures massivement parallèles à grain fin, de type rétine artificielle (voir annexe A), ces nouveaux choix ont ici pour objectif d'illustrer comment un apprentissage à partir d'images de bruit blanc peut assouplir le cadre *a contrario* et permettre de proposer un algorithme ε-fiable en présence d'une heuristique dirigée par les données.

Les segments candidats proposés correspondent aux suites les plus longues de pixels connectés ayant des directions locales proches. Nous commençons par expérimenter, à titre de comparaison, deux variables discriminantes basées sur le contraste et statistiquement indépendantes de la procédure d'extraction des candidats. Elles sont donc compatibles avec un raisonnement *a contrario* totalement analytique. Nous étudions ensuite le pouvoir discriminant d'une variable

basée sur la longueur des segments candidats, qui n'est cette fois pas indépendante de la procédure d'extraction. Nous montrerons alors que sa distribution sous l'hypothèse *a contrario* peut être "apprise" en mesurant les longueurs des segments observés dans des images de bruit blanc. Cet apprentissage *a contrario* sera finalement validé expérimentalement.

2.2.2 Définition de la notion de segment

Nous définissons tout d'abord les structures candidates susceptibles d'être des segments. Nous ne considérons ici que des images en niveau de gris, et le modèle de communication des architectures massivement parallèles considérées nous conduit à proposer la définition suivante pour la notion de segment, qui permet une extraction reposant uniquement sur des interactions locales entre pixels voisins.

Définition 3. *Un segment dans un cône de direction C est un ensemble de pixels connectés d'une épaisseur d'un seul pixel dans la direction orthogonale à la direction médiane de C, tel que :*

- *chaque pixel a une direction locale orthogonale au gradient dans le cône C ;*
- *pour chaque pixel qui n'est pas une extrémité (ayant deux voisins), la direction du vecteur formé par ses deux voisins est également dans C.*

La figure 2.1 donne des exemples de segments. La garantie d'une épaisseur de un pixel est assurée par une étape de localisation décrite dans la section 2.2.3. Les directions de gradient sont calculées par un opérateur de Sobel, et huit cônes de direction sont analysés, correspondant aux angles possibles dans un voisinage discret 5×5. Les cônes sont donc partiellement superposés, et la tolérance est de $22.5°$ à partir de l'axe central de chaque cône. Outre l'aspect pratique de ce choix, cette tolérance correspond aux valeurs optimales (quoiqu'empiriques) obtenues par d'autres travaux [GJMR08, BHR86].

2.2.3 Extraction des segments candidats

Pour extraire les segments candidats de l'image, nous proposons l'heuristique suivante, qui cherche à sélectionner les candidats les plus significatifs en extrayant les suites les plus longues de pixels connectés correspondant à la définition 3. Cet algorithme d'extraction est divisé en deux étapes :

1. Extraire huit images binaires correspondant à chacun des cônes de direction. Dans chacune de ces images, un pixel est allumé (de valeur 1) si le point correspondant dans l'image originale a une direction locale de gradient incluse dans le cône.
2. Pour chacune de ces images binaires, l'épaisseur des segments est réduite à un pixel dans la direction orthogonale à la direction médiane du cône en ne conservant que les pixels situés sur un maximum local de module du gradient.

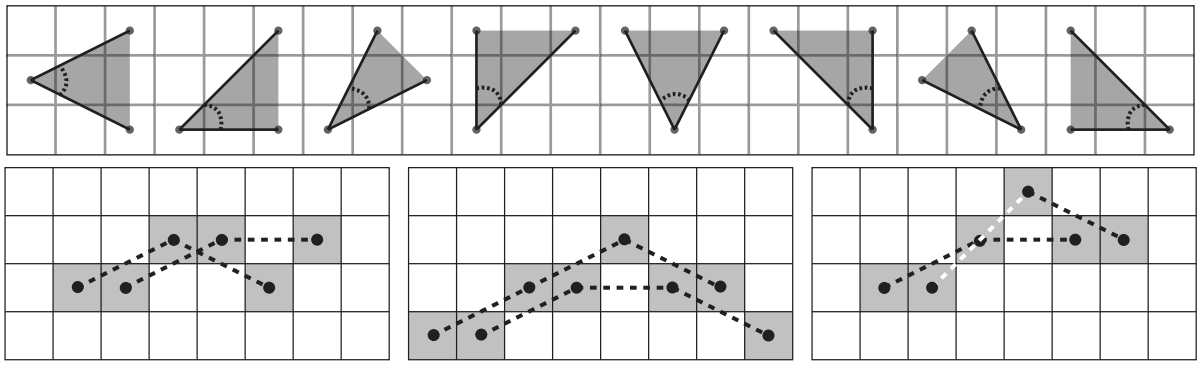


FIG. 2.1 – En haut : les huit cônes de direction. En bas : illustration de la définition 3 pour le premier cône, horizontal. Les pixels foncés doivent avoir une direction locale incluse dans le cône. Parmi les trois ensembles de pixels, seuls ceux de gauche et du milieu sont des segments valides d'après la définition 3, puisque toutes les directions créées par les voisins (en pointillés) sont dans le cône. Celui de droite ne satisfait pas les conditions puisque le voisinage en pointillés blancs a un angle trop élevé et sort de la tolérance du cône.

La figure 2.2 donne un exemple d'images binaires de direction. Les huit images sont ensuite traitées indépendamment.

La localisation des segments sur les crêtes du gradient (étape 2) est effectuée de manière itérative, en transférant à chaque itération les pixels qui ne sont pas des maxima vers leur voisin le plus fort, dans la direction orthogonale au cône de direction. À chaque transfert, l'intensité de gradient du pixel le plus faible est transférée à celle de son voisin le plus proche. Les intensités de gradient sont ainsi cumulées sur toute l'épaisseur du segment, donnant une importance égale à un segment peu contrasté mais épais et à un segment fin mais très contrasté. Cet algorithme est illustré sur la figure 2.3.

Nous disposons maintenant d'un ensemble de segments candidats. Il reste à discriminer ceux qui correspondent à une structure de la scène de ceux qui sont le résultat du hasard.

2.2.4 Modèle *a contrario* pour les segments

Pour appliquer un raisonnement *a contrario*, il faut tout d'abord définir un modèle permettant de calculer les distributions des variables discriminantes sous l'hypothèse où elles sont le résultat du hasard. Nous choisissons ici le modèle classique d'indépendance entre pixels.

Définition 4. *Modèle a contrario pour H_0.*
Sous l'hypothèse de hasard H_0, les pixels d'une image sont identiquement distribués et indépendants si leur distance est supérieure ou égale à 2.

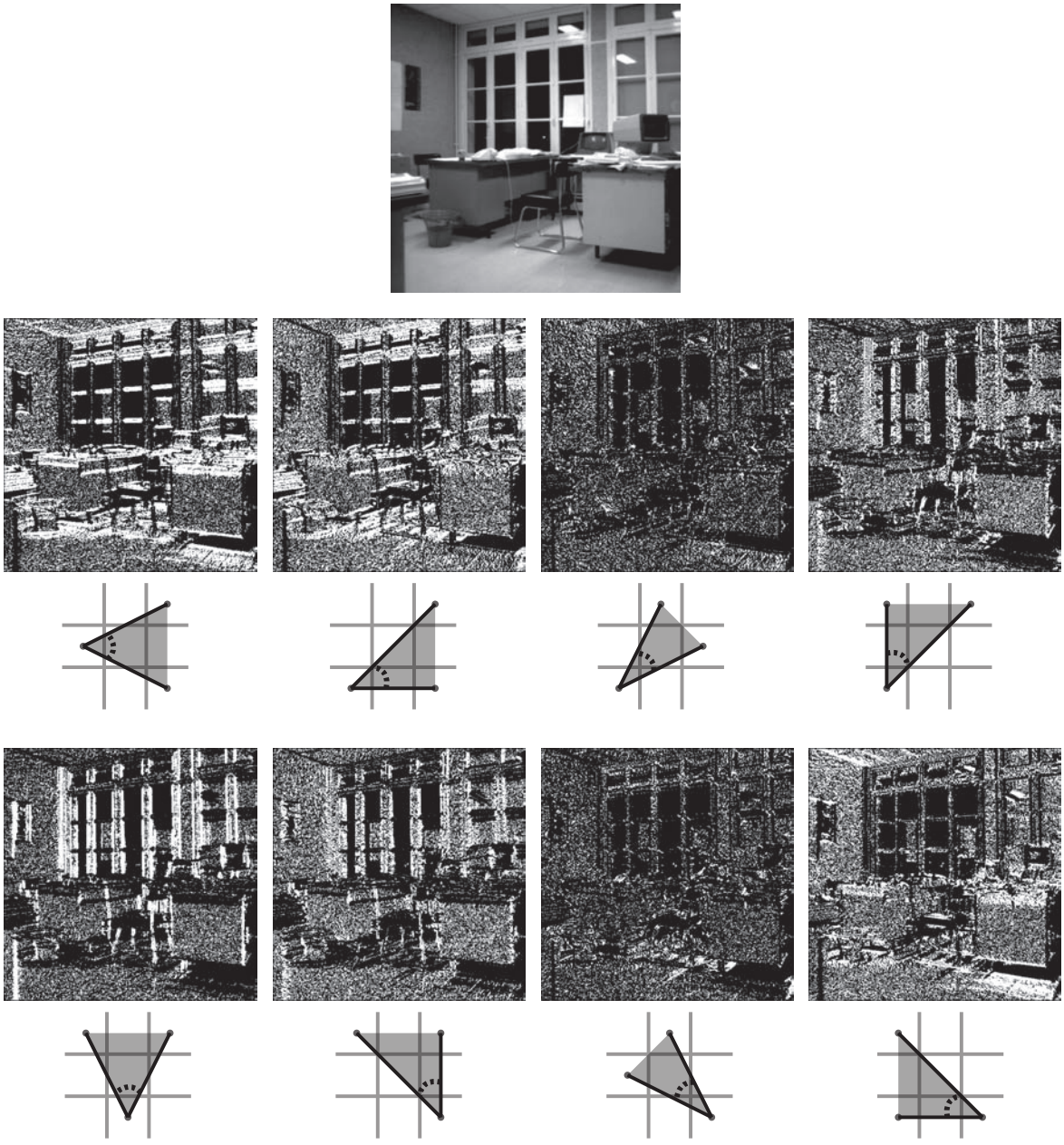


FIG. 2.2 – Images de direction binaires obtenues pour l'image "bureau". Chaque image correspond à un cône de direction, et un pixel blanc de valeur 1 signifie que l'orientation du gradient de l'image en ce point est incluse dans le cône.

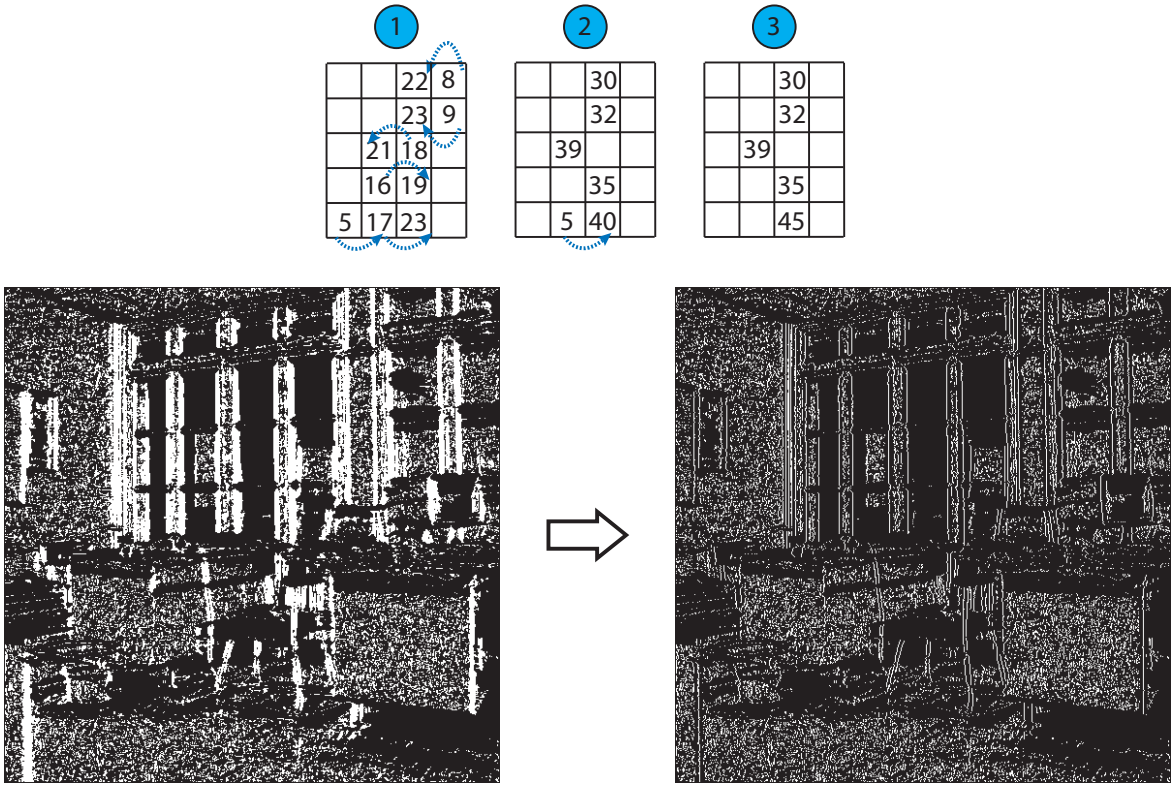


FIG. 2.3 – Localisation des segments sur les crêtes de gradient. À chaque itération, les pixels qui ne sont pas des maxima sont regroupés vers leur voisin le plus fort, dans la direction strictement orthogonale à la direction médiane du cône de direction. Les intensités de gradient sont ainsi cumulées sur toute l'épaisseur du segment jusqu'aux maxima locaux. Les images du bas correspondent à la localisation de l'image de direction verticale. Les autres images de direction sont également localisées avec le même algorithme.

Le modèle *a contrario* considère donc que les pixels sont arrangés spatialement de façon purement aléatoire. La distance minimale correspond à la distance de Nyquist : deux pixels d'une image bien échantillonnée ne sont pas indépendants si leur distance est inférieure à 2 [Pra01].

En raisonnant *a contrario*, les segments significatifs à détecter sont ceux pour lesquels une certaine variable discriminante est statistiquement trop élevée pour être le résultat du hasard, selon le modèle *a contrario* proposé. Nous étudions maintenant trois variables discriminantes : deux basées sur le contraste des segments, qui sont compatibles avec un raisonnement *a contrario* purement analytique, et une basée sur la longueur des segments candidats.

2.2.5 Segments significatifs par leur contraste minimal

Une première approche, inspirée de [DMM01], consiste à analyser l'intensité de gradient minimale λ le long du segment. Plus λ est élevée, plus il y a de chance que le segment coïncide avec un bord d'objet dans la scène. La distribution de λ sous H_0 dépend de la longueur du segment L et de la distribution G des intensités de gradient sur l'image. Comme [DMM01], nous estimons G empiriquement à partir de l'image. Cette variable n'est pas discriminante par elle-même et nous la considérons comme une variable conditionnante. L est en revanche discriminante et nous l'utiliserons en tant que telle dans la section 2.2.8. Mais suivant à nouveau [DMM01], nous la considérons ici comme une variable conditionnante, pour se focaliser uniquement sur le pouvoir discriminant du contraste minimal.

Notre heuristique d'extraction choisit les segments candidats indépendamment de leurs intensités de gradient. Donc, la distribution de λ sous l'hypothèse *a contrario* correspond à la probabilité que tous les pixels du segment aient un contraste supérieur à λ, dans le cas où les pixels éloignés d'au moins deux pixels sont indépendants et identiquement distribués. Conditionnellement à L et G, cela donne la probabilité de fausse alarme suivante pour un segment candidat S :

$$PFA_\lambda(S) = P_{H_0}(\lambda \geq \lambda(S) \mid L = L(S), G) = G[\lambda(S)]^{\frac{L(S)}{2}}$$

en notant $G[x]$ la fonction de répartition complémentaire $P(X \geq x)$ de la distribution du gradient sur l'image. Le facteur $\frac{1}{2}$ appliqué à $L(S)$ découle de la distance de Nyquist.

En utilisant la proposition 1 du chapitre 1, on obtient un algorithme ε-fiable en détectant les segments S tels que :

$$PFA_\lambda(S) < \frac{\varepsilon}{N_s}$$

avec N_s le nombre de segments candidats dans l'image en cours d'analyse. Un raisonnement purement analytique est donc possible avec cette variable discriminante.

L'inconvénient majeur de ce critère est sa faible résistance au bruit : il suffit qu'un seul pixel sur le segment ait un contraste faible pour que l'ensemble du segment devienne beaucoup moins significatif. Ceci se constate sur les figures 2.8 (bas) et 2.10.

2.2.6 Segments significatifs par leur contraste moyen

Pour pallier ce problème, nous proposons de prendre la moyenne de contraste μ comme variable discriminante, car elle prend mieux en compte l'ensemble du segment. Tout comme le minimum de contraste λ, la la distribution de μ dépend de la longueur du segment L et de la distribution globale de gradient G. La probabilité de fausse alarme pour un segment S devient alors :

$$\mathit{PFA}_\mu(S) = P_{H_0}(\mu \geq \mu(S) \mid L = L(S), G)$$

Il est toutefois plus difficile d'estimer la distribution de μ sous H_0 de façon précise. En effet, si pour les longs segments, une approximation par la loi normale est possible, elle s'avère trop optimiste pour les petits segments. Nous avons en effet constaté expérimentalement qu'elle tend à sous-estimer la probabilité réelle d'obtenir des valeurs faibles de μ par hasard, et n'assure donc plus l'ε-fiabilité de la détection.

Comme les petits segments représentent la majorité des candidats, il est nécessaire de calculer une probabilité plus précise. La loi de μ peut être calculée de façon exacte pour chaque longueur de segment par autoconvolution itérée de G. Soient $\{X_1, X_2, \ldots, X_n\}$ les intensités de gradient mesurées un pixel sur deux le long d'un segment candidat S de longueur $L(S) = 2 \times n$. Sous l'hypothèse H_0, les variables X_i sont indépendantes et identiquement distribuées selon G. La distribution F_n^G de la somme $M(S) = \sum_{i=1}^{n} X_i$ peut alors être déduite de G récursivement par la propriété suivante [GS97] :

$$F_n^G = F_{n-1}^G \star G$$

avec $f \star g$ le produit de convolution entre deux distributions f et g. Le cas initial $n = 1$ est donné par $F_1^G = G$. On en déduit la distribution exacte de μ sous H_0 pour un segment candidat S :

$$P_{H_0}(\mu \geq \mu(S) \mid L = L(S), G) = F_{\frac{L(S)}{2}}^G \left(\mu(S) \times \frac{L(S)}{2} \right)$$

Pour les grands segments, cette distribution peut être approchée par la loi normale :

$$P_{H_0}(\mu \geq \mu(S) \mid L = L(S), G) = \mathcal{N}_{\geq} \left(\mu(S), \mu_G, \frac{\sigma_G}{\sqrt{\frac{L(S)}{2}}} \right)$$

avec μ_G et σ_G respectivement la moyenne et l'écart-type des intensités de gradient sur l'image, et $\mathcal{N}_{\geq}$ la fonction de répartition complémentaire de la loi normale.

Nos expériences montrent que l'approximation normale est suffisamment précise à partir de $n = 20$. Nous pouvons finalement déduire ici aussi un algorithme de détection ε-fiable grâce à la proposition 1 en sélectionnant les segments S tels que :

$$\mathit{PFA}_\mu(S) < \frac{\varepsilon}{N_s}$$

avec N_s le nombre de segments candidats dans l'image en cours d'analyse.

Si cette variable est plus résistante au bruit, elle est en revanche moins sensible que le minimum de contraste λ pour les petits segments très contrastés . Quand l'image est de bonne qualité, il est en effet fréquent que tous les pixels alignés sur les bords des objets aient un contraste élevé, une information riche que le critère λ exploite plus fortement que la moyenne μ. C'est pourquoi nous étudions maintenant leur combinaison.

2.2.7 Combinaison du minimum et de la moyenne de contraste

Les deux variables reposent sur les intensités de gradient du segment et ne sont pas indépendantes, il est donc difficile de calculer leur probabilité jointe. Étant donné leur rôle complémentaire, nous proposons de simplement combiner les deux détecteurs. Pour chaque segment, les deux critères sont donc testés, et si au moins l'un d'eux est significatif le segment est détecté. Cette méthode nécessite d'ajuster le calcul des seuils pour continuer à garantir l'ε-fiabilité. Soient $\#FA_\lambda$ le nombre de fausses alarmes produites par le critère λ sur une image, et $\#FA_\mu$ le nombre de fausses alarmes produites par le critère μ. Le nombre total de fausses alarmes $\#FA_{\mu+\lambda}$ produites par la combinaison des deux critères est nécessairement inférieur à $\#FA_\lambda + \#FA_\mu$. On en déduit :

$$\begin{aligned} \mathbb{E}\left(\#FA_{\lambda+\mu}\right) &\leq \mathbb{E}\left(\#FA_\lambda + \#FA_\mu\right) \\ &\leq \mathbb{E}\left(\#FA_\lambda\right) + \mathbb{E}\left(\#FA_\mu\right) \quad \text{par linéarité de l'espérance} \\ &\leq 2 \times \varepsilon \end{aligned}$$

Il suffit donc de détecter les segments S tels que $PFA_\lambda(S) < \frac{\varepsilon}{2\times N_s}$ ou $PFA_\mu(S) < \frac{\varepsilon}{2\times N_s}$ pour continuer à garantir l'ε-fiabilité. Cette méthode reste donc robuste en terme de fausses alarmes. Elle est en revanche nécessairement moins sensible qu'un critère basé sur la probabilité jointe des deux variables, car ici un segment doit avoir au moins une des deux variables qui soit significative par elle-même. En pratique, cette approche très simple s'avère cependant capable de considérablement augmenter le nombre de segments détectés, car les deux variables sont relativement complémentaires.

2.2.8 Segments significatifs par leur longueur

Les deux variables précédentes sont capables de détecter la plupart des segments présents dans une image, et permettent un raisonnement *a contrario* totalement analytique. Les spécificités des architectures massivement parallèles nous ont cependant conduits à étudier le pouvoir discriminant de la longueur L, plus simple à calculer. La longueur de chaque segment correspond au nombre de pixels blancs connectés qui le constituent dans son image binaire de direction localisée. La distribution de la variable L sous l'hypothèse H_0 est cette fois beaucoup plus difficile à déterminer de façon analytique, car l'heuristique de sélection des candidats choisit précisément les séries de pixels blancs connectés les plus longues dans l'image. La longueur d'un segment candidat ne peut donc pas être considérée indépendante de l'heuristique. Une première solution, celle de [GJMR08], consiste à se placer dans le premier cas extrême de la section

1.4.4 et de considérer que l'heuristique est optimale, dans le sens où elle parcourra les candidats les plus significatifs. Cette hypothèse est valable ici, puisque les segments les plus significatifs par leur longueur sont nécessairement les plus longs. Cette approche nécessite cependant d'estimer le nombre total d'emplacements possibles pour un segment dans une image binaire. Étant donné la définition 3 de la notion de segment, basée sur des propriétés de connexité locale, ce nombre n'est pas calculable analytiquement, il s'agit d'un problème difficile de combinatoire énumérative.

Nous proposons de raisonner différemment. Le problème est : étant donné une image binaire de direction, quelle est la probabilité qu'un segment extrait par notre heuristique ait au moins L pixels blancs si les pixels sont indépendants et identiquement distribués ? La distribution de L dépend de la densité de pixels blancs p_b de l'image binaire : plus elle est élevée, plus il y a de chances d'observer par hasard des segments longs. Il est très facile de générer des images binaires suivant le modèle *a contrario* avec différentes densités de pixels blancs, aussi nous proposons d'estimer la distribution *a contrario* de L en fonction de p_b empiriquement. Pour cela, nous générons Q images binaires de bruit blanc pour chacune des valeurs de p_b discrétisées entre 0 et 1. Les longueurs des segments sont alors mesurées dans chaque image pour estimer la distribution conditionnelle empirique $P_{H_0}(L|p_b)$. En pratique, un pas de discrétisation de 0.01 pour p_b et un nombre d'images $Q = 1000$ s'avèrent suffisants pour obtenir une estimation fiable. La figure 2.4 montre les distributions obtenues pour quelques densités. Toutefois, les segments très longs ont une probabilité d'apparition trop faible dans des images de bruit blanc pour que leur *PFA* puisse être estimée empiriquement. Pour estimer la probabilité associée à de telles longueurs, il est nécessaire d'extrapoler les distributions empiriques obtenues. Nous avons pour cela recours à la théorie des grandes déviations, qui permet de modéliser la queue d'une distribution à partir d'un échantillon. La procédure est détaillée dans l'annexe C.

Les distributions obtenues sont très régulières, les estimations peuvent donc être considérées fiables. Nous pouvons maintenant en déduire un critère de décision de la même manière que pour les variables précédentes : un segment S de longueur L_S extrait d'une image binaire de densité p_b sera déclaré significatif par sa longueur si :

$$\mathit{PFA}_L(S) = N_s \times P_{H_0}(L \geq L(S) \mid p_b) < \frac{\varepsilon}{N_s}$$

avec N_s le nombre de segments candidats dans l'image en cours d'analyse. Notons que les distributions $P_{H_0}(L \geq L(S) \mid p_b)$ estimées sont très proches de distributions exponentielles. Ceci induit une dépendance seulement logarithmique des seuils de longueur minimaux en fonction du nombre moyen de fausses alarmes tolérées ε, et confirme que le choix de ε n'est généralement pas très sensible dans les approches *a contrario* [DMM08].

Cette détection basée uniquement sur la longueur ne produit pas de fausses alarmes, mais ne détecte que des segments longs. En effet, par définition, elle ne tient pas compte du contraste, et des segments assez longs peuvent apparaître par hasard dans du bruit si la densité est relativement élevée, les seuils assurant l'ε-fiabilité sont donc relativement élevés. Ainsi, la figure 2.4 indique que, pour une densité de 0.15 et pour un nombre de candidats N_s de l'ordre de 10000, deux valeurs classiques pour une image, un segment devra avoir au moins 7 pixels indépendants (et donc une longueur supérieure à 14) pour que $\log \mathit{PFA}_L$ soit inférieure à $\log(\frac{\varepsilon}{N_s}) \simeq -9.2$.

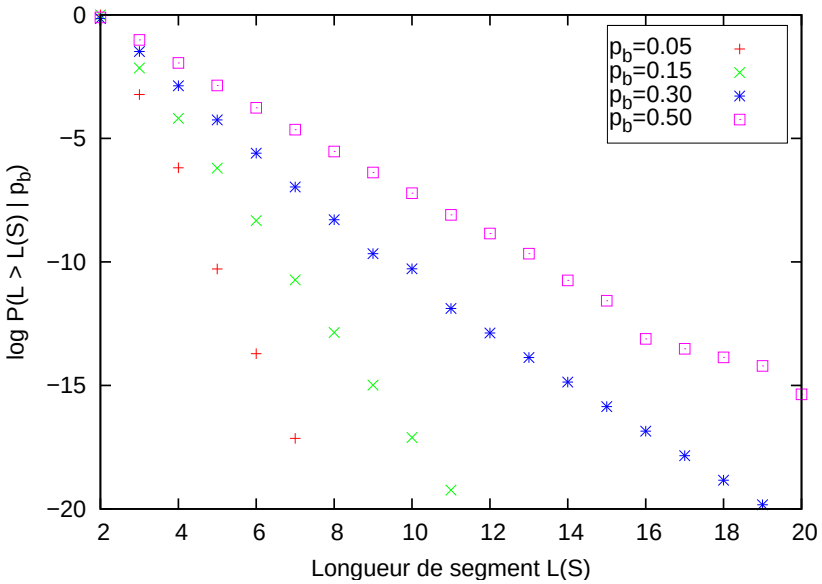


FIG. 2.4 – Fonction de répartition complémentaire empirique $P(L > L(S) \mid p_b)$ des longueurs de segment obtenues dans une image binaire pour différentes densités de pixels blancs, si les pixels sont indépendants et identiquement distribués. L'échelle est logarithmique. Ces courbes empiriques sont obtenues à partir de 1000 images générées pour chaque densité. Il s'agit visiblement de distributions exponentielles très stables, et donc estimables et extrapolables de façon fiable.

La sensibilité de l'algorithme peut être augmentée très simplement en appliquant la détection pour différents niveaux de seuils globaux sur les intensités de gradient. La procédure proposée est donnée par l'algorithme 3. L'idée est d'appliquer une série de seuils δ ne conservant à chaque itération que les δ pour cent des pixels les plus contrastés. Le nombre de pourcentages et donc de seuils testés est libre, mais en pratique des pourcentages allant de 10% à 100% par pas de 10% s'avèrent suffisamment précis. Il faut maintenant prendre en compte le nombre de seuils testés N_δ dans le calcul du *NFA* pour continuer à garantir l'ε-fiabilité. On sait que pour chaque seuil l'espérance du nombre de fausses alarmes est inférieur à ε, par linéarité l'espérance du nombre de fausses alarmes total est donc inférieur à $N_\delta \times \varepsilon$. Pour assurer l'$\varepsilon$-fiabilité, il faut donc ajuster le calcul des seuils pour détecter les segments S tels que :

$$PFA(S) = P_{H_0}(L \geq L(S) \mid p_b) < \frac{\varepsilon}{N_\delta \times N_s}$$

avec N_s le nombre de segments candidats au cours de l'itération où S a été détecté. Le comportement de l'algorithme est illustré dans la figure 2.5.

Algorithme 3 : À chaque itération, les pixels les moins contrastés sont supprimés des images de direction, faisant diminuer la densité p_b et donc le seuil de longueur minimal pour être significatif. Les petits segments contrastés seront donc détectés quand le seuillage sera suffisamment élevé.

pour chaque *pourcentage* δ **faire**
  pour chaque *image de direction localisée* **faire**
    Supprimer les pixels blancs de façon à ne conserver que les $\delta\%$ de pixels les plus contrastés ;
    Détecter les segments significatifs ;
  fin
fin

2.2.9 Validation expérimentale des seuils de détection

Pour s'assurer que ces algorithmes de détection ne produisent pas de fausses alarmes au sens du modèle *a contrario*, nous avons mesuré expérimentalement l'espérance et l'écart-type du nombre de segments détectés par chaque algorithme sur $Q = 1000$ images de bruit blanc uniformes. ε a été fixé à 1, ce qui signifie que la moyenne du nombre de fausses alarmes par image devrait être inférieure à 1. Le tableau 2.1 résume les valeurs obtenues pour chaque critère et en déduit le degré de précision des estimations, ainsi que le degré de confiance dans l'ε-fiabilité des algorithmes, calculé par la méthode standard suivante. Soit $\mathbb{E}_{\#\text{FA}}$ l'espérance réelle du nombre de fausses alarmes pour un algorithme, M l'espérance empirique mesurée, et S l'écart-type mesuré. On sait que $Y = \frac{\sqrt{Q}(M - \mathbb{E}_{\#\text{FA}})}{S}$ suit une loi de Student à $Q - 1$ degrés de liberté [Sap90]. On en déduit le degré de confiance dans l'ε-fiabilité dans chaque cas :

$$P\left(\mathbb{E}_{\#\text{FA}} < 1\right) = P\left(Y < \sqrt{Q}\frac{M-1}{S}\right)$$

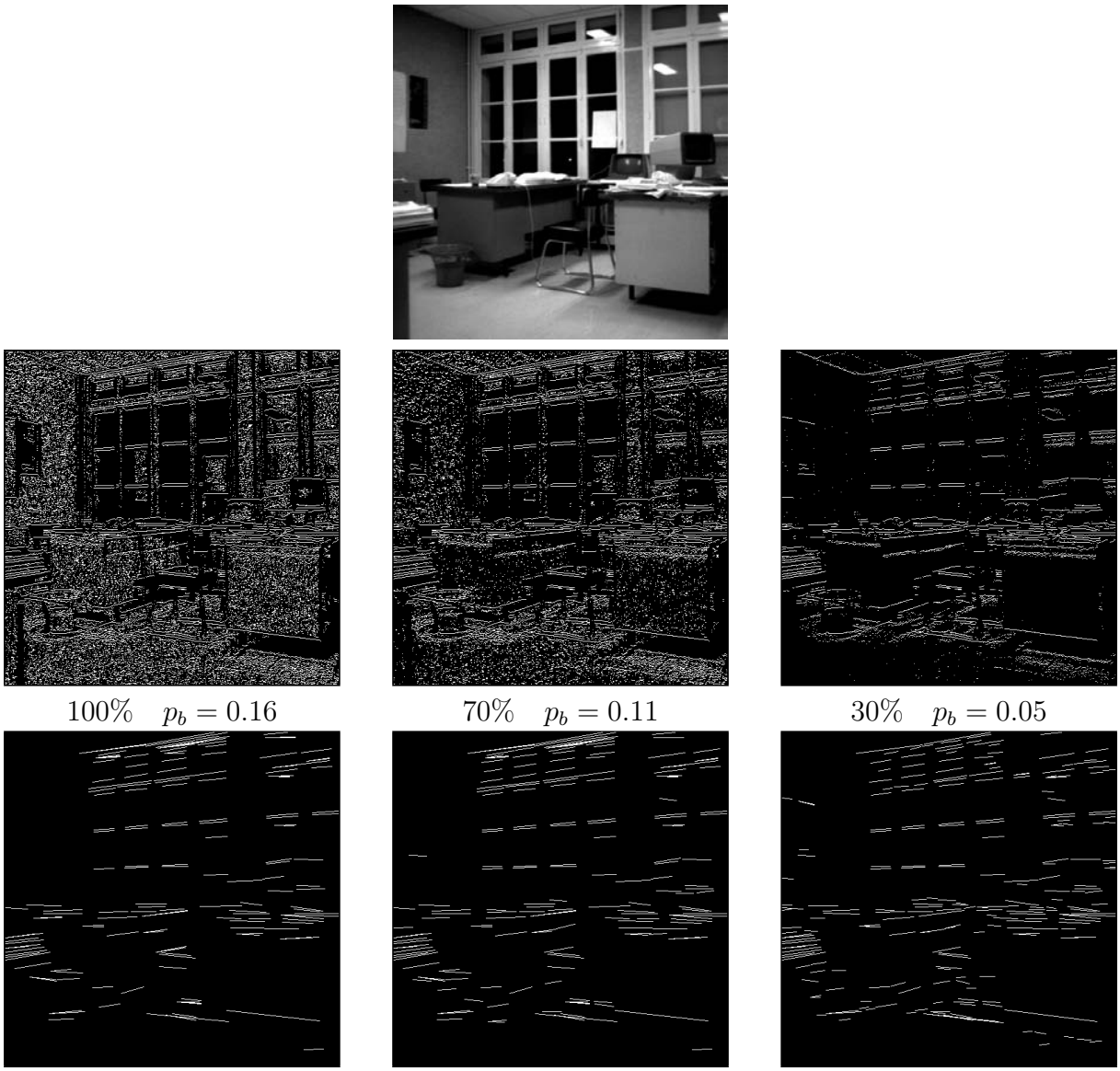


FIG. 2.5 – Images de direction horizontale localisées seuillées à différents pourcentages pour l'image "bureau". Quand tous les pixels sont conservés (à gauche), seuls les longs segments sont détectés. Plus le seuil est élevé, et plus les petits segments contrastés sont détectés. Ceci se remarque particulièrement sur les bords d'affiches en haut à gauche. Les segments détectés sont donc complémentaires et seront cumulés pour obtenir la liste finale des segments.

Critère	Espérance	Écart-type	Confiance
Minimum	0.023	0.15	$\simeq 1$
Moyenne	0.075	0.27	$\simeq 1$
Minimum+Moyenne	0.097	0.31	$\simeq 1$
Longueur	0.004	0.06	$\simeq 1$

TAB. 2.1 – Moments du nombre de segments considérés significatifs dans des images de bruit blanc uniforme bien échantillonnées. $Q = 1000$ images ont été générées pour obtenir ces valeurs. On constate que les espérances du nombre de fausses alarmes sont inférieures à $\varepsilon = 1$ dans tous les cas, avec une grande marge, ce qui garantit la robustesse des détections.

Les espérances empiriques observées sont bien en dessous de $\varepsilon = 1$ et sont plutôt de l'ordre de 10^{-1} à 10^{-2}. Comme le remarque l'étude détaillée de [GJ08], ceci est en partie dû à l'utilisation de variables discrètes. De plus, si deux pixels distants de moins de deux pixels ne sont pas indépendants, il reste malgré tout pessimiste de considérer qu'ils sont complètement dépendants et donc d'en éliminer un sur deux dans les calculs de *PFA*. Tout cela n'est cependant pas très critique étant donné la dépendance seulement logarithmique des détecteurs par rapport à ε.

2.2.10 Résultats

Les figures 2.6, 2.7, 2.8, 2.9 et 2.10 montrent les résultats obtenus sur des images usuelles et sur des images mises à disposition par [GJMR08]. Les différents critères sont testés, et le résultat obtenu par l'algorithme de [GJMR08] (LSD) est également donné à titre de comparaison. De par la définition 3, les segments obtenus sont souvent de forme relativement complexe. Pour donner une représentation compacte et plus claire, les segments sont représentés par des morceaux de droites ayant comme extrémités celles du segment correspondant. Ainsi, un segment est finalement défini par ses points de départ et d'arrivée.

Toutes les méthodes sont basées sur un raisonnement *a contrario* et sans paramètres libres. À chaque fois, aussi bien pour les détecteurs purement analytiques que pour celui basé sur un apprentissage dans des images de bruit blanc, les segments détectés correspondent toujours à des structures particulières de l'image et l'on n'observe pas de fausses alarmes dans les régions texturées ou dans des images de bruit. Les différences se situent au niveau des seuils de sensibilité.

Le critère de minimum λ détecte beaucoup de segments quand l'image est peu bruitée, alors que le critère de moyenne μ est moins sensible de façon générale mais plus robuste au bruit. Leur combinaison permet de détecter systématiquement la quasi-totalité des segments présents dans l'image. Le critère de longueur $L(S)$ couplé avec un seuillage multiple des contrastes s'avère également très sensible et capable de détecter la plupart des segments, même très peu contrastés.

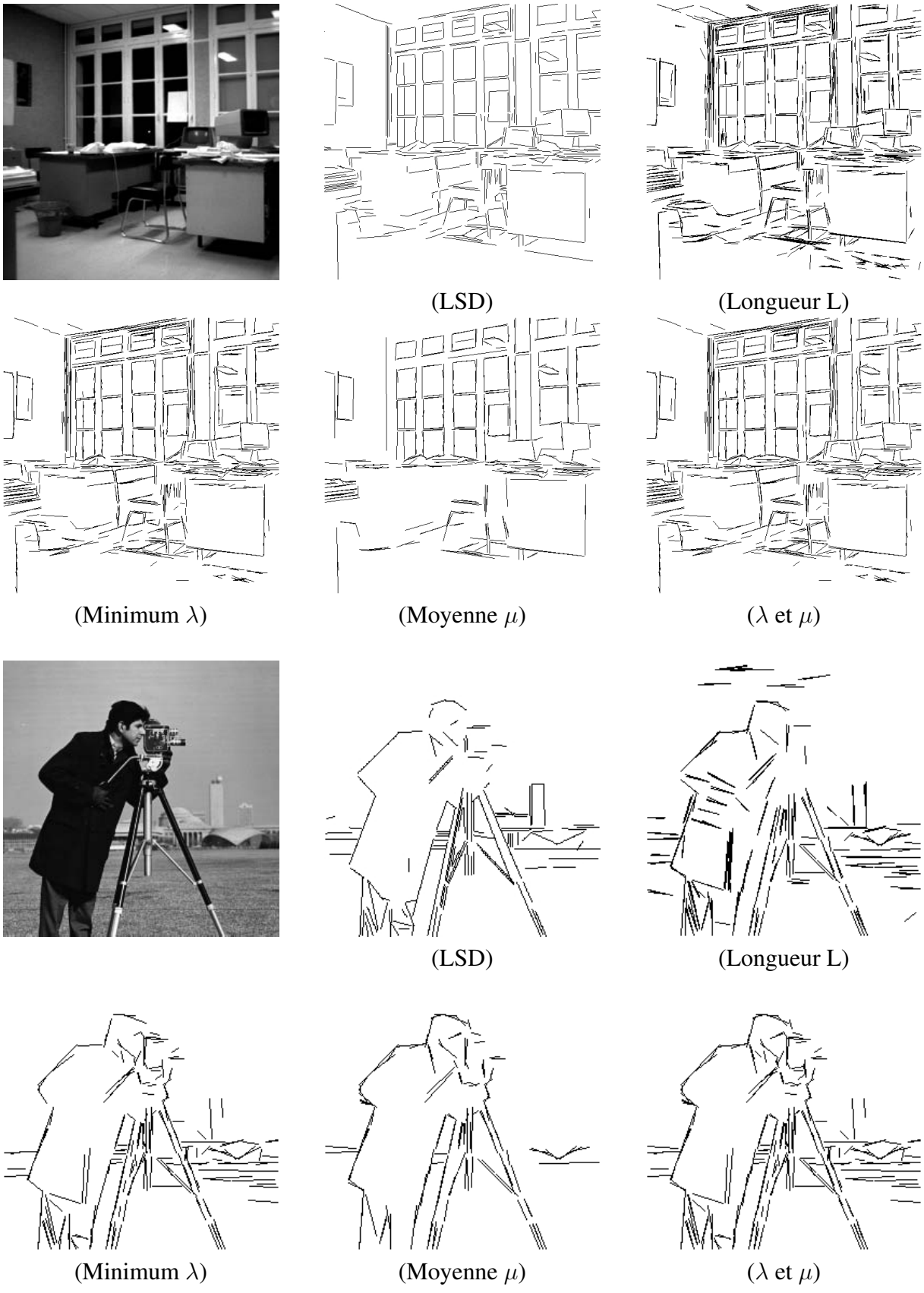


FIG. 2.6 – Détection de segments. Voir section 2.2.10.

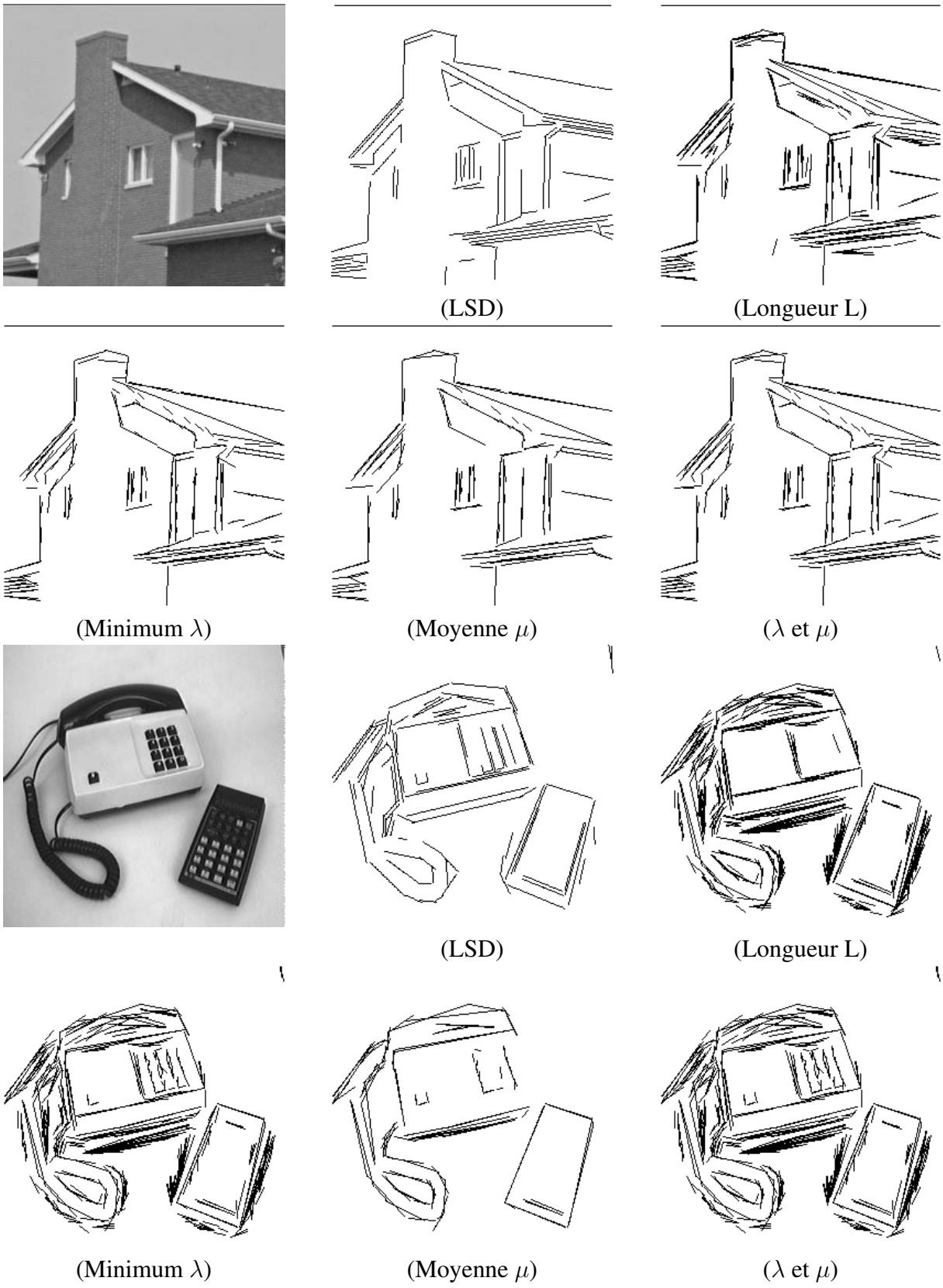


FIG. 2.7 – Détection de segments. Voir section 2.2.10.

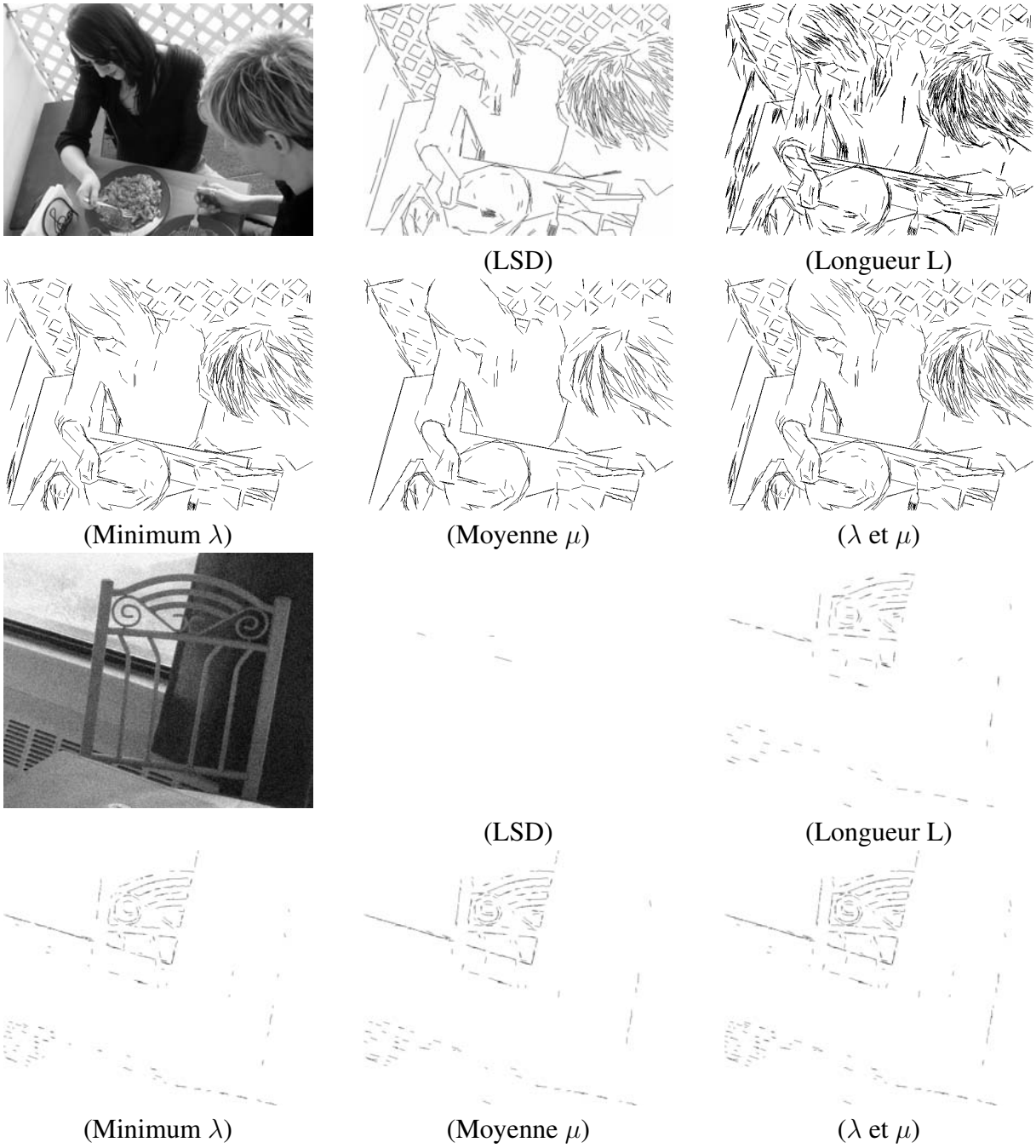


FIG. 2.8 – Détection de segments. Voir section 2.2.10.

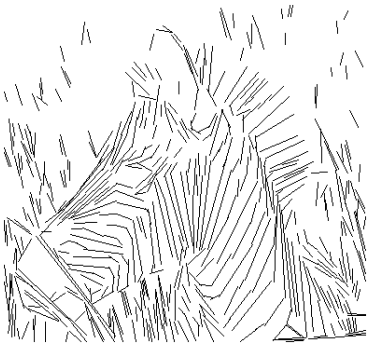
(LSD)

(Longueur L)

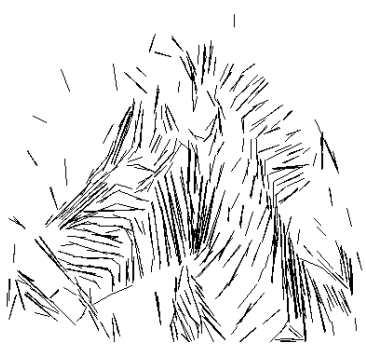
(Minimum λ)

(Moyenne μ)

(λ et μ)



(LSD)



(Longueur L)

(Minimum λ)

(Moyenne μ)

(λ et μ)

FIG. 2.9 – Détection de segments. Voir section 2.2.10.

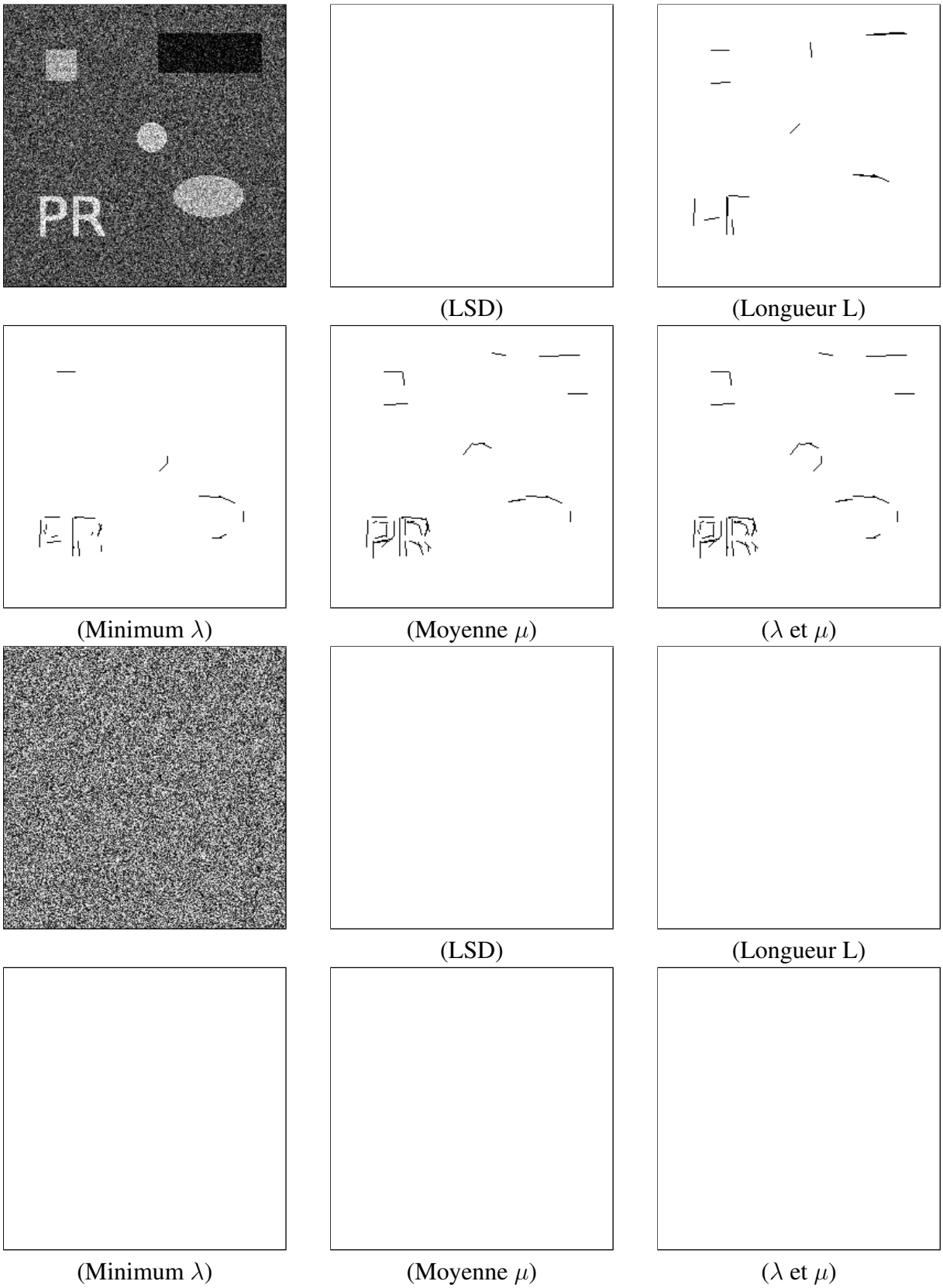


FIG. 2.10 – Détection de segments sur des images synthétiques de bruit. Voir section 2.2.10.

Globalement, notre méthode détecte plus de segments que la méthode LSD, en particulier sur les images bruitées (figure 2.8 en bas et figure 2.10). Ceci s'explique probablement par notre utilisation de l'information de contraste, alors que LSD s'appuie sur une variable discriminante purement géométrique. En contrepartie, nos segments sont plus morcelés, car leur extraction se base uniquement sur des interactions locales avec un seul pixel d'épaisseur, une fois la localisation effectuée. Ils sont également plus redondants puisque les cônes de direction se superposent. Ceci destinerait plutôt notre méthode à des applications qui ne nécessitent pas de traiter chaque segment individuellement, mais qui cherchent plutôt à extraire des tendances plus globales, comme par exemple les histogrammes de direction de segments utilisés dans l'approche de [FFJS08] pour la reconnaissance d'objets.

2.2.11 Discussion

Nous avons proposé plusieurs variantes d'un algorithme de détection de segments basé sur la méthodologie *a contrario*. Différentes variables discriminantes ont été étudiées, dont une basée sur la longueur qui n'est pas indépendante de la manière dont sont extraits les segments candidats. L'approche analytique traditionnelle étant alors inadaptée, nous avons montré qu'il était pertinent d'estimer la distribution de cette variable de façon empirique, en "apprenant" directement à partir d'images de bruit blanc la longueur des segments qui peuvent apparaître de façon accidentelle. Au final, la détection reste ε-fiable et se montre capable de détecter la quasi-totalité des segments d'une image.

2.3 Segmentation d' image en régions

2.3.1 Introduction

Pour montrer que le raisonnement *a contrario* peut être étendu à des applications encore plus difficiles à modéliser de façon purement analytique, nous considérons ici la segmentation d'une image en groupes de pixels connectés et homogènes, appelés régions. C'est un problème classique en vision par ordinateur, qui constitue souvent une brique importante pour des tâches de vision de plus haut niveau. Il correspond également à une forme de "groupement perceptuel" qui a été mis en évidence par les gestaltistes dans leurs études phénoménologiques de la vision humaine.

Il s'agit d'un problème difficile. Il est en effet très délicat de déterminer des critères d'homogénéité qui soient capables de s'adapter de façon cohérente à la grande variabilité des images naturelles. Pour cette raison, il est difficile d'obtenir un algorithme de segmentation qui soit "globalement" bon, c'est-à-dire dont les résultats soient satisfaisants sur l'ensemble des images. Même au sein d'une unique image il existe souvent des variations très importantes sur la nature des régions, qui impliquent des critères de décision différents. Ceci conduit le plus souvent à introduire des paramètres dans les différents algorithmes, qui seront ajustés manuellement pour chaque image ou bien appris à partir d'une base d'images. Dans [Sha08], il est même proposé d'apprendre, de façon statistique, quel algorithme de segmentation est le plus adapté pour chaque image. Cet apprentissage supervisé nécessite toutefois des images naturelles annotées, très difficiles à obtenir.

Aussi, en trois décennies de recherche, de nombreuses approches ont été proposées. Elles reposent toutes, de façon implicite ou explicite, sur un processus de décision qui doit décider, pour deux régions adjacentes, si elles doivent être considérées comme un "tout" homogène ou comme deux entités distinctes. Leurs différences résident principalement dans leurs hypothèses sous-jacentes sur l'image, dans les caractéristiques analysées, dans la façon de parcourir l'espace des segmentations possibles, et dans la manière de fixer les seuils de décision [ZY96].

Un premier groupe de méthodes modélise la segmentation d'image comme un problème de minimisation d'énergie globale [ZY96, TZ02, MS89, SM00]. Ces approches sont intéressantes car elles permettent de fixer un objectif quantitatif global à optimiser. En revanche, pour définir une fonction d'énergie, elles nécessitent des quantités *a priori* difficiles à estimer et dont la portée ne pourra pas être universelle. Par exemple dans [MS89] un paramètre est requis pour contrôler les influences respectives de la régularité de la segmentation et de l'attache aux données, et dans [TZ02] il est nécessaire d'estimer *a priori* également le nombre et la taille des régions. De plus, les énergies globales obtenues sont généralement des fonctions non convexes difficiles à optimiser, même si des progrès récents ont été obtenus [KH08, PBD06], notamment grâce aux approches à base de graphes.

L'autre groupe de méthodes le plus courant, sur lequel nous nous focalisons ici, repose sur des prédicats explicites, qui, pour deux régions adjacentes, décident si elles sont signifi-

cativement différentes ou si elles forment un tout homogène et doivent donc être regroupées. En pratique, une mesure de différence entre régions est introduite, généralement empirique [FH04, RP98] ou statistiquement fondée [ZY96, NN04, CM02, GB04]. Pour décider si deux régions adjacentes sont significativement différentes, il suffit alors d'introduire un seuil sur la mesure de différence. En se basant sur ce critère de décision, les régions sont ensuite itérativement regroupées et/ou divisées de façon à obtenir une segmentation finale stable où, idéalement, les régions adjacentes sont significativement différentes et où aucune région ne peut être découpée en sous-régions significativement différentes.

En raison de la grande variabilité des images naturelles, ces seuils de décision sont difficiles à déterminer et sont le plus souvent laissés à la discrétion de l'utilisateur, de façon directe au niveau régional [FH04, CM02, RP98] ou de façon plus globale en introduisant un critère d'arrêt [AO07]. Il est cependant souhaitable pour un algorithme de segmentation, quelles que soient ses qualités pratiques, de pouvoir garantir certaines propriétés, notamment de robustesse, sur les images à analyser. Cet objectif est incompatible avec l'utilisation de quantités *a priori*, car il semble en effet impossible de définir un *a priori* qui soit à la fois précis et adapté à l'ensemble des images naturelles [DMM01].

C'est pourquoi nous proposons ici d'aborder le problème de la segmentation par une approche *a contrario*, qui ne nécessite pas d'*a priori* quantitatif et repose essentiellement sur l'hypothèse intuitive que les différences observées entre deux régions issues de phénomènes différents sont statistiquement plus grandes que les différences qui peuvent être observées par hasard dans une zone homogène. À partir de cette hypothèse, deux régions seront déclarées significativement différentes si la probabilité d'observer des différences aussi fortes par hasard est suffisamment faible. Cette approche n'est pas nouvelle : dans plusieurs travaux un modèle d'homogénéité a été proposé (généralement des valeurs de pixels indépendantes et identiquement distribuées), et les différences entre régions sont ensuite déduites de la façon dont les régions dévient de ce modèle, par exemple en utilisant le test de Fisher dans [ZY96], le test de Wilcoxon-Mann-Whitney dans [CL94, WHMM06] ou l'inégalité des différences bornées dans [NN04].

Cependant, ces travaux ne fournissent pas pour autant de méthode fondée pour déterminer les seuils de décisions. Nous proposons ici de déterminer automatiquement les seuils de décisions en cherchant à assurer l'ε-fiabilité de l'algorithme final. Une fausse alarme se produit lorsque l'algorithme considère que deux régions sont significativement différentes alors qu'en réalité leurs différences sont le résultat du hasard. Nous souhaitons donc garantir que, quelle que soit l'image en entrée, l'espérance du nombre de fausses alarmes est inférieure à ε. Ceci assurera une robustesse universelle des partitions obtenues par l'algorithme.

L'ensemble des couples de régions possibles dans une image est énorme, ils ne peuvent donc pas être tous analysés. C'est pourquoi les algorithmes de segmentation reposent sur des heuristiques, le plus souvent gloutonnes et itératives, pour parcourir l'espace des candidats et trouver les régions les plus significativement différentes. Ces heuristiques sont souvent dirigées par les données, en regroupant par exemple en priorité les régions les plus similaires. Elles sont en revanche largement sous-optimales, et donc la méthode purement analytique de la section 1.4 n'est pas applicable ici. De plus, il peut être souhaitable de faire interagir plusieurs me-

sures de différences, et donc plusieurs variables discriminantes, ce qui est difficile dans le cadre purement analytique.

Ceci nous a conduits à développer une procédure d'estimation des seuils de décision basée sur des simulations *a contrario*. L'idée principale est de mesurer, pour une heuristique et des mesures de différences données, quelles sont les différences que l'on peut observer dans des images de bruit blanc. Ensuite, sous certaines conditions, nous allons montrer que nous pouvons en déduire des seuils qui garantissent l'ε-fiabilité de l'algorithme de segmentation.

Cette procédure plus souple nous permet de proposer une méthode générique de segmentation qui peut intégrer diverses mesures de différence et la plupart des heuristiques d'exploration ascendantes, tout en garantissant les mêmes propriétés de robustesse. À titre d'exemple, nous allons proposer un ensemble de mesures de différence et un algorithme de regroupement de régions qui pourra être utilisé en post-filtrage de la plupart des méthodes de segmentation existantes. Ce filtrage permet d'assurer qu'en moyenne, pour une image, moins de ε régions adjacentes dans les partitions obtenues auront des différences dues au hasard.

2.3.2 Algorithme de segmentation ε-fiable

Nous considérons la famille d'algorithmes de segmentation composés de deux éléments :
- Une fonction dite de *sélection* qui décide, pour un couple de régions adjacentes, si elles sont significativement différentes ou non. Deux régions seront dites *sélectionnées* — et donc gardées distinctes — si leurs différences sont considérées significatives par la fonction de sélection.
- Une heuristique qui explore les couples de régions dans l'image à segmenter, et qui construit la partition finale en fonction des décisions de la fonction de sélection sur les régions rencontrées. Pour obtenir un algorithme de segmentation valide, l'heuristique ne doit conserver dans la partition finale que des régions dont les voisines sont significativement différentes. Nous donnerons un exemple d'heuristique dans la section 2.3.9.

Nous pouvons maintenant définir plus précisément la notion d'algorithme de segmentation ε-fiable.

Définition 5. *Algorithme de segmentation ε-fiable.*
Soit $\mathcal{A}$ un algorithme de segmentation constitué d'une heuristique $\mathcal{H}$ et d'une fonction de sélection S. Soit $\mathcal{H}_W$ l'ensemble des couples de régions adjacentes analysés par $\mathcal{H}$ et considérés significativement différents par S sur une image. Considérons maintenant le nombre de couples de régions dans $\mathcal{H}_W$ dont les différences sont en réalité le résultat du modèle a contrario choisi pour l'hypothèse de hasard H_0. Si l'espérance de ce nombre est inférieure à ε, alors $\mathcal{A}$ est dit ε-fiable pour le modèle a contrario proposé.

Cette définition correspond à la définition d'algorithme ε-fiable de la section 1.2 : un algorithme de segmentation ε-fiable doit produire, en moyenne, moins de ε fausses alarmes par image. Une fausse alarme correspond ici à un couple de régions jugé significativement différent alors que ses différences sont le résultat du hasard.

Il nous faut maintenant définir un modèle *a contrario*.

Définition 6. *Modèle a contrario pour H_0. Sous l'hypothèse de hasard H_0, les valeurs de pixels au sein de deux régions adjacentes sont indépendantes et identiquement distribuées.*

Dans ce modèle, il n'y a donc pas de dépendance spatiale entre les pixels, qui sont organisés de façon purement aléatoire. Les différences entre groupes de pixels connectés sont donc accidentelles. Notons cependant que pour les images naturelles, l'échantillonnage de Nyquist lui-même implique que seuls les pixels dont la distance est supérieure à deux peuvent être indépendants [DMM01]. Nous ignorons toutefois cet élément dans la suite et la distance de Nyquist sera implicitement prise en compte dans la section 2.3.7 en générant des images de bruit blanc bien échantillonnées.

2.3.3 Probabilité de fausse alarme pour un couple de régions

Nous proposons maintenant un exemple de probabilité de fausse alarme pour les couples de régions. Cette probabilité sera seuillée dans la section 2.3.4 pour constituer une fonction de sélection *a contrario*. Notons que pour obtenir un algorithme de segmentation ε-fiable, il n'est pas strictement nécessaire d'utiliser une fonction de sélection qui soit elle-même basée sur un raisonnement *a contrario*. Ce choix est cependant motivé par la volonté de s'affranchir de tout *a priori* numérique et apporte une cohérence qui sera utile pour démontrer l'ε-fiabilité de l'algorithme final.

Nous détaillons tout d'abord les quatre variables discriminantes que nous proposons, avec leurs distributions respectives sous l'hypothèse *a contrario*. Ces variables seront ensuite utilisées conjointement pour définir une *PFA* sur les couples de régions.

Différence de distributions de niveaux de gris

Nous proposons tout d'abord de mesurer les différences de distributions de niveaux de gris entre les deux régions en reposant sur le test de Wilcoxon-Mann-Whitney. Ce test est bien adapté, car il est valable pour toutes les tailles de régions, et ne requiert pas d'hypothèse sur les distributions des deux régions. Ce test probabiliste mesure, de façon indirecte, les différences entre les médianes des valeurs de pixels sur les deux régions. Soit $w = (R_1, R_2)$ le couple constitué des deux régions adjacentes considérées. Les valeurs de pixels sur les deux régions sont regroupées et triées, et le test repose sur la variable aléatoire U calculée en additionnant les rangs des pixels de R_1. La distribution de U sous l'hypothèse où les valeurs de pixels dans les deux régions sont indépendantes et issues de la même distribution dépend des tailles N_1 et N_2 de respectivement R_1 et R_2. La distribution de U en fonction de N_1 et N_2, notée $P_{H_0}[U(w)] = P_{H_0}(U \leq U(w) \mid N_1 = N_1(w), N_2 = N_2(w))$, est calculable de façon exacte pour les petites régions, et est donnée par l'approximation suivante pour les grandes régions (en pratique, à

partir de 20 pixels) :

$$P_{H_0}[U(w)] = \mathcal{N}_{\leq}\left(U(w), \frac{N_1 N_2}{2}, \sqrt{\frac{N_1 N_2 (N_1 + N_2 + 1)}{12}}\right)$$

avec $\mathcal{N}_{\leq}(x, \mu, \sigma)$ la fonction de répartition de la loi normale de paramètres (μ, σ) évaluée en x. Plus les médianes des deux régions sont proches, et plus U est proche de $\frac{N_1 N_2}{2}$. On déduit de U la première variable de distance discriminante $D_1 = |U - \frac{N_1 N_2}{2}|$, dont la distribution sous l'hypothèse *a contrario*, notée $P_{H_0}[D_1(w)] = P_{H_0}(D_1 \geq D_1(w) \mid N_1 = N_1(w), N_2 = N_2(w))$, est trivialement déduite :

$$P_{H_0}[D_1(w)] = 2 \times \mathcal{N}_{\geq}\left(D_1(w), 0, \sqrt{\frac{N_1(w) N_2(w) (N_1(w) + N_2(w) + 1)}{12}}\right)$$

avec $\mathcal{N}_{\geq}$ la fonction de répartition complémentaire de la loi normale.

Différences de variance

Pour capturer également le caractère homogène de chacune des deux régions, nous proposons de mesurer à quel point chacune des variances de R_1 et R_2 est faible par rapport à la variance globale de $R_1 \cup R_2$. Soit $\sigma^2(w)$ et $\mu_4(w)$ les moments d'ordre 2 et 4 de la distribution de niveau de gris du couple w. Ces variables ne sont pas discriminantes par elles-mêmes, aussi nous les considérons comme des variables conditionnantes. Grâce à l'approximation suivante [Sap90], il est alors possible de calculer la distribution de la variance empirique S_1^2 de R_1 sous l'hypothèse *a contrario*, notée $P_{H_0}[S_1^2(w)] = P_{H_0}(S_1^2 \leq S_1^2(w) \mid \sigma^2 = \sigma^2(w), \mu_4 = \mu_4(w), N_1 = N_1(w))$:

$$P_{H_0}[S_1^2(w)] \simeq \mathcal{N}_{\leq}\left(S_1^2(w), \sigma^2(w), \frac{\sqrt{\mu_4(w) - \sigma^4(w)}}{\sqrt{N_1(w)}}\right)$$

On peut appliquer le même raisonnement pour la variance empirique S_2^2 de R_2 :

$$P_{H_0}[S_2^2(w)] \simeq \mathcal{N}_{\leq}\left(S_2^2(w), \sigma^2(w), \frac{\sqrt{\mu_4(w) - \sigma^4(w)}}{\sqrt{N_2(w)}}\right)$$

On en déduit deux variables discriminantes $D_2 = \frac{1}{1+S_1}$ et $D_3 = \frac{1}{1+S_2}$. Plus $D_2(w)$ et $D_3(w)$ sont grandes, et plus les régions sont différentes, et leurs distributions sont données par :

$$P_{H_0}(D_2 \geq D_2(w) \mid \sigma^2 = \sigma^2(w), \mu_4 = \mu_4(w), N_1 = N_1(w)) = P_{H_0}[S_1^2(w)]$$

$$P_{H_0}(D_3 \geq D_3(w) \mid \sigma^2 = \sigma^2(w), \mu_4 = \mu_4(w), N_2 = N_2(w)) = P_{H_0}[S_2^2(w)]$$

Ces approximations ne sont toutefois pas valides pour des petits échantillons, et donc nous n'utilisons ces deux mesures de distance que si N_1 et N_2 sont supérieures à 20. Ceci correspond à des régions de l'ordre de 4×5 pixels, pour lesquelles nous reposons donc uniquement sur les autres mesures de distance.

Contraste le long de la frontière

Reprenant les travaux de [DMM01], la dernière variable discriminante correspond au contraste minimal le long de la frontière entre les deux régions. La différence réside ici dans l'estimation de la distribution régionale $G[x] = P(X \geq x)$ des intensités de gradient, qui est calculée empiriquement sur $R_1 \cup R_2$ au lieu de l'image entière, afin de rester cohérent avec le modèle *a contrario*. En calculant les intensités de gradient avec un schéma de différences 2x2, les valeurs sont indépendantes sous l'hypothèse *a contrario* si leur distance est supérieure à 2. Ceci conduit à retenir comme variable discriminante D_4 la valeur minimale des intensités de gradient obtenues en prenant un pixel sur deux le long de la frontière. La longueur de la frontière et la distribution G sont prises comme variables conditionnantes, et l'on obtient :

$$P_{H_0}(D_4 \geq D_4(w) \mid G = G(w), L_{12} = L_{12}(w)) = G(w)[D_4(w)]^{\frac{L_{12}(w)}{2}}$$

Combinaison des différentes distances

Nous avons retenu quatre variables discriminantes D_1, D_2, D_3, D_4 et six variables conditionnantes $\mathbb{Y} = \{N_1, N_2, \sigma^2, \mu_4, G, L_{12}\}$. Nous en déduisons la probabilité de fausse alarme suivante pour un couple de régions w :

$$\textit{PFA}(w) = P_{H_0}(D_1 \geq D_1(w), D_2 \geq D_2(w), D_3 \geq D_3(w), D_4 \geq D_4(w) \mid \mathbb{Y} = \mathbb{Y}(w))$$

$\textit{PFA}(w)$ est la probabilité jointe d'observer quatre distances aussi fortes que celles de w sous l'hypothèse *a contrario* où les pixels sont indépendants et identiquement distribués dans les deux régions. Cette probabilité est difficile à calculer de façon exacte, aussi nous proposons l la fonction suivante pour ordonner les couples de régions en fonction de leur significativité.

Définition 7. *Fonction de différence F. Soit $\mathbb{Y} = \{N_1, N_2, \sigma^2, \mu_4, G, L_{12}\}$ le vecteur aléatoire regroupant les six variables conditionnantes. Pour un couple de régions w, la fonction de différence F est définie par :*

$$F(w) = -\log\left(\prod_{k=1}^{4} P_{H_0}(D_k \geq D_k(w) \mid \mathbb{Y} = \mathbb{Y}(w))\right)$$

$F(w)$ correspond à $-\log \textit{PFA}(w)$ dans le cas où les variables discriminantes sont indépendantes. L'opposé du logarithme permet d'avoir un schéma numérique plus adapté et d'obtenir une mesure de différence : plus $F(w)$ est élevée, plus l'hypothèse H_0 est improbable, et plus les régions sont considérées significativement différentes. F est seulement une approximation de $-\log \textit{PFA}$, mais nous supposons ici que l'ordre donné par F aux couples de régions est très proche de celui donné par *PFA* : quand un couple de régions est considéré plus significativement différent qu'un autre d'après la mesure *PFA*, alors il le sera également, dans la majorité des cas, d'après la mesure F. Cette hypothèse pourra éventuellement être vérifiée expérimentalement pour les mesures de distance choisies. Notons toutefois que cette approximation ne remet pas en cause la fiabilité de l'algorithme de segmentation obtenu en terme de nombre de fausses alarmes, elle peut seulement diminuer la sensibilité et la pertinence des régions détectées.

2.3.4 La fonction de sélection S_δ

Il est immédiat de construire une fonction de sélection à partir de F en introduisant un seuil. Ce seuil devrait alors être choisi de manière à garantir que cette fonction de sélection, couplée avec l'heuristique d'exploration choisie, constitue un algorithme de segmentation ε-fiable. Mais un unique seuil a peu de sens. En effet, dans une image, il y a beaucoup plus de régions possibles de taille moyenne que de régions de très petite ou de très grande taille. De fait, pour une image de taille $N \times M$, il y a seulement $N \times M$ régions de un seul pixel, et une seule région de $N \times M$ pixels. Entre ces deux extrêmes, le nombre de régions possibles croît puis décroît de façon exponentielle, de façon assez similaire à des coefficients binomiaux. Aussi, statistiquement, il est beaucoup plus probable d'observer par hasard de grandes différences au sein des couples de régions de taille moyenne qu'au sein des couples de petites ou grandes régions.

Une autre source de variabilité importante est l'heuristique d'exploration elle-même, qui peut être meilleure pour trouver les différences parmi les couples de régions d'une certaine taille, indépendamment du nombre de couples potentiellement analysables. Par exemple, une heuristique naïve qui regroupe itérativement les régions en suivant un ordre "scanline" (parcours de l'image ligne après ligne) prend souvent de mauvaises décisions locales, et elle est donc moins susceptible de trouver des couples de grandes régions qui ont des différences significatives que des couples de petite taille, comme le montre la figure 2.12.

Pour prendre en compte ces variabilités et donc être plus discriminant, le seuil devrait donc être adapté à la taille des régions analysées. Il est à noter que l'on retrouve ici une motivation similaire à celles de [GM06] et [RMlHM07], qui ont pondéré les *PFA* des observations en fonction du nombre de candidats dans leur catégorie. L'intérêt d'un seuillage multiple dans notre cas sera confirmé expérimentalement dans la section 2.3.9. Il y a environ $\frac{(N\times M)^2}{2}$ tailles possibles pour un couple de régions dans une image $N \times M$. Ce nombre est trop grand pour associer un seuil à chaque taille, car les seuils seront estimés individuellement par des simulations de Monte-Carlo dans la section 2.3.5. Plus le nombre de seuils est élevé, plus la procédure d'estimation sera longue. Pour réduire le nombre de seuils à estimer, nous partitionnons les différents cas en reposant sur une fonction de quantification.

Définition 8. *Fonction de partition.*
Soit $N \times M$ la taille de l'image à analyser et K une constante entière positive. Une fonction de partition $\mathcal{J}$ est une fonction de quantification qui associe à chaque couple de tailles de régions un entier compris entre 1 et K. Il s'agit donc d'une fonction de $\{1 \dots N \times M\}^2$ dans $\{1 \dots K\}$.

Une fonction de partition quantifie donc les couples de tailles de région entre 1 et K. Pour construire une fonction adaptée à la segmentation d'image, il est naturel de commencer par une log-quantification des tailles. En effet, un compte précis au pixel près n'est pas nécessaire pour des régions de grande taille, seule une précision relative est utile. Donc, étant donné les dimensions de l'image N et M, nous introduisons $\mathrm{lq}_S(n)$, une fonction qui quantifie une taille

de région n (comprise entre 1 et $N \times M$) sur S niveaux (entre 1 et S) :

$$\mathrm{lq}_S(n) = \left\lfloor \frac{S \times \log(n)}{\log(N \times M + 1)} \right\rfloor + 1$$

On peut maintenant associer à chaque couple de régions de tailles (N_1, N_2) le couple de tailles quantifiées $(\mathrm{lq}_S(N_1), \mathrm{lq}_S(N_2))$. Il y a alors S^2 tailles possibles. En ordonnant N_1 et N_2 de façon à ce que $N_1 \leq N_2$, on réduit même ce nombre à $\frac{S(S+1)}{2}$. Ceci permet finalement de partitionner les couples de régions en $K = \frac{S(S+1)}{2}$ tailles différentes, et nous appelons $\mathcal{J}_{2d}$ la fonction de partition correspondante.

Cette fonction ne nécessite aucune hypothèse et est donc adaptée à n'importe quelle heuristique, à condition que K soit suffisamment grand pour assurer un échantillonnage assez fin. En pratique, $S = 100$ est un choix raisonnable pour log-quantifier la taille de chaque région, et le nombre de seuils est alors ramené à $K = \frac{S(S+1)}{2} = 5050$. Cependant, une étude empirique sur les heuristiques que nous allons utiliser dans la section 2.3.9 montre que ce nombre de seuils peut être réduit encore plus sans perte significative de précision. En effet, la figure 2.11 montre que pour une taille minimale de région fixée, la taille de la plus grande région n'influence que faiblement la capacité à trouver de grandes différences. Ceci est particulièrement vrai pour les petites régions, qui sont également les plus sensibles. C'est pourquoi nous introduisons la fonction de partition suivante, qui dépend uniquement de la taille de la plus petite région :

$$\mathcal{J}(N_1, N_2) = \lfloor K \times \frac{\log(\min(N_1, N_2))}{\log(\frac{N \times M}{2} + 1)} \rfloor + 1$$

C'est la fonction $\mathcal{J}$ qui sera utilisée dans toutes nos expérimentations. Comme il n'y a plus qu'une seule dimension à quantifier, la section 2.3.9 montrera que le nombre de catégories de tailles peut être réduit sans incidence à $K = 100$.

Nous pouvons maintenant définir la fonction de sélection *a contrario* finale, basée sur un seuillage multiple de la fonction de différence F (définition 7).

Définition 9. *Soit $\delta = (\delta_1, \delta_2, \ldots, \delta_K)$ un $K-$uplet de seuils réels, et $\mathcal{J}$ une fonction de partition. Nous appelons S_δ la fonction qui sélectionne un couple de régions w si et seulement si :*

$$F(w) > \delta_{\mathcal{J}(N_1, N_2)}$$

où N_1 et N_2 sont les nombres de pixels dans la première et la deuxième région.

Cette fonction sélectionne donc les couples de régions dont les différences, mesurées par la fonction F, sont supérieures au seuil correspondant à leurs catégories de taille. Il reste maintenant à rechercher les valeurs des K seuils qui permettent d'assurer l'ε-fiabilité de l'algorithme de segmentation pour une heuristique d'exploration donnée.

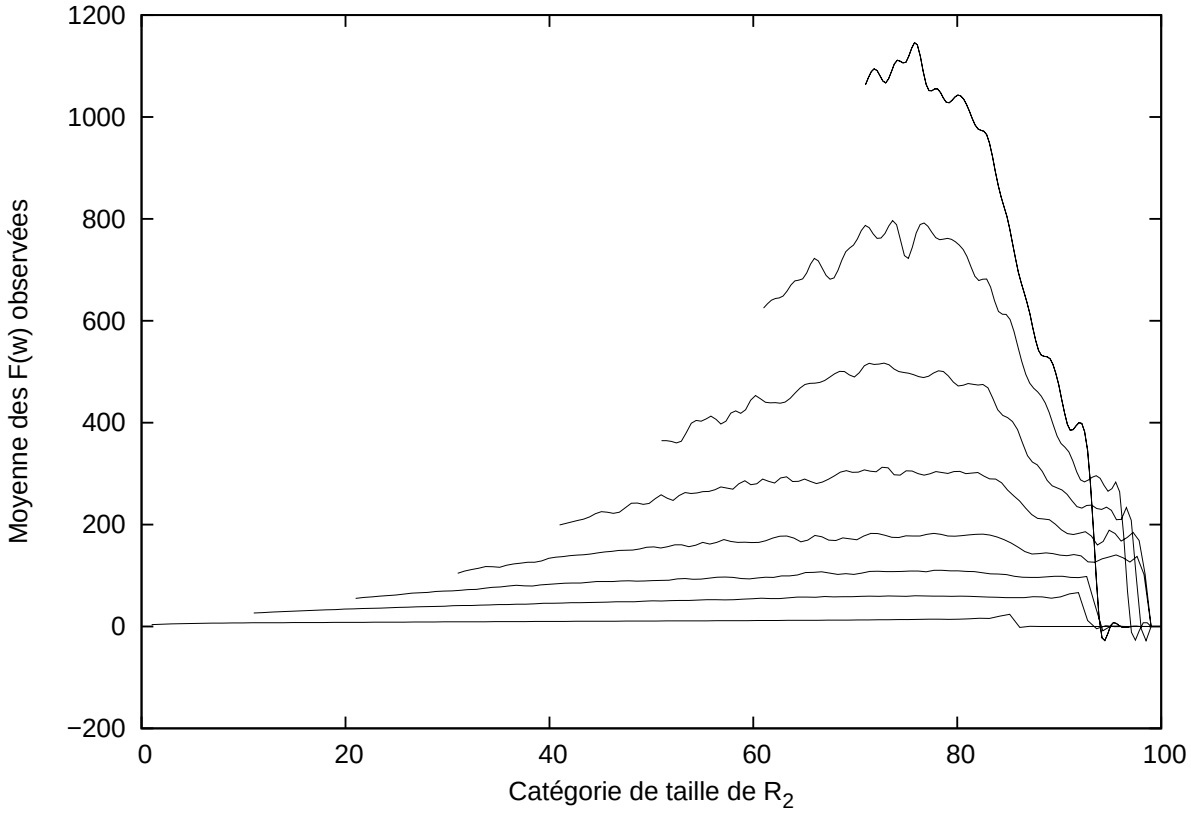


FIG. 2.11 – Moyenne des valeurs de différences F observées en appliquant l'algorithme 4 (voir section 2.3.9) avec une initialisation par ligne de partage des eaux sur des images de bruit blanc uniforme de taille 256×256. Pour chaque couple de régions w, soit R_1 la plus petite région et R_2 la plus grande. Chaque courbe correspond à une catégorie de taille pour R_1 fixe, allant de 1 (en bas) à 71 (en haut) par pas de 10 (pour des raisons de lisibilité, les 29 catégories restantes ne sont pas représentées). L'abscisse donne la catégorie de taille de R_2. Les valeurs de F ne dépendent que de façon marginale de la taille de R_2 pour une taille minimale donnée, surtout pour les petites régions, qui sont les plus sensibles. Ceci conduit au choix de la fonction $\mathcal{J}$ de la section 2.3.4.

2.3.5 Calcul des seuils de significativité

2.3.6 Calcul purement analytique impossible

Les seuils de significativité ne peuvent pas être calculés de façon purement analytique ici, car plusieurs des conditions nécessaires exhibées dans la section 1.4 ne sont pas satisfaites. Premièrement, la probabilité de fausse alarme que nous utilisons, à travers la fonction F, repose sur plusieurs variables discriminantes. Deuxièmement, les heuristiques d'exploration utilisées pour la segmentation sont le plus souvent dirigées par les données mais largement sous-optimales, comme le montre la figure 2.12. Aucun des deux cas extrêmes de la section 1.4 n'est donc adapté. Enfin, même si l'heuristique pouvait être considérée optimale, c'est-à-dire si elle analysait la quasi-totalité des couples de régions très différents, le calcul du nombre total de couples de régions possibles dans une image est un problème difficile de combinatoire énumérative.

Nous proposons donc une autre méthodologie, basée sur des simulations, pour estimer les seuils assurant l'ε-fiabilité pour une heuristique donnée.

2.3.7 Calcul des seuils par simulation *a contrario*

Nous simplifions tout d'abord le problème grâce à la proposition suivante, qui permet d'estimer chaque seuil de façon indépendante.

Proposition 2. *Si un algorithme de segmentation $\mathcal{A}$ est $\frac{\varepsilon}{K}$-fiable pour chacune des K catégories de taille données par la fonction de partition, alors $\mathcal{A}$ est ε-fiable.*

La preuve est immédiate en utilisant la linéarité de l'espérance. Nous allons maintenant chercher les seuils δ_k qui assurent l'ε-fiabilité pour une heuristique et une taille d'image données, sur des images de bruit blanc uniforme (b.b.u.). Nous montrerons ensuite que sous certaines conditions, ces mêmes seuils assurent également l'ε-fiabilité sur toutes les images.

Grâce à la proposition 2, il est simplement nécessaire de prouver l'$\frac{\varepsilon}{K}$-fiabilité pour chaque catégorie de taille. Nous proposons l'algorithme suivant :

1. Initialiser les seuils $\delta = (\delta_1, \delta_2, \ldots, \delta_K)$
2. Générer Q images de bruit blanc uniforme de taille $N \times M$ en échantillonnant un signal de bruit blanc uniforme dont les plus hautes fréquences ont été supprimées, pour prendre en compte la distance de Nyquist (voir section 2.3.2).
3. Pour chaque image :
   - Appliquer l'heuristique d'exploration $\mathcal{H}$ avec la fonction de sélection S_δ, et compter le nombre de couples de régions sélectionnés par S_δ pour chaque catégorie de taille.
4. Pour chaque catégorie de taille k :
   - Calculer la moyenne empirique m_k et l'écart-type empirique s_k du nombre de couples de régions sélectionnés par S_δ sur les Q images

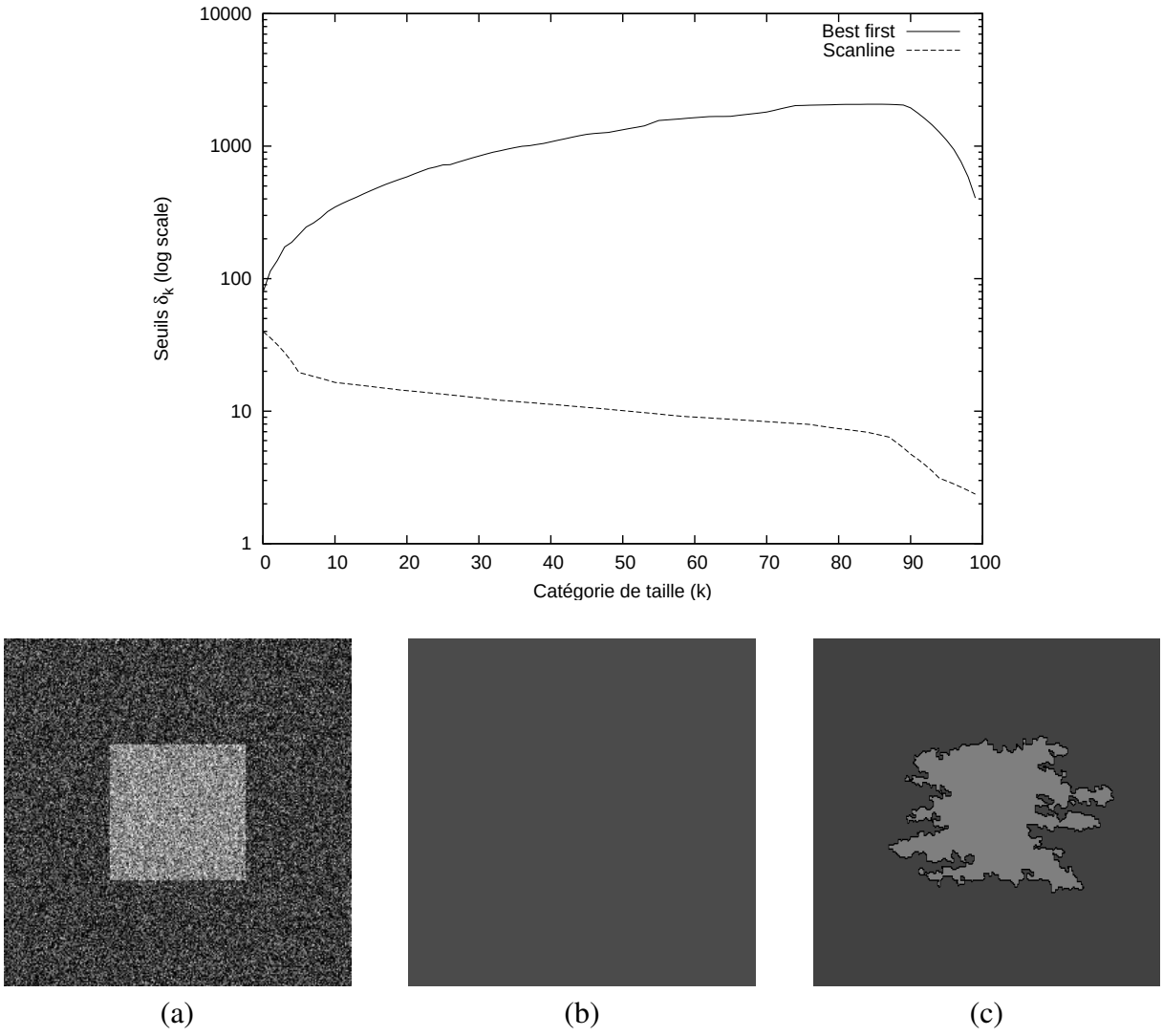


FIG. 2.12 – Cette figure montre qu'il est primordial d'adapter les seuils de décision à l'heuristique, et pas seulement au nombre de régions analysées. Le graphique du haut montre les seuils appris pour l'algorithme 4 (voir section 2.3.9) avec une initialisation par ligne de partage des eaux et deux ordres de regroupements différents. "Best first" regroupe le couple de régions avec la valeur de F la plus faible à chaque itération. "Scanline" regroupe simplement les régions non significativement différentes à chaque itération, dans l'ordre naturel de parcours d'une image, de la gauche vers la droite, et de haut en bas. Le nombre de régions analysées est approximativement équivalent dans les deux cas, mais les valeurs de F qui peuvent être observées dans des images de bruit blanc sont bien plus grandes avec l'ordre "best first". Ceci est confirmé sur l'image du carré (a). En utilisant les seuils appris pour "best first" avec l'ordre "scanline", aucun couple de régions n'est suffisamment différent pour être détecté (b). En utilisant cette fois les seuils adaptés à l'heuristique "scanline", la région du carré est détectée. Des observations similaires ont été faites dans [NN04] à propos de l'ordre de parcours des régions (figure 4), mais nous les nuançons ici en montrant que des seuils adaptés peuvent améliorer les résultats du parcours "scanline".

– Calculer un intervalle de confiance sur l'espérance réelle μ_k en utilisant la propriété classique :

$$P(M_k \leq m_k) = F_{Q-1}(\frac{m_k - \mu_k}{s_k}\sqrt{Q-1})$$

avec $F_n(x)$ la fonction de répartition d'une loi de Student avec n degrés de liberté.

– Pour un niveau de confiance choisi, si la borne supérieure estimée de μ_k devient supérieure à $\frac{\varepsilon}{K}$, alors qu'elle était inférieure à l'itération précédente, fixer le seuil δ_k final à sa valeur précédente. De même, si la borne supérieure estimée de μ_k devient inférieure à $\frac{\varepsilon}{K}$, alors qu'elle était supérieure à l'itération précédente, fixer le seuil δ_k final à sa valeur courante. Dans les autres cas, augmenter δ_k si la borne supérieure estimée de μ_k est supérieure à $\frac{\varepsilon}{K}$, sinon diminuer δ_k.

5. Répéter jusqu'à ce que tous les seuils soient fixés.

À chaque étape, μ_k représente l'espérance du nombre de fausses alarmes produites par l'algorithme pour la k-ième catégorie de taille. Après convergence, nous savons, avec le niveau de confiance choisi, que l'algorithme de segmentation est ε-fiable pour des images $N \times M$ de bruit blanc uniforme bien échantillonnées.

En pratique, une très bonne initialisation est obtenue en appliquant une première fois cet algorithme avec des seuils infinis pour empêcher toute sélection, et en stockant les valeurs de F obtenues sur les couples de régions analysés par $\mathcal{H}$ à l'étape 3. Les valeurs maximales de F pour chaque catégorie constituent alors une excellente initialisation.

De plus, la *PFA* augmente de façon exponentielle avec les variables discriminantes, et donc ε n'est pas une valeur très sensible. Tous les calculs sont effectués sur une échelle logarithmique, et des seuils qui assurent une fiabilité du même ordre que ε sont suffisants pour obtenir des résultats satisfaisants.

Aussi, en utilisant l'initialisation ci-dessus, l'algorithme proposé devient plutôt une procédure de validation que d'estimation, et la convergence des seuils est généralement obtenue en quelques itérations en utilisant un facteur géométrique de 1.01 pour augmenter/diminuer δ_k.

Notre méthode nécessite de calculer des seuils différents pour chaque taille d'image. Ceci n'est pas très problématique car les images sont le plus souvent de taille standard et les seuils peuvent donc être pré-calculés pour les tailles usuelles. De plus, les seuils évoluent de façon progressive avec la taille des images, comme le montre la figure 2.13, il est donc possible d'utiliser de l'interpolation pour les tailles non conventionnelles.

2.3.8 Conditions d'ε-fiabilité sur des images arbitraires

Les seuils calculés dans la section 2.3.7 garantissent que, en moyenne, moins de ε couples de régions seront considérés significativement différents dans une image de bruit blanc uniforme (b.b.u.). Ceci assure l'ε-fiabilité de la segmentation sur des images de b.b.u. Que peut-on en déduire sur les autres images ? Une image de b.b.u. correspond au pire cas où l'image ne contient que des couples de régions issus du modèle *a contrario*, et donc où toute détection serait une

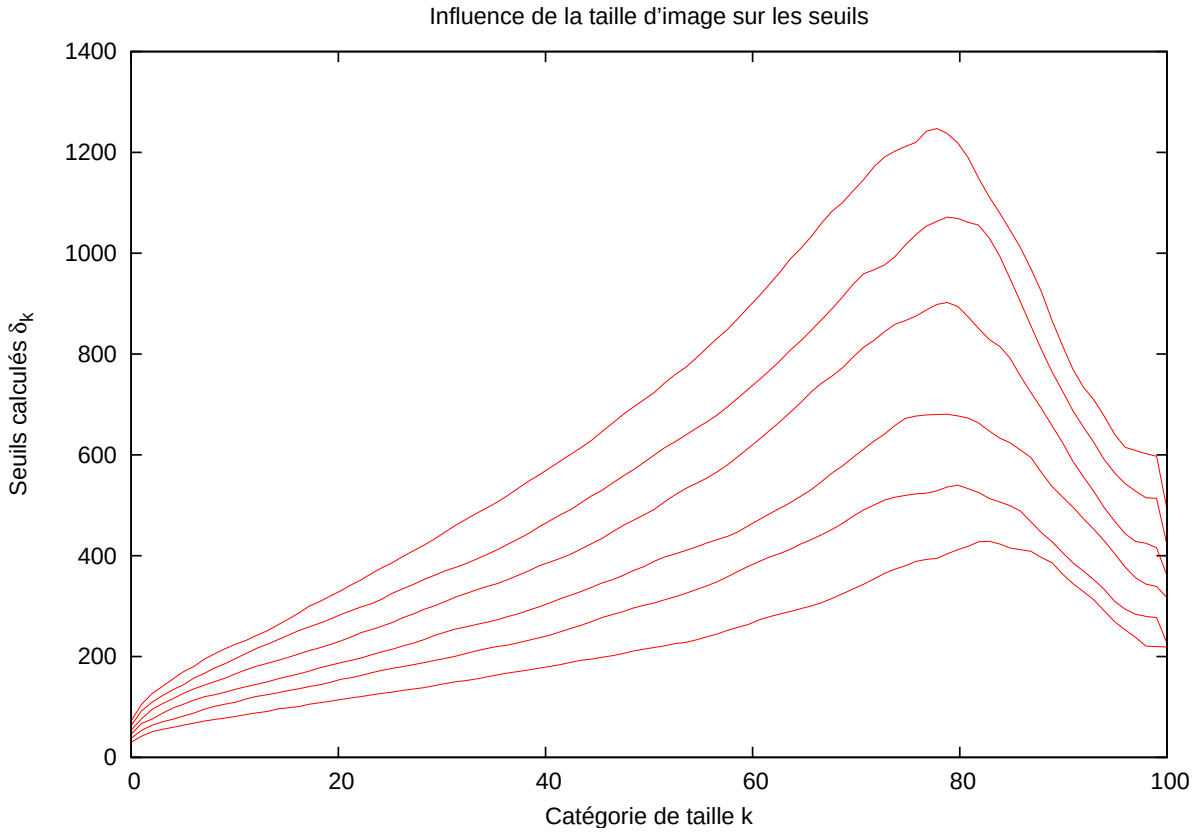


FIG. 2.13 – Valeurs de F observées en appliquant l'algorithme 4 (voir section 2.3.9) avec une initialisation par ligne de partage des eaux sur 1000 images de bruit blanc uniformes avec des tailles différentes. Les courbes sont empilées par taille d'image croissante, allant de 100×100 pixels pour la plus petite à 200×200 pour la plus grande par pas de 20 pixels sur chaque dimension. Comme les valeurs évoluent de façon régulière avec la taille des images, elles peuvent être interpolées avec précision pour les tailles intermédiaires.

fausse alarme. Aussi, si l'algorithme de segmentation n'est pas biaisé par l'uniformité du bruit blanc (le modèle *a contrario* suppose seulement que les pixels sont indépendants et identiquement distribués), ni par le fait que l'image entière est du bruit pur, alors on peut s'attendre à ce que l'algorithme produise nécessairement moins de fausses alarmes sur une image qui n'est pas du bruit pur. Ce raisonnement est formulé de façon plus formelle dans la proposition suivante, qui donne des conditions pour que les seuils obtenus dans la section 2.3.7 assurent l'ε-fiabilité sur n'importe quel type d'image.

Proposition 3. *Soit $\mathcal{A}$ un algorithme de segmentation composé d'une heuristique $\mathcal{H}$ et de la fonction de sélection S_δ. On note $\Gamma(w)$ la distribution de niveaux de gris d'un couple de régions w, et $\Gamma_u(w)$ le cas particulier où la distribution de w est uniforme. $\mathcal{A}$ est ε-fiable pour des images de taille $N \times M$ sous les conditions suivantes :*

1. *$\mathcal{A}$ est ε-fiable pour des images $N \times M$ de bruit blanc uniforme bien échantillonnées*

2. $\forall \delta, \forall \Gamma(w), \quad P_{H_0}(F(w) > \delta \mid \Gamma(w)) \leq P_{H_0}(F(w) > \delta \mid \Gamma_u(w))$

3. $\forall \delta, \forall \Gamma(w), \quad P_{H_0}(\mathcal{H}(w) \mid F(w) > \delta, \Gamma(w)) \leq P_{H_0}(\mathcal{H}(w) \mid F(w) > \delta, \Gamma_u(w))$

4. *Soit R une région dans une image où les pixels sont indépendants et identiquement distribués. Soit $\Omega = I \backslash R$ le reste de l'image. Alors, pour un couple de régions w inclus dans R, on doit avoir :*

$$\forall \delta, \quad P_{H_0}(\mathcal{H}(w) \mid F(w) > \delta, \Omega) = P_{H_0}(\mathcal{H}(w) \mid F(w) > \delta)$$

Une preuve mathématique est donnée dans l'annexe B. Nous clarifions simplement ici le sens des différentes conditions. La première est simplement l'ε-fiabilité de l'algorithme sur des images de b.b.u., qui peut être obtenue en utilisant la procédure de la section 2.3.7. La seconde spécifie que les valeurs de F dans une image (indépendamment de l'heuristique choisie) ne doivent pas être biaisées par la distribution de niveaux de gris du couple de régions, sachant que les pixels sont indépendants et identiquement distribués (i.i.d.). F est dérivée d'un produit de probabilités, et chacune d'entre elles s'adapte à la distribution des régions. Donc, la seule influence qu'il reste est l'entropie de la distribution, qui peut interdire la présence de grandes différences. Par exemple, dans le cas extrême d'une distribution de type Dirac, tous les pixels auront la même valeur et la seule valeur de F possible est $-log(1) = 0$. Comme la distribution uniforme est celle ayant l'entropie maximale, l'inégalité sera vraisemblablement vérifiée.

La troisième condition requiert que l'heuristique ne soit pas meilleure pour trouver les couples de régions i.i.d. avec une valeur F élevée quand la distribution n'est pas uniforme. Autrement dit, la distribution des régions très différentes ne doit pas aider, par elle-même, l'heuristique à les trouver.

Les conditions 2 et 3 peuvent être vérifiées expérimentalement par des simulations sur des images de bruit blanc avec différentes distributions. La figure 2.14 montre les valeurs de F obtenues pour différentes distributions par une des heuristiques qui seront utilisées dans la section 2.3.9. Elle permet de s'assurer que la probabilité jointe qu'un couple de régions soit analysé

par l'heuristique et ait une grande valeur de F ne peut que diminuer ou rester stable quand la distribution sous-jacente n'est pas uniforme.

La quatrième condition est plus difficile à vérifier de façon formelle. Étant donné une région i.i.d. R, incluse dans une image I, cette condition signifie que la capacité de l'heuristique à trouver des couples de régions dans R avec de fortes différences ne doit pas être biaisée par les propriétés des régions voisines. Intuitivement, cette condition sera au moins vérifiée pour les heuristiques ascendantes.

Il faut toutefois noter qu'en pratique, ces conditions sont plus strictes que nécessaire car l'ordre de grandeur des valeurs de F qui peuvent être obtenues par hasard est bien plus faible que les valeurs typiques obtenues sur des images naturelles, comme on peut le constater sur la figure 2.15. Ceci permet d'utiliser une large gamme d'heuristiques, et la section 2.3.9 montrera trois exemples satisfaisants.

2.3.9 Résultats

La méthode de segmentation proposée peut s'appuyer sur n'importe quelle heuristique d'exploration, si elle satisfait les conditions de la proposition 3. Nous avons choisi de l'illustrer avec l'algorithme 4, qui est basé sur une heuristique gloutonne qui regroupe itérativement les régions les moins différentes jusqu'à ce qu'il ne reste que des couples de régions significativement différents. Cet algorithme a été combiné avec trois différentes initialisations : une ligne de partage des eaux classique avec pré-filtrage gaussien [BM93] (appelée LPE), l'algorithme basé sur les graphes de [FH04] (appelé EGBIS), et l'algorithme de regroupement de régions de [NN04] (appelé SRM). Pour EGBIS et SRM, nous avons utilisé les programmes fournis par les auteurs.

Algorithme 4 : Algorithme de segmentation basé sur un post-filtrage d'une segmentation initiale en regroupant itérativement les régions les moins significativement différentes d'après la mesure de différence *a contrario* F, jusqu'à ce qu'il ne reste plus que des régions significativement différentes d'après la fonction de sélection S_δ. Le partitionnement initial est libre, au pire une région par pixel.

```
Créer une partition initiale ;
Calculer F pour tous les couples de régions adjacentes;
tant que Certains couples n'ont pas été sélectionnés par S_δ faire
    Regrouper le couple avec la valeur de F minimale;
    Mettre à jour F pour les couples de régions adjacentes ;
fin
```

Dans toutes les expériences, ε est fixé à 1 et le nombre K de catégories de taille à 100, ce qui assure une quantification suffisante. Le degré de confiance pour l'estimation des seuils de la section 2.3.7 est fixé à 0.99. Les trois algorithmes d'initialisation ont leurs propres paramètres : $\sigma = 0.8$ pour le filtre gaussien de la LPE, $\sigma = 0.8$, $k = 150$, *minarea* $= 20$ pour EGBIS, et $Q = 256$ pour SRM. Ces paramètres visent à capturer le maximum de détails dans l'image.

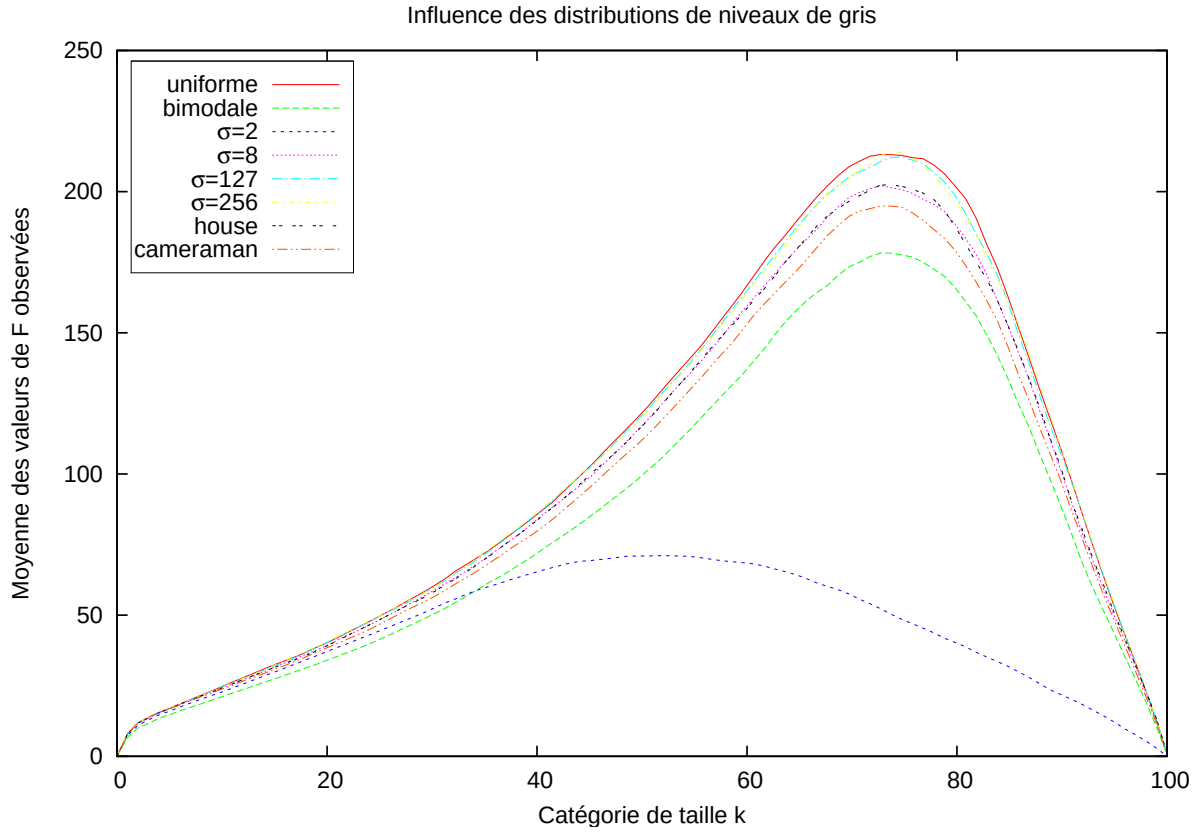


FIG. 2.14 – Valeurs moyennes de F observées en appliquant l'algorithme 4 (voir section 2.3.9) avec une initialisation par ligne de partage des eaux sur 1000 images de bruit blanc de taille 128×128 avec différentes distributions. "Bimodale" correspond à une distribution binaire avec $P(X = 0) = P(X = 255) = 0.5$. "Uniforme" correspond à du bruit blanc uniforme. "House" est la distribution de l'image de la figure 2.17, et "cameraman" est la distribution de l'image de la figure 2.18. Les autres courbes correspondent à des distributions gaussiennes centrées sur 127, avec différents écarts-types σ. $\sigma = 2$ est presque $P(X = 127) = 1$ et $\sigma = 256$ est presque uniforme. Mis à part les deux cas extrêmes bimodale et $\sigma = 2$, les courbes sont similaires au cas uniforme, montrant une bonne indépendance de F vis-à-vis de la distribution des niveaux de gris des régions.

Pour EGBIS, le paramètre *minarea*, correspondant à la taille minimale des régions conservées, peut être fixé à 0 sans changer le résultat final, car les régions trop petites seront le plus souvent non significativement différentes de leurs voisines, et donc regroupées avec d'autres régions par l'algorithme 4. Nous le conservons cependant à 20 pour mieux mettre en valeur les régions non trivialement regroupées par l'heuristique *a contrario*.

Le temps de calcul nécessaire pour le post-filtrage *a contrario* dépend du degré de sur-segmentation fourni par le partitionnement initial. Les temps observés pour des images naturelles 256×256 ne dépassent pas 500ms pour l'initialisation par LPE, 200ms pour EGBIS, et 100ms pour SRM sur un Intel Core 2 Duo cadencé à 2.4GHz, sans tirer parti du parallélisme multi-coeurs.

La figure 2.15 montre les seuils obtenus pour l'initialisation par LPE, et toutes les valeurs de différence F observées lors de la segmentation de l'image "house". On observe de larges déviations par rapport au modèle *a contrario*, matérialisées par des valeurs de F largement au-dessus des seuils. Ces valeurs correspondent à des régions significativement différentes. L'ordre de grandeur des déviations confirme la robustesse de l'approche.

Des résultats sur des images naturelles et synthétiques sont présentés dans les figures 2.16, 2.17, 2.18, 2.19, 2.20, 2.21 et 2.22. L'initialisation par LPE produit une forte sur-segmentation, qui est bien corrigée par le post-filtrage *a contrario*. Cependant, l'heuristique de l'algorithme 4 est sensible aux ambiguïtés locales et tend à produire des régions aux contours complexes. De meilleurs résultats sont obtenus avec une initialisation par EGBIS. Avec les paramètres choisis, EGBIS conserve la plupart des détails de l'image, mais produit en contrepartie un grand nombre de fausses alarmes. Celles-ci sont efficacement filtrées par les regroupements *a contrario*, qui conservent toutefois la plupart des détails. Même avec un paramètre de complexité Q élevé, la méthode SRM perd beaucoup de détails, mais produit en revanche peu de fausses alarmes. Ici, le post-filtrage *a contrario* conserve la plupart des régions, excepté dans le cas très bruité des figures 2.16 et 2.19.

Notre approche *a contrario* détecte les couples de régions dont les différences, obtenues à partir d'un certain nombre de mesures de distances, ne sont pas dues au hasard. Aussi, si les mesures de distances choisies sont sensibles aux gradients d'illumination et aux ombres, ce qui est le cas ici, notre méthode les détecte. Cependant, ces détections peuvent être évitées en utilisant une initialisation qui les élimine, par exemple EGBIS. En pratique, les besoins spécifiques pour une application précise peuvent être pris en compte soit en choisissant des mesures de distances adaptées, soit en choisissant une initialisation pertinente.

2.3.10 Discussion

À travers la segmentation d'images, nous avons montré qu'il est possible d'appliquer un raisonnement *a contrario* quand deux des conditions nécessaires au cadre purement analytique (voir section 1.4) ne sont pas respectées :

– plusieurs variables discriminantes hétérogènes sont utilisées conjointement ;

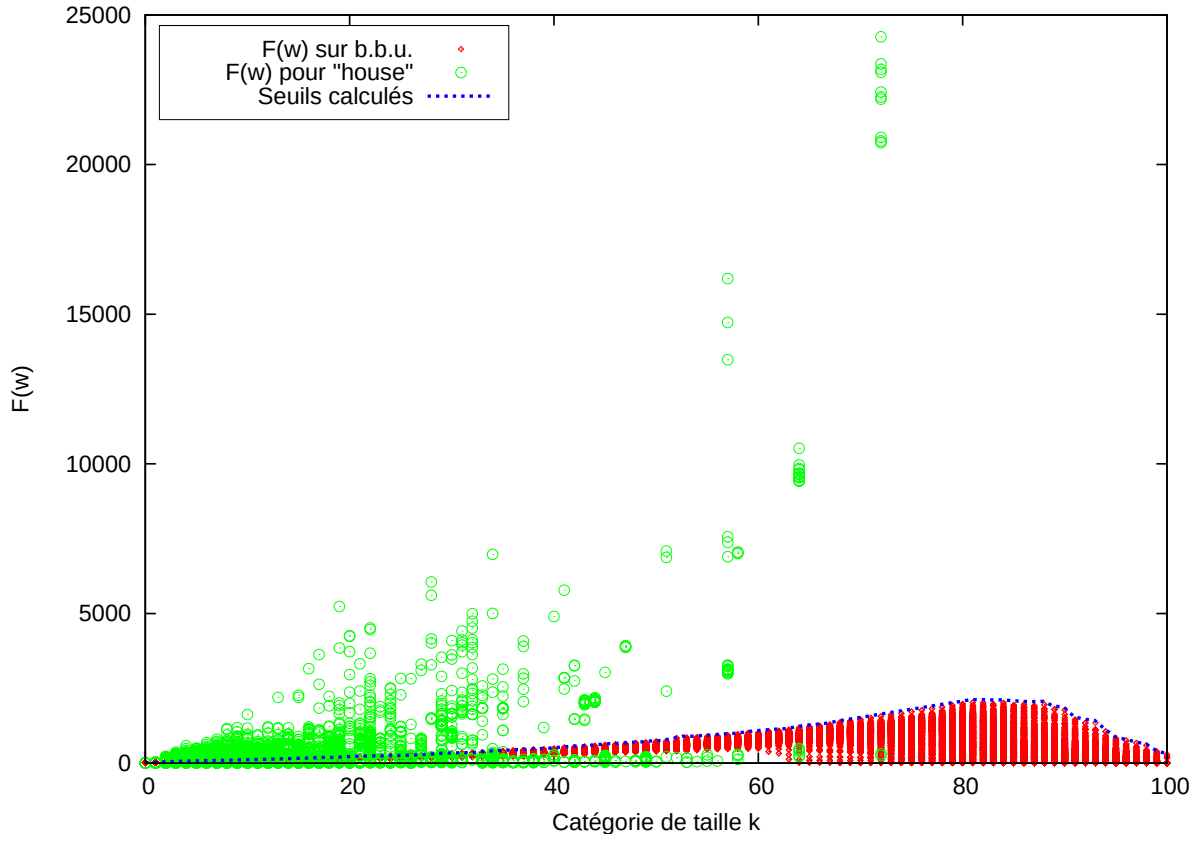
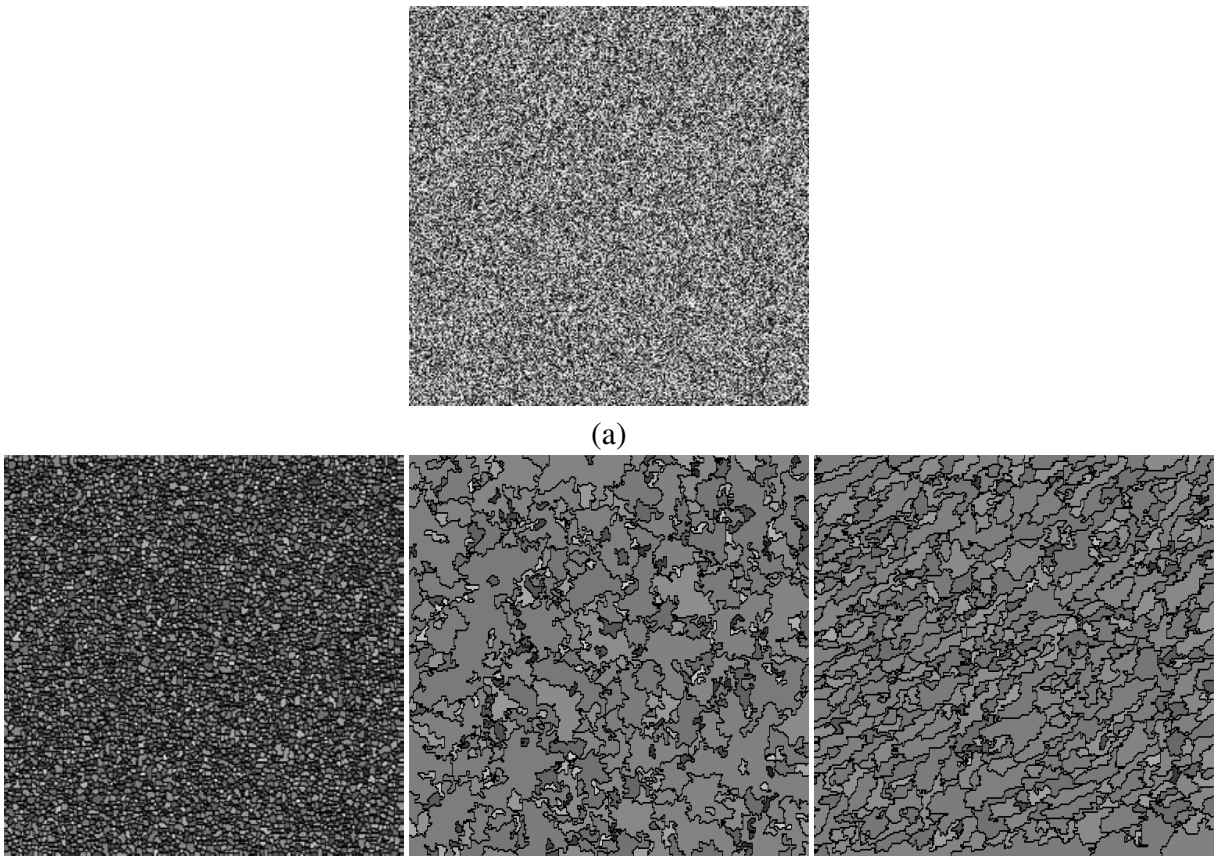


FIG. 2.15 – Valeurs de différence F obtenues par l'algorithme 4 avec une initialisation par LPE. L'algorithme est exécuté sur 1000 images de bruit blanc uniforme. Pour chaque image, la valeur maximale de F pour chaque catégorie de taille est représentée (en rouge). Les seuils optimaux obtenus par la procédure de la section 2.3.7 correspondent approximativement à ces valeurs maximales. Les valeurs de F obtenues sur l'image "house" sont également montrées (cercles noirs). De très grandes déviations sont observées, montrant la capacité de la fonction F à discriminer les conséquences du hasard des conséquences de phénomènes physiques.



(a)

(b) (c) (d)

FIG. 2.16 – (a) Image de bruit blanc uniforme 256×256 (b) Segmentation par LPE : 6229 régions (c) Segmentation par EGBIS : 349 régions (d) Segmentation par SRM : 489 régions Les trois algorithmes d'initialisation que nous expérimentons produisent des fausses alarmes sur des images de bruit pur. Dans chacun des cas, aucun couple de régions n'est conservé par le post-filtrage *a contrario* de l'algorithme 4.

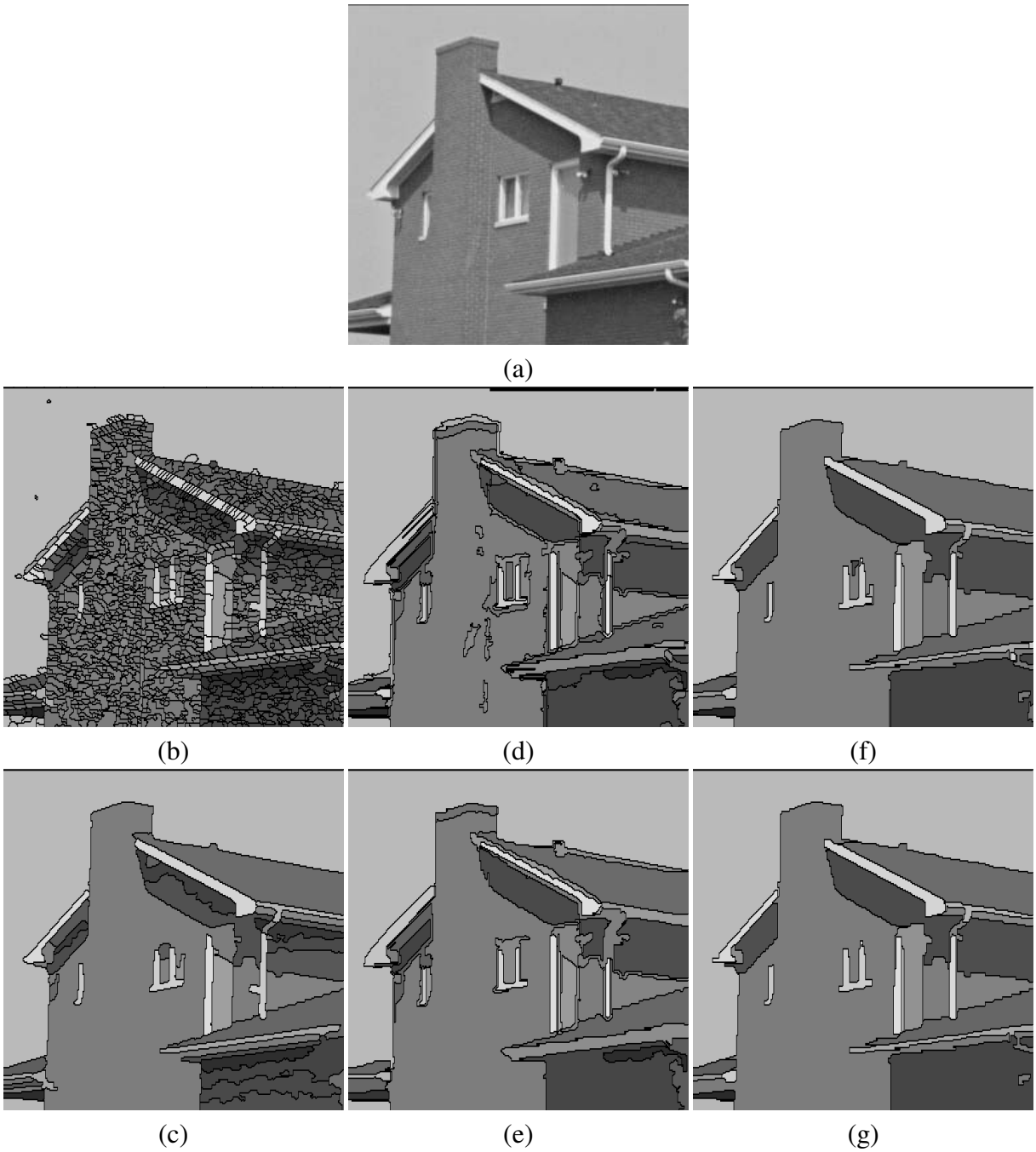
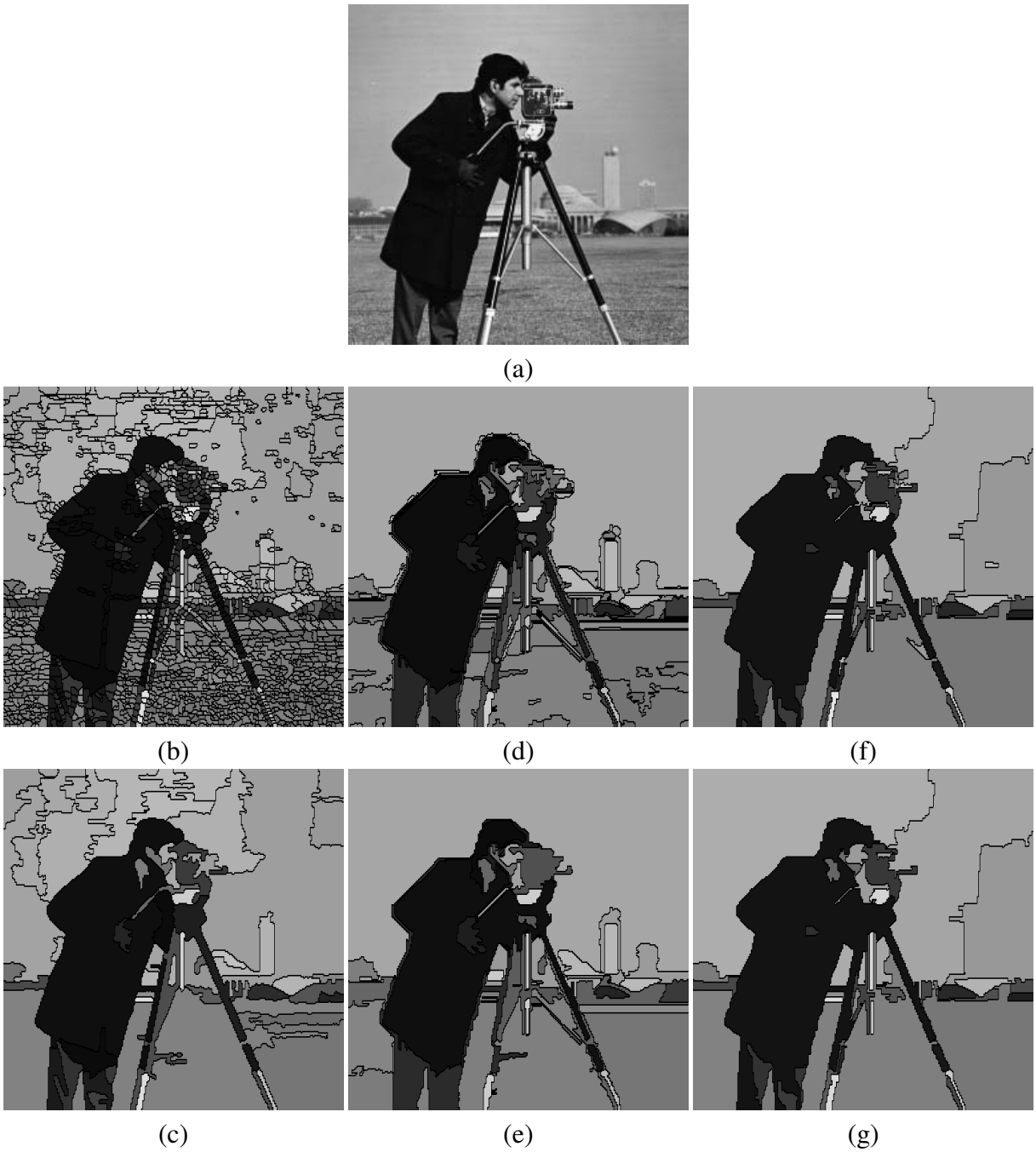


FIG. 2.17 – (a) Image "house" (b) Segmentation par LPE : 1839 régions (c) Post-filtrage par l'algorithme 4 : 47 régions. (d) Segmentation par EGBIS : 130 régions (e) Post-filtrage par l'algorithme 4 : 34 régions. (f) Segmentation par SRM : 43 régions (g) Post-filtrage par l'algorithme 4 : 31 régions.



(a)

(b) (d) (f)

(c) (e) (g)

FIG. 2.18 – (a) Image "cameraman" (b) Segmentation par LPE : 1784 régions (c) Post-filtrage par l'algorithme 4 : 64 régions. (d) Segmentation par EGBIS : 184 régions (e) Post-filtrage par l'algorithme 4 : 36 régions. (f) Segmentation par SRM : 64 régions (g) Post-filtrage par l'algorithme 4 : 42 régions.

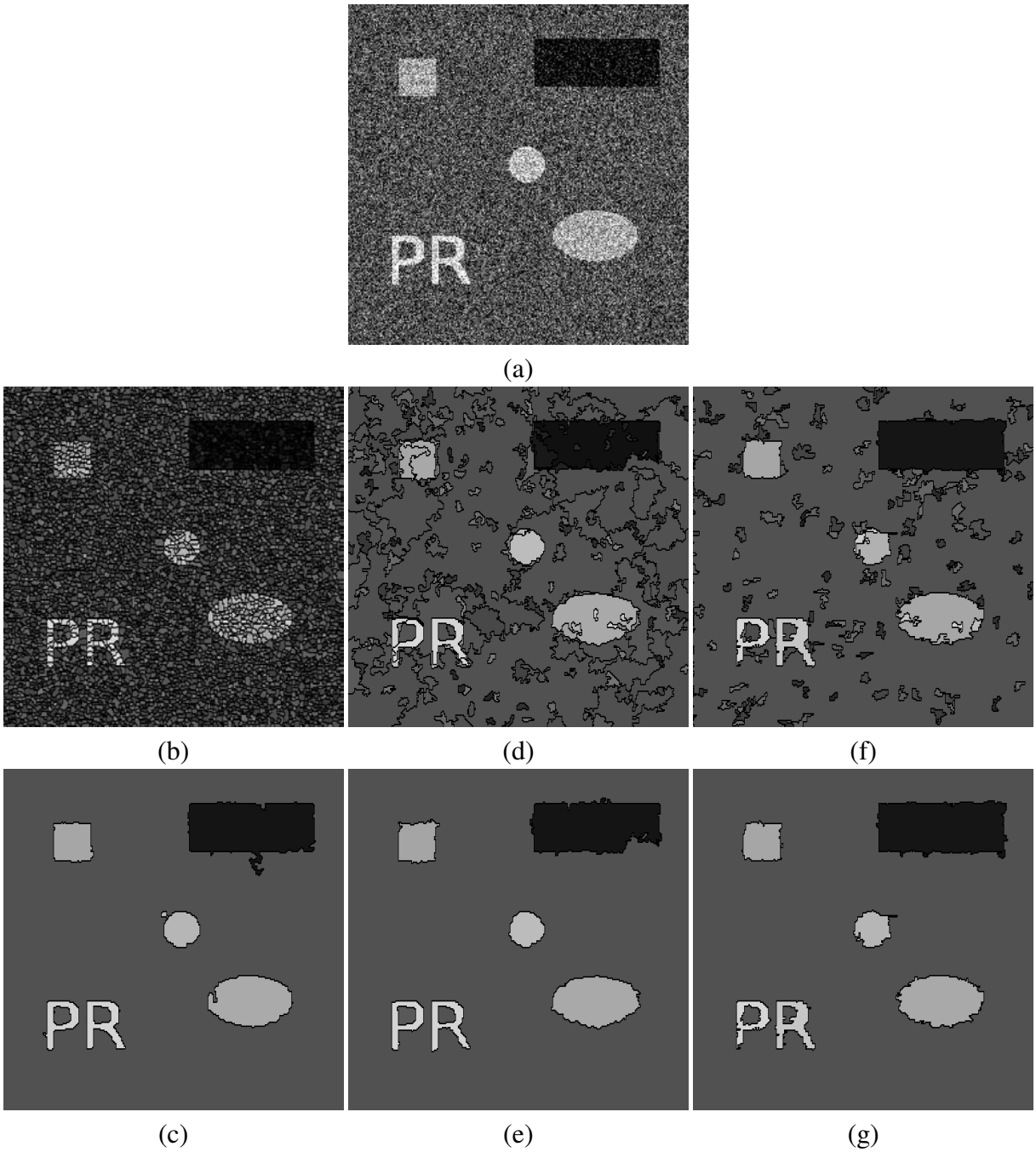


FIG. 2.19 – (a) Image synthétique très bruitée (b) Segmentation par LPE : 5662 régions (c) Post-filtrage par l'algorithme 4 : 9 régions. (d) Segmentation par EGBIS : 232 régions (e) Post-filtrage par l'algorithme 4 : 9 régions. (f) Segmentation par SRM : 184 régions (g) Post-filtrage par l'algorithme 4 : 9 régions.

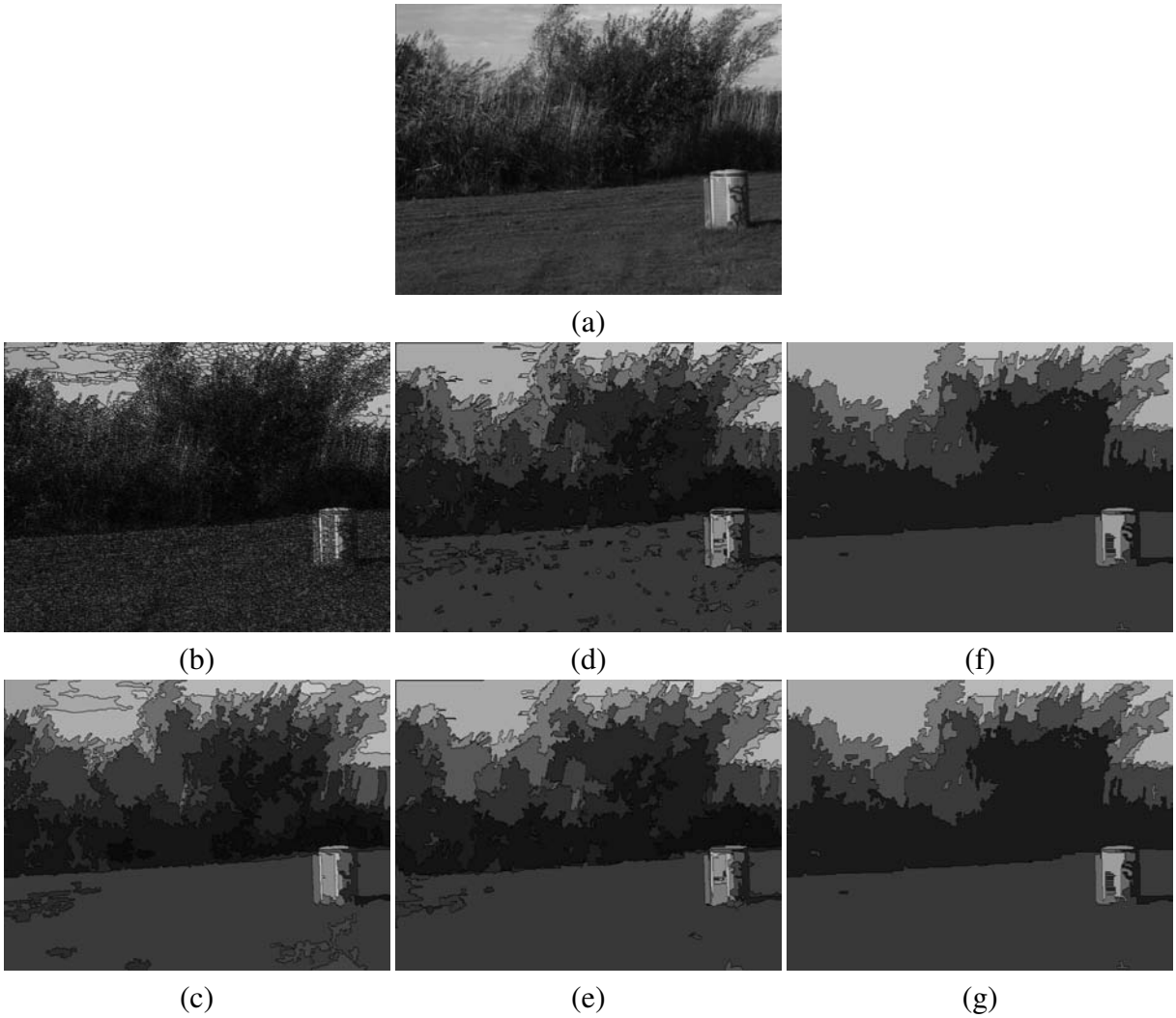
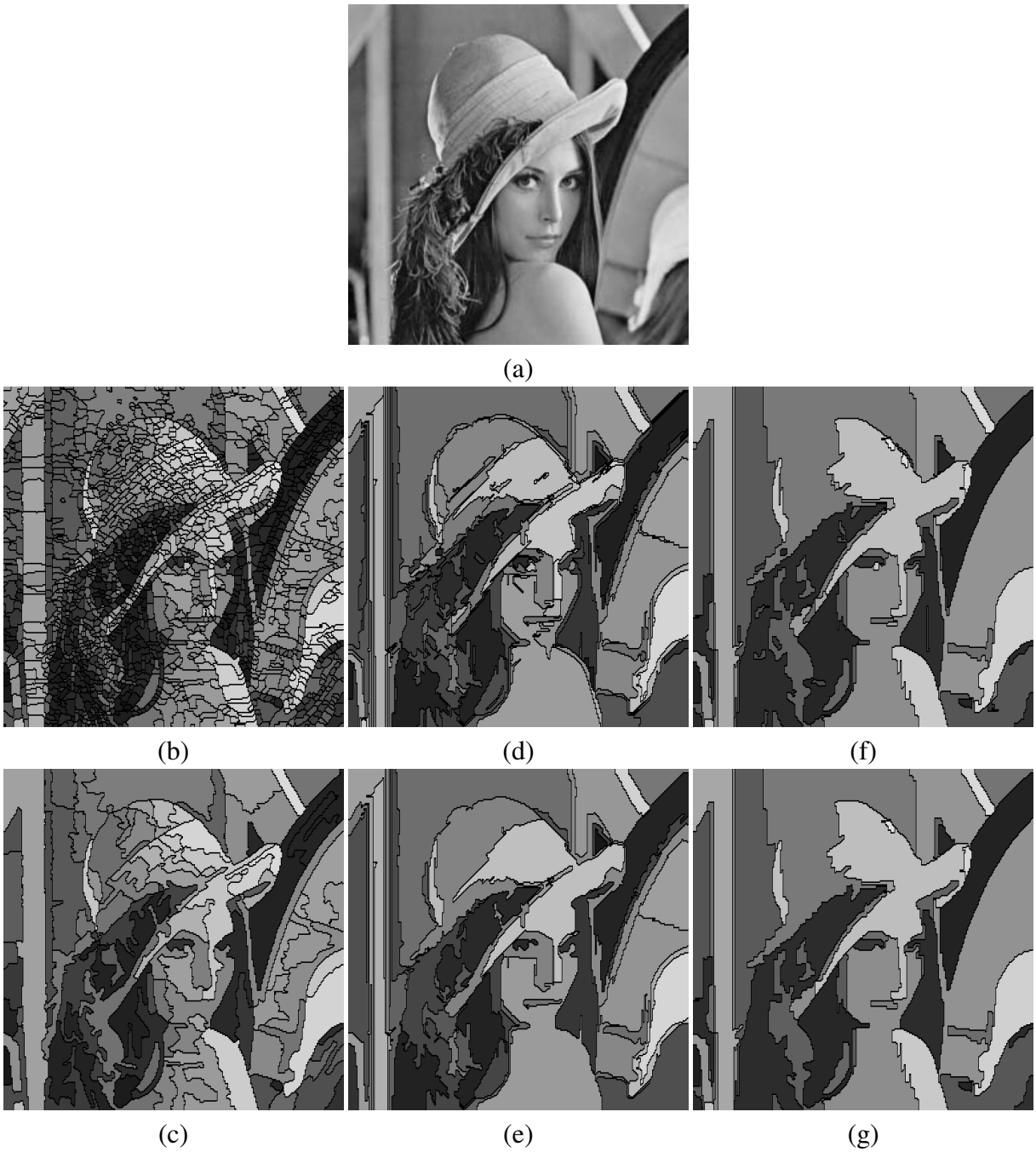


FIG. 2.20 – (a) Image "trash" (b) Segmentation par LPE : 14677 régions (c) Post-filtrage par l'algorithme 4 : 269 régions. (d) Segmentation par EGBIS : 682 régions (e) Post-filtrage par l'algorithme 4 : 87 régions. (f) Segmentation par SRM : 62 régions (g) Post-filtrage par l'algorithme 4 : 48 régions.



(a)

(b) (d) (f)

(c) (e) (g)

FIG. 2.21 – (a) Image "Lena" (b) Segmentation par LPE : 1855 régions (c) Post-filtrage par l'algorithme 4 : 94 régions. (d) Segmentation par EGBIS : 205 régions (e) Post-filtrage par l'algorithme 4 : 50 régions. (f) Segmentation par SRM : 62 régions (g) Post-filtrage par l'algorithme 4 : 48 régions.

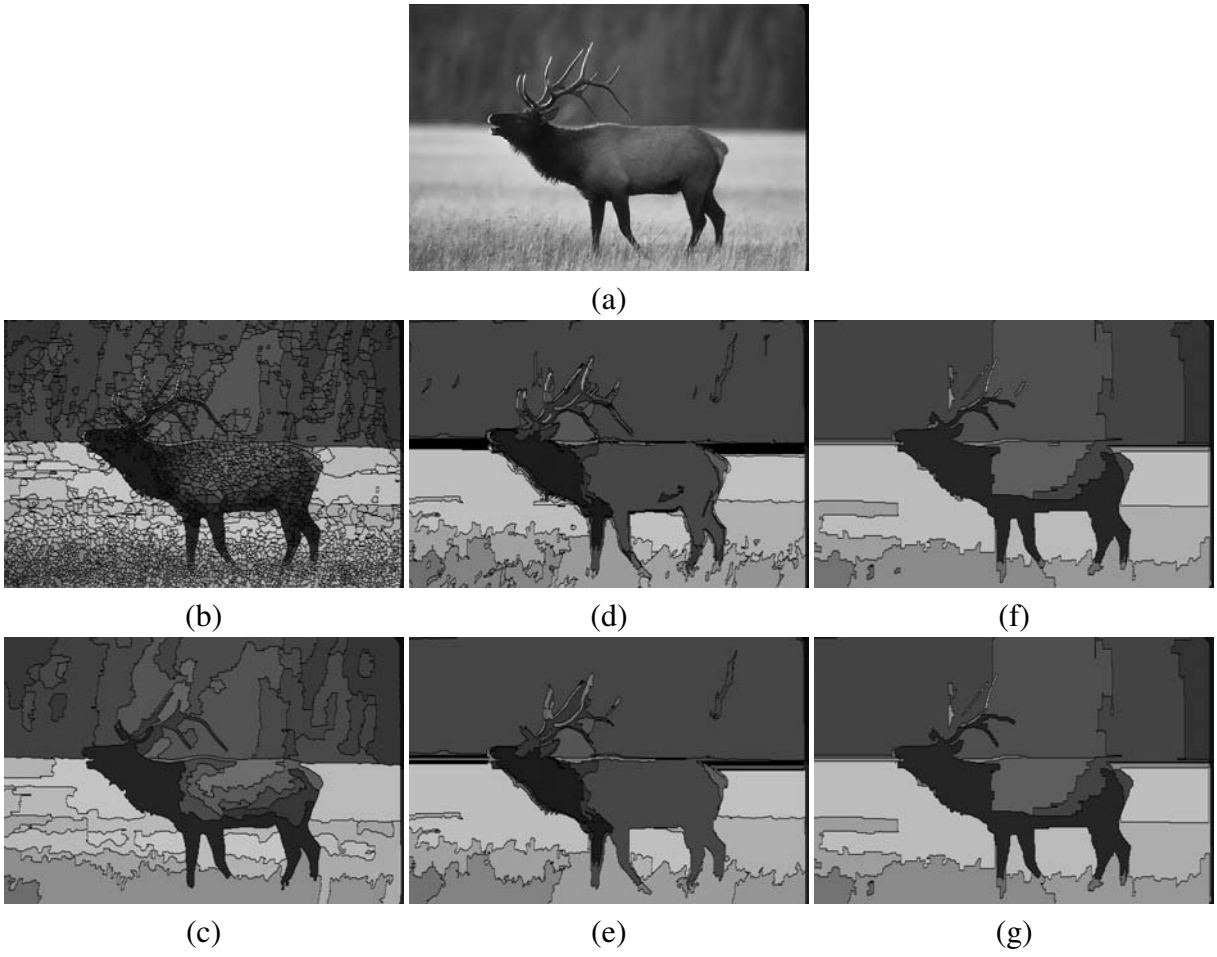


FIG. 2.22 – (a) Image "elan" (b) Segmentation par LPE : 3191 régions (c) Post-filtrage par l'algorithme 4 : 234 régions. (d) Segmentation par EGBIS : 234 régions (e) Post-filtrage par l'algorithme 4 : 50 régions. (f) Segmentation par SRM : 44 régions (g) Post-filtrage par l'algorithme 4 : 36 régions.

– les candidats analysés sont obtenus via une heuristique d'exploration largement sous-optimale, mais fortement dirigée par les données.

La solution proposée consiste à mesurer empiriquement, en générant des images suivant le modèle *a contrario*, quelles valeurs peuvent être obtenues par hasard pour les variables discriminantes. On peut ensuite garantir la même propriété d' ε-fiabilité que dans le cas purement analytique en cherchant les seuils qui garantissent, en moyenne, moins de ε détection dans des images de bruit pur. Cette approche de Monte-Carlo nécessite toutefois quelques conditions sur la *PFA* choisie et sur les heuristiques. En effet, il n'est généralement pas possible de simuler tous les types d'images suivant le modèle *a contrario*, ni de générer des images qui ne suivraient que partiellement le modèle *a contrario*, ce qui a conduit aux conditions de la proposition 3 :

– indépendance de la *PFA* vis-à-vis de la distribution de niveau de gris. Cette propriété est essentielle pour ne pas avoir à calculer des seuils adaptés à toutes les distributions possibles ;
– indépendance également de l'heuristique vis-à-vis de la distribution de niveau de gris, pour les mêmes raisons ;
– indépendance de l'heuristique vis-à-vis de l'entourage d'un candidat : il ne doit pas être moins probable pour l'heuristique de trouver un candidat avec de fortes valeurs discriminantes s'il est entouré de candidats suivant le modèle *a contrario* que s'il est entouré de candidats suivant un modèle quelconque.

Sous ces conditions, moins restrictives, nous avons pu développer une méthodologie *a contrario* de post-filtrage de segmentations, capable de s'adapter à une grande classe d'heuristiques d'exploration et de mesures de différence. Au final, l'algorithme ne requiert aucun *a priori* quantitatif, et donc aucun autre paramètre que la fiabilité désirée. Le seul *a priori* est de nature qualitative : nous considérons que les différences entre deux régions sont d'autant plus significatives qu'elles sont improbables dans un modèle où les pixels sont indépendants et donc spatialement non structurés. Les résultats expérimentaux sont satisfaisants, appliquée en post-filtrage d'algorithmes de segmentation existants, notre méthodologie permet effectivement de filtrer les fausses alarmes, tout en conservant la quasi-totalité des détails. Ceci a été validé pour trois algorithmes différents.

Diverses améliorations peuvent toutefois être envisagées. Premièrement, des heuristiques d'exploration plus évoluées devraient amener de meilleurs résultats que notre heuristique gloutonne. En particulier, le critère de décision proposé est adapté à des procédures multi-échelle, puisque deux régions significativement différentes peuvent également être significativement différentes d'une autre région adjacente une fois regroupées. L'algorithme 4 est purement ascendant et cherche à préserver le maximum de détails en ne regroupant jamais deux régions significativement différentes. Il serait cependant possible de continuer le processus de regroupement de régions, et de construire un arbre contenant les régions significativement différentes à des échelles de plus en plus grandes. De plus, en comparant la significativité des couples de régions, il deviendrait alors possible d'étendre la notion de maximalité des travaux initiaux de [DMM00b] et de déterminer automatiquement l'échelle d'analyse la plus significative pour un ensemble de régions.

Enfin, la gamme des phénomènes physiques de la scène pris en compte dans la segmentation dépend de la capacité des mesures de distance à les capturer. Il pourrait par exemple être plus

judicieux de choisir des mesures insensibles aux gradients d'illumination pour certaines applications. D'autres mesures pourraient également être ajoutées, comme des critères de convexité ou de régularité de la forme des régions. Des mesures adaptées aux images couleurs seraient également intéressantes et facilement envisageables.

Chapitre 3

Apprentissage *a contrario* haut niveau à partir d'images naturelles

3.1 Introduction

Les deux applications précédentes ont abordé des primitives de bas/moyen niveau. Elles partagent comme point commun de travailler directement au niveau pixellique, et dans les deux cas le modèle *a contrario* utilisé portait donc directement sur les pixels eux-mêmes en les considérant indépendants et identiquement distribués. Ainsi, dans ces applications, l'apprentissage *a contrario* s'est appuyé sur des simulations d'images de bruit blanc. Nous envisageons maintenant une problématique de plus haut niveau tirant partir d'un apprentissage *a contrario* à partir d'images naturelles : la détection d'objets individuels (par opposition à la détection de catégories). Étant donné une base de données d'objets, représentés chacun par une seule photo, il s'agit de détecter et de localiser dans une nouvelle image les objets de cette base. Nous prenons comme hypothèse que les objets sont rigides ou presque. La difficulté principale provient des changements d'aspect que les objets peuvent subir, liés notamment aux occultations, aux changements d'illumination, de point de vue, d'échelle, etc. Un exemple est donné dans la figure 3.1.

La détection d'objets est une tâche importante en vision par ordinateur, qui a diverses applications pratiques, par exemple en robotique pour interagir avec des objets [EKJ07] ou pour se localiser [DRMS07, AFDMar], dans un but touristique en fournissant des informations contextuelles sur l'environnement [BFVG06, WWWC04, FSP06], ou encore pour rechercher des images par le contenu [VT02].

Les premières méthodes de détection et de reconnaissance d'objets étaient basées sur des comparaisons directes entre les images des objets à rechercher et les différents emplacements possibles dans l'image à analyser. Parmi les mesures de similarité usuelles, on trouve la corrélation croisée, les sommes des différences au carré ou plus récemment des distances plus évoluées comme celle de Hausdorff [HLO99]. Ces techniques ne sont cependant pas adaptées à

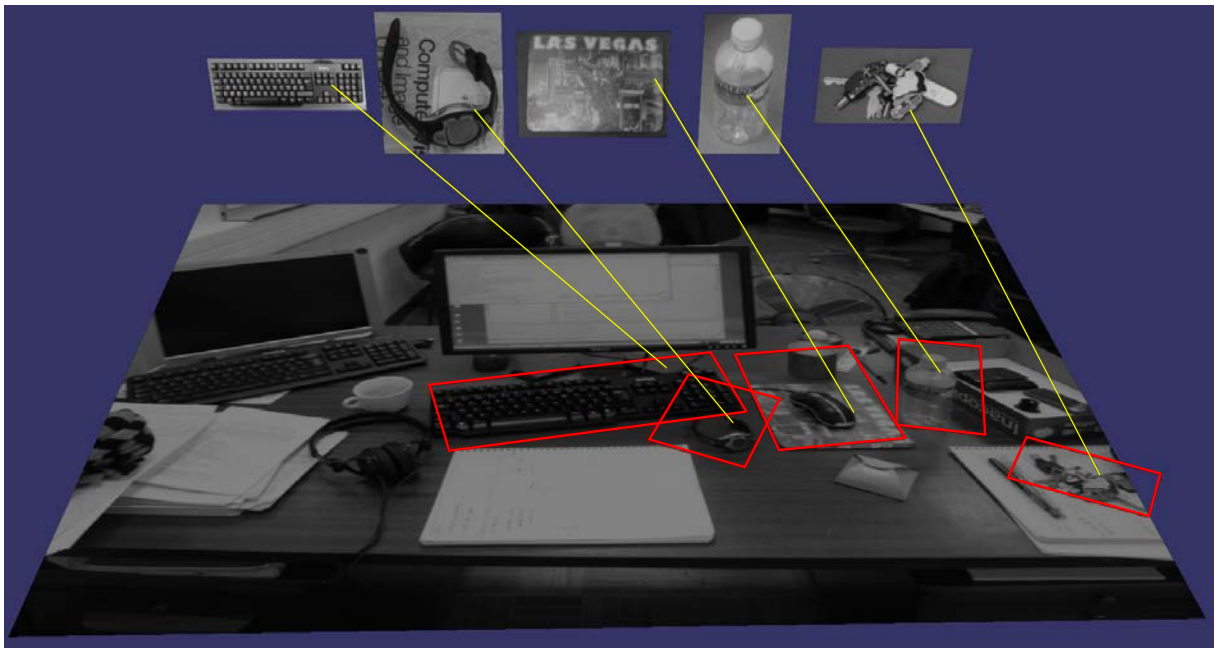


FIG. 3.1 – Détection d'objets dans une image à partir d'une base de photos. Plusieurs sources de variations entre le modèle de la base et son apparition dans une scène s'ajoutent pour rendre la tâche difficile. On remarquera en particulier les changements d'illumination sur le clavier, le changement de fond pour la montre, les occultations pour le tapis de souris, les changements de point de vue pour la bouteille, ou encore les déformations d'objets non rigides pour le trousseau de clés. Il serait bien entendu nécessaire de disposer de plusieurs vues de chaque objet pour gérer les changements drastiques de point de vue, ou de disposer de modèles adaptés pour les objets non rigides. Notre ambition est toutefois d'assurer une certaine robustesse vis-à-vis de ces variations, dans la limite du possible lorsque l'on ne dispose que d'une seule image.

de grandes bases d'objets car les positions possibles pour chacun des objets de la base doivent être analysées une par une dans l'image. De plus, malgré de récents progrès (e.g. [PW08]), elles restent difficilement capables de prendre en compte des occultations ou des changements d'illumination trop importants.

Ces limitations expliquent que depuis une dizaine d'années un large consensus se soit constitué autour des méthodes basées sur la mise en correspondance de caractéristiques locales, plus efficaces et généralement plus robustes. L'algorithme classique est le suivant : pour chaque objet de la base, des points ou des zones d'intérêt sont choisis, par exemple des coins ou des maxima locaux, et une signature locale (appelée aussi descripteur) est calculée autour de chacune de ces zones et stockée dans la base. Ensuite, pour trouver les objets dans une nouvelle image, il suffit de lui appliquer le même algorithme d'extraction de zones d'intérêt et de calcul de signatures locales, puis de rechercher, pour chaque zone, si une signature est suffisamment proche dans la base. En regroupant les mises en correspondance cohérentes, il est possible de décider si un objet est présent. Les temps de calcul sont considérablement réduits puisqu'ils ne dépendent plus que du nombre de zones d'intérêt analysées, généralement faible, et le stockage des objets dans la base est également considérablement simplifié. La résistance aux occultations est immédiate puisqu'il suffit qu'un nombre suffisant de zones d'intérêt dans les régions non occultées puissent être mises en correspondance.

Ces approches purement locales s'avèrent cependant limitées pour la détection d'objets peu texturés, dans lesquels peu de points ou de zones d'intérêt peuvent être extraits. Les méthodes à base de comparaison globales restent plus adaptées pour ce type d'objet. C'est pourquoi nous nous intéressons dans ce chapitre à la combinaison des deux approches en générant des hypothèses rapidement grâce à des mises en correspondance de caractéristiques locales, puis en complétant ces indices locaux à l'aide d'une mesure globale de corrélation avec la photo de l'objet dans la base. Nous proposons également d'améliorer les taux de détection en utilisant les similarités entre descripteurs, non pas pour décider individuellement de façon précoce si chaque mise en correspondance est pertinente, mais plutôt pour mesurer globalement la significativité des groupes de mises en correspondance compatibles. Afin d'obtenir une méthode sans paramètre et sans *a priori* sur la position et l'apparence des objets dans les images de test, nous combinons ces indices hétérogènes à l'aide d'un raisonnement *a contrario* : l'information portée par une mesure de similarité sera quantifiée à partir de la probabilité d'observer une similarité aussi forte sous l'hypothèse où aucun objet de la base n'est présent dans l'image. Un objet sera donc détecté à partir du moment où les mesures de similarité seront significativement plus grandes que celles qui peuvent être obtenues par hasard. Étant donné la nature des variables discriminantes, il ne sera pas possible de calculer leurs distributions *a contrario* de façon purement analytique. Nous verrons alors qu'il est possible d'apprendre ces distributions de façon robuste à partir de quelques images naturelles.

3.2 Détection d'objet à partir de caractéristiques locales

Nous commençons par un tour d'horizon succinct des algorithmes à base de caractéristiques locales. Ils se divisent généralement en quatre grandes étapes, résumées sur la figure 3.2 :

1. Extraire les zones d'intérêt pour chacun des objets de la base, calculer leurs signatures locales et les stocker dans une base. Cette étape est faite hors-ligne ;
2. Extraire les zones d'intérêt et les signatures locales correspondantes dans l'image à analyser ;
3. Les mettre en correspondance avec les zones de la base en utilisant les signatures stockées, puis sélectionner les mises en correspondance dont les signatures sont suffisamment proches ;
4. Regrouper les mises en correspondance conduisant à des hypothèses compatibles et décider pour chaque hypothèse si les indices locaux sont suffisants pour produire une détection.

Chacune de ces étapes a fait l'objet de nombreux travaux, nous en présentons maintenant les plus significatifs.

3.2.1 Extraction de zones d'intérêts et calcul de signatures locales

L'avancée la plus significative dans ce domaine est probablement apparue avec les points SIFT de David G. Lowe [Low99, Low04]. Ces points correspondent à des extrema locaux de l'espace-échelle, calculés par des différences de gaussiennes. Les signatures locales sont des histogrammes des orientations de gradient dans les zones entourant le point, comme le montre la figure 3.3. Les signatures sont calculées à partir d'un voisinage 16x16 autour du point, divisé en 4x4 sous-fenêtres carrées. Un histogramme quantifié sur 8 orientations est calculé pour chaque fenêtre, ce qui donne un descripteur à $4 \times 4 \times 8 = 128$ dimensions. Pour assurer une invariance par rotation, l'orientation principale du point est utilisée comme référence pour le calcul des histogrammes. Ce descripteur s'avère robuste aux changements d'illumination et aux transformations affines légères. Il est également très discriminant pour les zones suffisamment texturées, il est possible de correctement retrouver un point dans une base de plusieurs dizaines de milliers de descripteurs avec un bon taux de réussite.

Beaucoup d'autres méthodes d'extraction de zones d'intérêt et de calcul de signatures ont été proposées. Parmi les les plus populaires, on notera le détecteur de Harris [HS88], celui de Lindeberg [Lin94] ou celui de Kadir et Brady [KZB04]. Matas et al. [MCUP04] proposent d'extraire des régions d'intérêts correspondant à des régions plus claires ou plus sombres que leur voisinage. Pour le calcul de signatures locales, on notera les filtres de Freeman et Adelson [FA91], les "shape context" de Belongie et al. [BMP02], ou encore des variantes de SIFT comme PCA-SIFT [KS04] ou GLOH [MS05]. Des études détaillées de leurs performances respectives pour la détection d'objets sont proposées dans [MS05] et [MP07]. Il apparaît dans ces études comparatives que le descripteur SIFT reste un des meilleurs choix dans la plupart des cas, c'est donc celui que nous retenons.

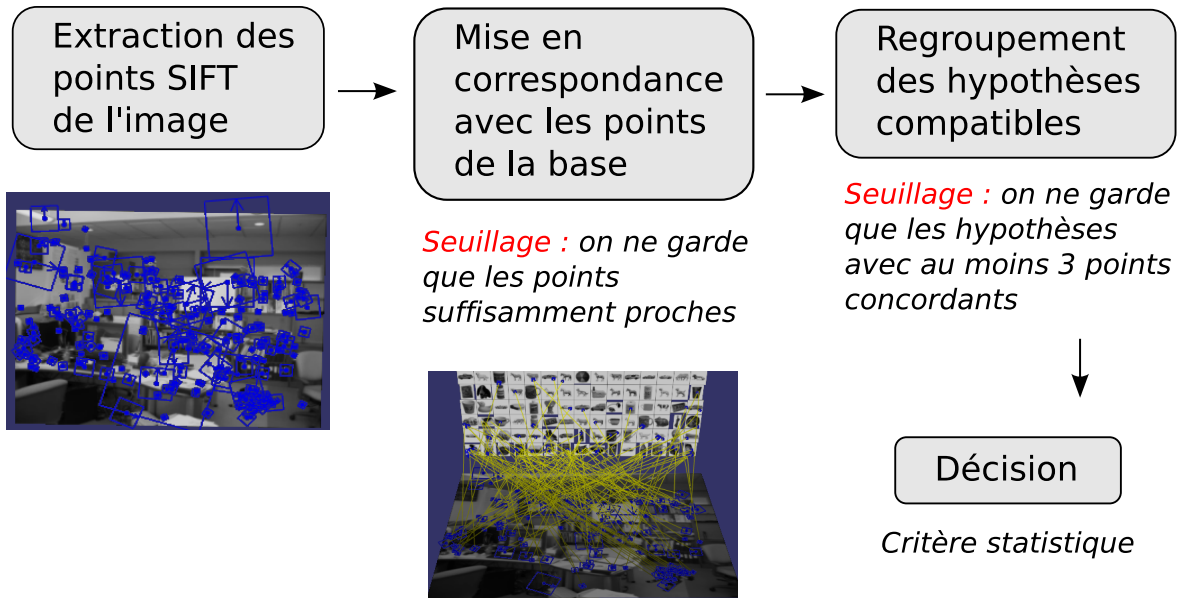


FIG. 3.2 – Algorithme de détection d'objet à base de mise en correspondance de caractéristiques locales SIFT proposé dans [Low04]. Dans une première étape hors-ligne (non représentée), des points d'intérêts sont extraits des images de la base de données et leurs descripteurs sont stockés. Pour détecter les objets de la base dans une image inconnue, on commence par extraire les points d'intérêt de l'image avec le même algorithme que pour les objets de la base. Chaque point de l'image est alors associé au point le plus proche de la base en se basant sur les similarités entre descripteurs. Seules les associations pour lesquelles la distance entre descripteurs est suffisamment faible sont conservées. Les mises en correspondance conduisant à des hypothèses compatibles sur la présence d'un objet sont ensuite regroupées, et une dernière étape décide de la significativité des hypothèses en se basant sur le nombre d'associations de points concordantes.

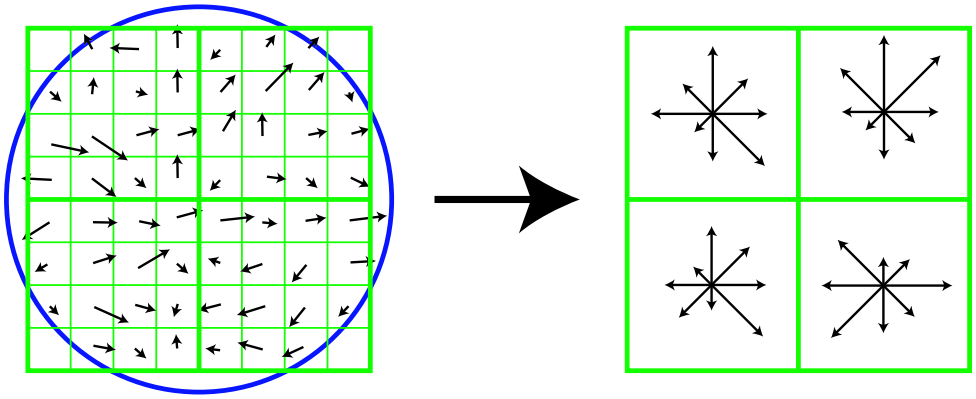


FIG. 3.3 – Descripteur SIFT (figure extraite de [Low04]). Les orientations et les intensités des gradients sont calculées dans le voisinage du point d'intérêt (à gauche). Les intensités sont pondérées par une fenêtre gaussienne (indiquée par le cercle bleu). Le voisinage est ensuite divisé en fenêtres, et un histogramme des orientations (quantifiées en 8 valeurs) est calculé pour chaque fenêtre en accumulant les intensités de gradients pour chaque orientation (à droite). Ainsi, dans chaque fenêtre, la longueur d'une flèche correspond à la somme des intensités des gradients dont l'orientation coïncide. La figure montre ici un descripteur basé sur un voisinage 8x8 divisé en 2x2 sous-fenêtres, alors que le descripteur complet proposé par [Low04] est calculé sur un voisinage 16x16 divisé en 4x4 sous-fenêtres.

3.2.2 Mise en correspondance de points SIFT

À cette étape, le but est de trouver pour chaque point SIFT de l'image si un point de la base lui correspond en comparant leurs signatures locales. Il faut pour cela choisir une mesure de distance entre descripteurs. [Low04] propose une simple distance euclidienne. Des distances plus évoluées ont été proposées, par exemple la Earth Mover Distance (EMD) [LO07] ou la distance du χ^2 [ZMLS07], si l'on considère que les écarts entre points similaires sont distribués selon une loi normale. En pratique, la distance EMD est la plus puissante, mais sa complexité est quasi-quadratique. Les distances euclidiennes et du χ^2 ont en revanche une complexité linéaire, et l'étude de [ZMLS07] ainsi que nos expérimentations montrent que celle du χ^2 est la plus robuste des deux. Suivant le même raisonnement que [FL07], c'est donc celle que nous choisissons. Pour deux descripteurs SIFT u et v de dimension 128, elle est définie par :

$$D_{\chi^2}(u,v) = \sum_{i=1}^{128} \frac{(u_i - v_i)^2}{u_i + v_i}$$

La distance étant choisie, il reste à définir une stratégie pour associer les points de l'image avec les points de la base. L'approche la plus simple consiste à rechercher pour chaque point de l'image si un point de la base est suffisamment proche, en seuillant la mesure de distance entre descripteurs. Tous les descripteurs n'ayant pas le même pouvoir discriminant, il est difficile de choisir un seuil globalement satisfaisant. Un meilleur critère a été proposé par [Low04], en seuillant le ratio des distances entre d'une part le point de l'image et son plus proche voisin dans

la base, et d'autre part le point de l'image et son deuxième plus proche voisin. Cette approche suppose que chaque point de l'image ne peut correspondre qu'à un seul point de la base, et donc que les objets ne sont présents qu'une seule fois dans la base. Le ratio permet alors de prendre en compte de façon approximative la densité de descripteurs de la base similaires au point de l'image, et donc le pouvoir discriminant du point. Cependant, comme le signale [RGD07], ce ratio reste une approximation qui rend la méthode moins sensible pour des objets avec des motifs répétitifs, et le choix du seuil optimal sur le ratio dépend toujours de la base utilisée. Pour ces raisons, [RGD07] propose d'utiliser un raisonnement *a contrario* prenant en compte la complexité de la base entière pour sélectionner les mises en correspondance valides.

Nous pensons cependant qu'il n'est pas souhaitable de prendre de décision à ce niveau là. En effet, il est difficile de prendre une décision fiable en se basant sur un seul point. La pertinence d'une mise en correspondance pourra être évaluée avec beaucoup plus de confiance une fois qu'elle sera regroupée avec les autres associations compatibles, ou bien en s'appuyant sur des mesures globales. L'intérêt principal d'un pré-filtrage à ce niveau est le gain de temps lié à la diminution du nombre d'associations à analyser par la suite. Nous verrons cependant dans le chapitre 4 qu'une architecture adaptée permet de s'affranchir de ces limitations. C'est pourquoi nous nous contentons à cette étape d'associer chaque point de l'image avec son plus proche voisin dans la base. Des algorithmes très efficaces ont été développés dans ce but, notamment celui de [BL97] basé sur un partitionnement en kd-tree de l'espace des descripteurs, et celui de [ACV07] qui est une adaptation du Local Sensitive Hashing de [GIM99] aux descripteurs SIFT. Nous avons choisi ce dernier car [ACV07] a montré sa supériorité pour de grandes bases.

3.2.3 Regroupement des associations compatibles

À chaque point SIFT de l'image sont associées sa position dans l'image, son orientation principale, et l'échelle à laquelle il a été extrait. À chaque point SIFT de la base sont également associées ces informations, relatives à l'objet correspondant. Ces informations sont suffisantes pour estimer la pose d'un objet de la base dans l'image à partir d'une seule mise en correspondance de deux points SIFT, comme le montre la figure 3.4. En regroupant les mises en correspondance qui estiment des poses cohérentes, il est possible d'obtenir une confiance beaucoup plus forte dans la présence d'un objet.

Différentes méthodes ont été proposées pour cette étape de regroupement [WR97, LW88]. Une des plus utilisées est celle de [Low04] qui s'appuie sur une transformée de Hough généralisée [Bal81]. Le principe consiste à projeter chaque mise en correspondance dans un espace de pose à 4 dimensions : la position 2d (x, y) supposée de l'objet de la base dans l'image, son orientation θ et son échelle σ. Ensuite, cet espace est quantifié en cellules, puis chaque mise en correspondance est ajoutée à la cellule correspondante. Il suffit ensuite d'analyser les cellules pour déterminer combien d'associations sont compatibles avec chaque pose. Étant donné la grande dimension de l'espace des poses, la plupart des cellules seront vides. Une implantation à base de tables associatives permet alors de ne parcourir que les cellules non vides et donc de détecter les groupes de mises en correspondance compatibles très efficacement.

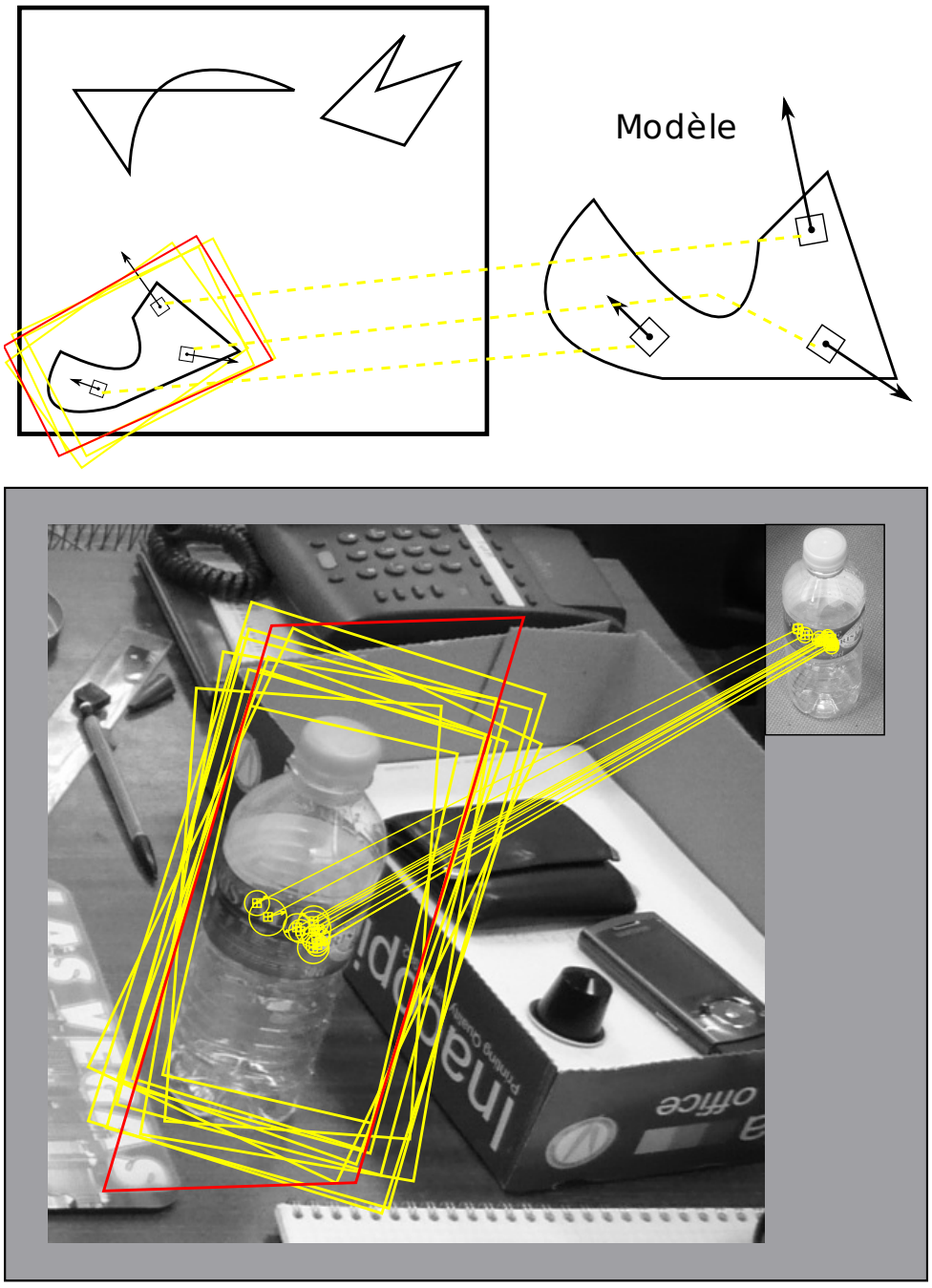


FIG. 3.4 – Chaque point SIFT embarque une information d'orientation principale (direction de sa flèche), d'échelle (longueur de sa flèche) et d'emplacement par rapport à l'objet. Chaque mise en correspondance d'un point SIFT de l'image avec un point SIFT de la base mène donc à une hypothèse complète de pose de l'objet dans l'image, matérialisée par un rectangle jaune. En regroupant les estimations de pose compatibles, il est possible de détecter les objets présents de façon robuste. La pose finale de l'objet peut finalement être estimée en calculant la transformation affine minimisant l'erreur de positionnement de chaque point (en rouge, voir section 3.2.4).

Le principal défaut de cette méthode provient de la quantification de l'espace des poses. Les poses sont représentées par une transformation de similarité à 4 paramètres, qui n'est qu'une approximation de la transformation complète à 6 degrés de liberté que peut subir un objet dans une scène 3D. Il est donc nécessaire de quantifier l'espace des poses en cellules larges, et donc moins précises. Des effets de bords peuvent également apparaître pour les poses situées à la limite entre deux cellules, [Low04] propose donc d'ajouter chaque association dans sa cellule et dans les cellules voisines les plus proches, ce qui diminue encore la précision. Cependant, en pratique, si le même objet est peu susceptible d'apparaître plusieurs fois dans l'image à des positions proches, ce manque de précision dans l'étape de groupement n'est pas très gênant. De plus, la pose finale de l'objet pourra ensuite être estimée de façon plus précise avec une transformation affine en utilisant les positions relatives des points SIFT mis en correspondance.

Si cette quantification est toutefois problématique, les associations compatibles peuvent être regroupées dans l'espace de pose en utilisant des algorithmes de regroupements usuels [DHS01, PNSK05]. Il existe cependant peu de méthodes capables de détecter des groupements de façon robuste sans connaître *a priori* le nombre de groupes. Une avancée significative a été obtenue dans [CDD$^+$07] à l'aide d'un raisonnement *a contrario*, qui permet de retrouver les groupements les plus significatifs sans paramètres. Toutefois, ces méthodes requièrent généralement une densité de points concordants plus importante et sont plus coûteuses en temps de calcul. Pour la détection d'objet, il est fréquent de devoir détecter des groupes de seulement deux ou trois associations compatibles. De plus, les méthodes de regroupement globales s'appuient sur l'ensemble des mises en correspondance pour déterminer quels sont les groupements significatifs, ce qui rend impossible toute détection avant que tous les points SIFT aient été associés. Nous verrons dans le chapitre 4 que cette propriété empêche d'obtenir un algorithme de détection incrémental.

Pour ces raisons, nous conservons la méthode à base de transformée de Hough de [Low04] pour identifier les groupes de correspondances compatibles. Il reste maintenant à évaluer la significativité des hypothèses générées par ces groupes.

3.2.4 Estimation de la pose finale de l'objet

Chaque mise en correspondance fournit une estimation de la pose de l'objet de la base dans l'image via une transformation de similarité à quatre paramètres : orientation, échelle et position 2d. Pour estimer de façon plus fine la pose de l'objet dans l'image, [Low04] calcule la transformation affine minimisant l'erreur de projection pour chacun des points SIFT appariés dans l'image et dans le modèle. Cette estimation est généralement fiable à partir de trois points. Avec les critères de significativité que nous proposons dans la section 3.3, la reconnaissance peut être effectuée à partir d'une seule mise en correspondance, si les descripteurs sont très proches et si la similarité globale est forte. S'il y a une seule mise en correspondance, nous prenons directement la transformation de similarité estimée comme transformation finale, et s'il y a deux mises en correspondance, nous prenons la moyenne des deux transformées de similarité estimée. Au-delà, nous utilisons la méthode de [Low04].

3.3 Mesure *a contrario* de la significativité d'une hypothèse

La mise en correspondance de caractéristiques locales permet de générer efficacement des hypothèses de présence d'un objet. Il reste maintenant à décider, pour chaque hypothèse, si elle correspond réellement à un objet de la base présent ou non. [Low04] et [CDD+07] reposent uniquement sur le nombre de mises en correspondance compatibles pour décider si un groupement est significatif, et donc s'il correspond à un objet. [MMP04] propose un modèle probabiliste plus complet intégrant également les degrés de similarité entre les descripteurs. Il s'agit d'un modèle bayésien basé sur la probabilité *a posteriori* de présence d'un objet : un modèle de similarité doit être fourni pour les associations accidentelles et un autre pour les associations correctes. Des modèles gaussiens sont proposés, dont les paramètres sont estimés à partir de la base d'objets.

La modélisation des similarités entre descripteurs dans le cas où un objet est présent est difficile et n'a réellement de sens que si l'algorithme est utilisé dans un environnement relativement contrôlé, dans lequel un *a priori* sur l'apparence des objets est pertinent. Si les objets sont susceptibles d'apparaître avec une grande diversité de variations, comme c'est généralement le cas dans les scènes réelles, nous proposons de recourir uniquement à un modèle des similarités accidentelles en reposant sur un raisonnement *a contrario*.

Pour une application de détection d'objets, l'hypothèse *a contrario* naturelle est qu'aucun objet de la base n'est présent dans l'image à analyser. En modélisant les similarités qui peuvent apparaître accidentellement dans ce cas entre des zones de l'image et les objets de la base, les objets présents peuvent alors être détectés en recherchant les similarités qui sont statistiquement trop grandes pour être le résultat de l'hypothèse *a contrario*. Ainsi, les objets seront détectés dans la scène sans *a priori* quantitatif sur leur apparence, il suffit que leur similarité avec un objet de la base soit significativement plus grande que les similarités observées par hasard.

Il reste à choisir les variables discriminantes sur lesquelles le raisonnement *a contrario* doit s'appuyer. Les hypothèses de présence d'un objet sont générées par des groupements de mises en correspondance compatibles de points SIFT. Suivant [Low04], il est naturel d'utiliser comme première variable discriminante le nombre d'associations compatibles. Lorsque les objets sont petits dans la scène ou peu texturés, il est parfois nécessaire de détecter des objets à partir de 2 ou 3 associations correctes. De plus, nous avons décidé dans la section 3.2.3 de ne pas filtrer *a priori* les mises en correspondance, il est donc assez fréquent d'obtenir accidentellement plusieurs associations compatibles. Pour augmenter le pouvoir discriminant, nous ajoutons une deuxième variable mesurant le degré de similarité entre les points SIFT associés grâce à la moyenne des similarités entre descripteurs. Ces deux variables peuvent malgré tout s'avérer insuffisantes pour les objets peu texturés et de petite taille ayant subi de fortes variations, car peu de points SIFT seront présents et les différences entre descripteurs peuvent rester relativement importantes. Pour pallier ce problème, nous rajoutons une troisième variable discriminante, globale, basée sur un calcul de corrélation. Nous détaillons maintenant le calcul des distributions *a contrario* de ces trois variables puis leur combinaison.

3.3.1 Significativité basée sur le nombre d'associations compatibles

Une hypothèse de présence d'un objet est créée pour chaque regroupement de mises en correspondance compatibles de points SIFT. Chaque hypothèse $\mathcal{H}$ correspond donc à une cellule de l'espace de Hough. Nous notons $N_v(\mathcal{H})$ la variable aléatoire qui désigne le nombre d'associations de points SIFT ajoutées dans la cellule de l'hypothèse $\mathcal{H}$. Plus N_v est grande, plus il y a de chance que l'objet de la base concerné soit effectivement présent. En raisonnant *a contrario*, il nous faut calculer la probabilité d'obtenir une valeur aussi grande pour N_v sous l'hypothèse H_0 où aucun objet de la base n'est présent dans l'image. [Low01] propose un calcul analytique de cette distribution, que nous reprenons à l'identique dans un premier temps. Ce calcul repose tout d'abord sur la probabilité qu'une association A_i de deux points SIFT tombe accidentellement dans la cellule de $\mathcal{H}$. Il s'agit de la probabilité jointe que le point de la base appartienne à l'objet M concerné par $\mathcal{H}$, et que l'orientation θ, l'échelle σ et la position $\Delta = (x, y)$ estimées par A_i soient égales à celles de $\mathcal{H}$ après quantification. Nous notons p_{pose} cette probabilité :

$$p_{pose}(\mathcal{H}) = P_{H_0}(\theta = \theta(\mathcal{H}), \sigma = \sigma(\mathcal{H}), \Delta = \Delta(\mathcal{H}), M = M(\mathcal{H}))$$

p_{pose} dépend de la quantification de l'espace de Hough : plus il est grossier, plus il est probable qu'une association tombe par hasard dans la cellule de $\mathcal{H}$. Nous utilisons la même quantification que [Low04] :

- l'orientation supposée de l'objet est divisée en 12 cellules de 30 degrés ;
- l'échelle supposée est quantifiée par des puissances entières de 2 ;
- la position supposée de l'objet dans l'image est quantifiée par pas de $0.2 \times L_{max}$ avec L_{max} le maximum entre la hauteur et la largeur du modèle de la base à l'échelle estimée.

[Low01] considère alors que les différentes dimensions sont indépendantes sous H_0 :

$$p_{pose}(\mathcal{H}) = P_{H_0}(\theta = \theta(\mathcal{H})) \times P_{H_0}(\sigma = \sigma(\mathcal{H})) \times P_{H_0}(\Delta = \Delta(\mathcal{H})) \times P_{H_0}(M = M(\mathcal{H}))$$

Puis, les orientations et la position des objets sont considérées uniformément distribuées sous H_0 :

$$P_{H_0}(\theta = \theta(\mathcal{H})) = \frac{1}{12}$$
$$P_{H_0}(\Delta = \Delta(\mathcal{H})) = 0.2 \times 0.2$$

La quantification des échelles étant peu précise et la plupart des points SIFT étant détectés à de petites échelles, $P_{H_0}(\sigma = \sigma(\mathcal{H}))$ est empiriquement estimée à $\frac{1}{2}$. Enfin, la probabilité qu'un point SIFT de l'image soit associé à un point SIFT de l'objet $M(\mathcal{H})$ lorsque cet objet n'est pas présent dans l'image est calculée par le rapport entre le nombre de points SIFT de l'objet dans la base et le nombre de points SIFT total dans la base :

$$P_{H_0}(M = M(\mathcal{H})) = \frac{\textit{nombre de points SIFT de } M(\mathcal{H}) \textit{ dans la base}}{\textit{nombre total de points SIFT dans la base}}$$

$p_{pose}(\mathcal{H})$ donne une estimation de la probabilité qu'une mise en correspondance de points SIFT soit compatible par hasard avec la pose de l'hypothèse $\mathcal{H}$. [Low01] en déduit alors la distribution du nombre de correspondances N_v en fonction du nombre $N_z(\mathcal{H})$ de points SIFT de l'image potentiellement compatibles avec $\mathcal{H}$. N_z correspond au nombre de points SIFT de l'image susceptibles de donner lieu à une mise en correspondance compatible avec $\mathcal{H}$. Il est calculé en projetant le modèle de la base dans l'image à l'aide de la transformation de similarité associée à la cellule de $\mathcal{H}$ puis en comptant le nombre de points SIFT dans la zone délimitée par la projection. En supposant que les mises en correspondance sont indépendantes sous H_0, la probabilité que N_v soit aussi grand que $N_v(\mathcal{H})$ est donnée par la probabilité qu'au moins $N_v(\mathcal{H})$ points SIFT parmi les $N_z(\mathcal{H})$ points potentiels tombent dans la cellule de $\mathcal{H}$, ce qui donne :

$$P_{H_0}^{Lowe}(\mathcal{H}) = P_{H_0}^{Lowe}(N_v \geq N_v(\mathcal{H}) \mid N_z = N_z(\mathcal{H})) = \mathcal{B}_{\geq}(N_v(\mathcal{H}), N_z(\mathcal{H}), p_{pose}(\mathcal{H}))$$

avec $\mathcal{B}_{\geq}$ la fonction de répartition complémentaire de la loi binomiale.

$P_{H_0}^{Lowe}(\mathcal{H})$ est une bonne estimation de la distribution du nombre de mises en correspondance accidentellement compatibles avec une hypothèse $\mathcal{H}$ quand les associations sont peu nombreuses et espacées, ce qui peut être le cas si les associations sont sévèrement filtrées au préalable comme dans [Low04]. Dans notre cas, les associations ne sont pas sélectionnées, et si un point SIFT de l'image est proche d'un point SIFT de la base, il est fréquent que d'autres points SIFT très proches spatialement et donc ayant une signature très similaire s'associent également avec les points SIFT voisins de l'objet dans la base. Les descripteurs de ces points SIFT sont calculés dans des zones largement superposées, et ces mises en correspondance ne peuvent donc pas être considérées indépendantes, même sous l'hypothèse H_0 où aucun objet de la base n'est présent.

Ces dépendances sont très difficiles à modéliser de façon analytique, c'est pourquoi nous allons recourir à un apprentissage à partir d'images naturelles, comme détaillé en section 3.5. Une première option consisterait à apprendre directement $P_{H_0}(N_v \geq N_v(\mathcal{H}))$. Pour obtenir une estimation aussi adaptative que $P_{H_0}^{Lowe}$, il faudrait cependant prendre en compte des quantités telles que le nombre de points SIFT de l'objet dans la base, et le nombre de points SIFT de l'image potentiellement compatibles N_z, qui varie pour chaque hypothèse. Pour rendre l'apprentissage plus simple et plus générique, nous choisissons directement $P_{H_0}^{Lowe}(\mathcal{H})$ comme variable discriminante. Le pouvoir discriminant est toujours tiré de N_v, mais l'apprentissage est nettement simplifié puisqu'il s'agit uniquement d'apprendre l'influence des phénomènes non pris en compte par le calcul analytique $P_{H_0}^{Lowe}(\mathcal{H})$, comme la dépendance entre les points SIFT proches.

3.3.2 Significativité basée sur la force des associations compatibles

La variable $P_{H_0}^{Lowe}(\mathcal{H})$ prend en compte le nombre de mises en correspondance compatibles, mais pas les degrés de similarités entre les descripteurs SIFT. Pour mesurer la confiance dans une association entre un point SIFT de l'image et un point SIFT de la base, nous reprenons ici le ratio des distances tel qu'il a été présenté dans [Low04], qui est un bon compromis

entre simplicité et robustesse. Soit $D(k, NN_1(k))$ la distance du χ^2 entre un point SIFT k de l'image et le descripteur SIFT le plus proche dans la base noté $NN_1(k)$ (Nearest Neighbor 1), et $D(k, NN_2(k))$ la distance avec le second point le plus proche dans la base, noté $NN_2(k)$. Alors le ratio D_r des distances pour la mise en correspondance du point k est défini par :

$$D_r(k) = \frac{D(k, NN_1(k))}{D(k, NN_2(k))}$$

Ce ratio est compris entre 0 et 1 et, plus il est faible, plus le descripteur est discriminant et plus le point k est susceptible de réellement correspondre au point de la base $NN_1(k)$.

Pour prendre en compte la confiance dans l'ensemble des mises en correspondance d'une hypothèse $\mathcal{H}$, nous introduisons la moyenne des ratios des distances μ :

$$\mu(\mathcal{H}) = \frac{1}{N_v(\mathcal{H})} \sum_{k \in \mathcal{H}} D_r(k)$$

Nous supposons dans un premier temps que les similarités entre descripteurs sont indépendantes sous H_0. Ceci nous permet d'estimer analytiquement la distribution *a contrario* de $\mu(\mathcal{H})$ en fonction de la distribution de D_r en utilisant la même procédure que dans la section 2.2.6. La distribution $P_{H_0}(D_r \leq D_r(k))$ est en revanche difficile à estimer. Comme le note [CDD+07], cette distribution dépend de la base d'objets utilisée, et elle n'est pas estimable analytiquement. Nous allons donc également apprendre cette distribution en section 3.5.

De plus, pour les mêmes raisons que dans la section 3.3.1, les similarités des mises en correspondance ne peuvent pas être considérées totalement indépendantes, même sous H_0. Nous proposons donc la même approche que pour $P_{H_0}^{Lowe}$ en utilisant $P_{H_0}(\mu \geq \mu(\mathcal{H}) \mid N_v = N_v(\mathcal{H}))$ comme variable discriminante, notée $P_{H_0}^{\mu}(\mathcal{H})$, puis en apprenant sa distribution à partir d'images naturelles pour prendre en compte les dépendances.

3.3.3 Extraction du sous-groupe de mises en correspondance le plus significatif

Une hypothèse ne devient pas nécessairement plus significative quand le nombre d'associations compatibles augmente. En effet, des associations très peu significatives avec des ratios de distance D_r très élevés peuvent augmenter la moyenne μ et ainsi faire diminuer la significativité mesurée par $P_{H_0}^{\mu}(\mathcal{H})$. Nous corrigeons ce défaut en insérant une étape de sélection qui détermine le sous-groupe d'associations compatibles le plus significatif. Soit Ω l'ensemble des mises en correspondance compatibles avec $\mathcal{H}$. En utilisant nos variables discriminantes, la significativité d'un sous-groupe $G \subset \Omega$, notée $P_{H_0}(G)$, est donnée par :

$$P_{H_0}(G) = P_{H_0}(P_{H_0}^{\mu} \leq P_{H_0}^{\mu}(G), P_{H_0}^{Lowe} \leq P_{H_0}^{Lowe}(G))$$

$P_{H_0}^{\mu}$ est calculée conditionnellement à N_v, elle peut donc être considérée indépendante de $P_{H_0}^{Lowe}$, et $P_{H_0}(G)$ peut être estimée par :

$$P_{H_0}(G) = P_{H_0}(P_{H_0}^{\mu} \leq P_{H_0}^{\mu}(G)) \times P_{H_0}(P_{H_0}^{Lowe} \leq P_{H_0}^{Lowe}(G))$$

Notre objectif devient alors de déterminer le sous-groupe $G^* \subset \Omega$ qui minimise $P_{H_0}(G)$:

$$G^* = \underset{G \subset \Omega}{\operatorname{argmin}} P_{H_0}(G)$$

G^* peut être déterminé efficacement en utilisant la monotonie de $P_{H_0}^{Lowe}$, qui diminue avec le nombre d'associations, et la monotonie de $P_{H_0}^{\mu}$, qui diminue avec la moyenne μ pour un nombre d'associations donné. La significativité maximale pour un sous-groupe de k associations est ainsi obtenue en choisissant les k associations ayant les ratios de distances les plus faibles. Soit $\Omega = \{A_1, A_2, \ldots, A_{N_v}\}$ l'ensemble des associations triées par ordre croissant de ratio de distances D_r. On note $G_i = \{A_1, \ldots, A_i\}$ l'ensemble des associations de rang inférieur à i. Il y a donc N_v ensembles G_i possibles. G^* est alors donné par :

$$G^* = \underset{G_i}{\operatorname{argmin}} P_{H_0}(G_i)$$

Les valeurs de $P_{H_0}^{\mu}$ et $P_{H_0}^{Lowe}$ sont finalement calculées à partir de G^*.

3.3.4 Significativité basée sur la similarité d'apparence globale

Les deux variables précédentes s'appuient sur des similarités entre caractéristiques locales. Celles-ci sont très discriminantes, mais nécessitent des objets suffisamment gros et texturés pour être suffisamment nombreuses. Même à l'échelle la plus fine, il faut que les voisinages 16x16 des points SIFT puissent être inclus dans l'objet. Les points SIFT mis en correspondance le sont donc le plus souvent à des échelles fines, les points SIFT de trop grande échelle incluant rapidement de l'information provenant du fond autour des objets. Comme le fond peut changer d'une image à l'autre, les descripteurs associés à ces points peuvent devenir très différents : l'orientation des gradients peut changer voire même s'inverser si le nouveau fond est plus clair que l'objet alors que l'ancien était plus foncé, et vice-versa. Ce constat a d'ailleurs conduit [SH05] à proposer une variante de SIFT invariante aux changements de fonds, mais qui nécessite une segmentation *a priori* des objets, envisageable pour les images des modèles, mais généralement hors de portée pour les images de test. Enfin, comme le constate [ZMLS07], les caractéristiques locales sont parfois trop invariantes et en deviennent d'autant moins discriminantes quand les variations sont faibles.

C'est pourquoi nous proposons de rajouter une mesure de similarité globale, adaptée aux tendances générales des objets lisses. Beaucoup de mesures peuvent être utilisées, citons par exemple la distance de Hausdorff comme celle de [HLO99] ou la distance de Hamming évoluée de [PW08], qui propose également une revue détaillée des mesures de similarité rapides. On pourra également se référer à [CC04, CC02] pour une étude comparative des mesures de corrélation plus usuelles, utilisée notamment pour la détection de visages [BP93] ou le suivi d'objets. Nous avons simplement choisi ici une somme des différences absolues (SAD), suffisante pour nous permettre de montrer comment des caractéristiques hétérogènes peuvent être combinées dans un cadre *a contrario* pour améliorer les taux de détection. Une des limitations

principales de ce type de méthode, à savoir le temps de calcul pour parcourir toutes les positions et transformations possibles pour l'objet recherché, est levée ici puisque le modèle de la base et sa pose présumée sont déjà fournis par les regroupements de caractéristiques locales.

Pour limiter les biais liés aux valeurs absolues des intensités de niveau de gris et ajouter une certaine robustesse aux changements globaux d'illumination, nous normalisons les valeurs de niveau de gris de la zone de l'image concernée et de l'image modèle entre 0 et 1 par étirement linéaire des histogrammes locaux, en ignorant 5% des valeurs les plus grandes et 5% des valeurs les plus faibles. Il s'agit d'une méthode de normalisation couramment utilisée, par exemple dans [Fil07]. Ensuite l'image modèle est projetée dans le repère de l'image analysée en utilisant la transformation estimée par les mises en correspondance de points SIFT. Nous projetons l'image du modèle dans le repère de l'image et non l'inverse, car le modèle est généralement plus grand que l'instance observée dans une nouvelle image. Les différences entre pixels sont ensuite additionnées puis normalisées pour obtenir la mesure de SAD finale pour une hypothèse $\mathcal{H}$:

$$D_{sad}(\mathcal{H}) = \frac{\sum_{(x,y)\in I_{\mathcal{H}}} |I(x,y) - I_{\mathcal{H}}(x,y)|}{\sum_{(x,y)\in I_{\mathcal{H}}} 1}$$

avec I l'image analysée, et $I_{\mathcal{H}}$ l'image du modèle de l'hypothèse $\mathcal{H}$ projetée dans le repère de I par la transformation estimée pour $\mathcal{H}$.

D_{sad} constitue notre troisième variable discriminante, dont la distribution *a contrario* doit également être apprise. Si les images des modèles de la base de données incluent une trop grande quantité de fond, il peut être nécessaire de segmenter au préalable les images pour diminuer l'influence du fond dans le calcul de D_{sad}. Il existe de nombreuses techniques capables de segmenter un unique objet prédominant sur un fond homogène, par exemple [XAB07, XM08]. Dans notre cas, les objets utilisés pour l'évaluation dans la section 3.6 ont été photographiés sur un fond noir, et un simple seuillage suffit pour enlever une grande partie du fond.

3.3.5 Combinaison des différentes variables

Nous avons introduit trois variables discriminantes complémentaires : $P_{H_0}^{Lowe}$, $P_{H_0}^{\mu}$ et D_{sad}. Les deux premières peuvent être considérées indépendantes puisque $P_{H_0}^{\mu}$ est calculée conditionnellement au nombre de mises en correspondance analysé par $P_{H_0}^{Lowe}$. En revanche, la similarité globale D_{sad} ne peut être considérée totalement indépendante des similarités locales, mais cette dépendance est très difficile à estimer analytiquement. Nous avons recours ici à la même technique que précédemment en introduisant la variable $P_{H_0}^{*}(\mathcal{H})$, qui approche analytiquement la distribution *a contrario* jointe de ces trois variables discriminantes :

$$P_{H_0}^{*}(\mathcal{H}) = P_{H_0}(P_{H_0}^{Lowe} \leq P_{H_0}^{Lowe}(\mathcal{H})) \times P_{H_0}(P_{H_0}^{\mu} \leq P_{H_0}^{\mu}(\mathcal{H})) \times P_{H_0}(D_{sad} \leq D_{sad}(\mathcal{H}))$$

$P_{H_0}^{*}(\mathcal{H})$ est une approximation qui ne prend pas en compte les dépendances qui peuvent exister, mais elle nous sert de base pour apprendre la probabilité *a contrario* finale pour une hypothèse $\mathcal{H}$:

$$PFA(\mathcal{H}) = P_{H_0}(P_{H_0}^{*} \leq P_{H_0}^{*}(\mathcal{H}))$$

3.4 Prise de décision finale

Pour évaluer notre algorithme de détection d'objet, nous aurons recours à des courbes de type précision/rappel dans la section 3.6 en utilisant directement $PFA(\mathcal{H})$ comme mesure de la confiance dans une hypothèse $\mathcal{H}$ de pose d'un objet. Un seuillage n'est donc pas nécessaire pour l'évaluation. Toutefois, pour une utilisation réelle de l'algorithme de détection, il est intéressant de pouvoir déterminer automatiquement un seuil garantissant l'ε-fiabilité de l'algorithme, en utilisant la même méthode que pour les autres travaux *a contrario*. Puisque chaque point de l'image est associé à un seul point de la base, le nombre d'hypothèses testées pour une image est nécessairement inférieur ou égal au nombre N_p de points SIFT extraits. De plus, la plupart des points ne sont généralement pas regroupés, et le nombre final d'hypothèses testées est très proche de N_p.

On en déduit le critère suivant pour assurer l'ε-fiabilité de l'algorithme de détection (proposition 1) :

$$PFA(\mathcal{H}) < \frac{\varepsilon}{N_p}$$

La validité de ce critère dépend de la validité des distributions apprises, que nous vérifions maintenant expérimentalement.

3.5 Apprentissage des distributions *a contrario*

Les distributions *a contrario* de $P_{H_0}^{Lowe}$, D_r, $P_{H_0}^{\mu}$, D_{sad} et $P_{H_0}^{*}$ ne sont pas calculables analytiquement, et nous proposons de les apprendre à partir d'images naturelles. Ces variables prennent déjà en compte un certain nombre de sources de variabilité, aussi il ne reste à apprendre que des influences plus génériques.

Pour évaluer la robustesse d'un apprentissage à partir d'images naturelles, nous avons mesuré les distributions obtenues à partir de trois ensembles d'apprentissage, le premier contenant des images d'intérieur, le second des images plutôt urbaines (présence de bâtiments, etc.), et le troisième des images plutôt rurales (champs, plage, etc.). Ces images ont été extraites à partir d'un ensemble d'images provenant de diverses sources : la base MIT-CSAIL d'objets et de scènes [TMFR03], la base de test de segmentations de Berkeley [MFTM01], les images de fonds de [SH05] et les images intérieures et extérieures de [FFIKP07]. Dans chaque catégorie, les images d'apprentissage ont été choisies aléatoirement, nous avons simplement vérifié qu'aucun objet similaire à ceux de la base n'était présent dans les images. Chaque image produisant à elle seule plusieurs milliers d'hypothèses de pose et donc d'exemples d'apprentissage, une dizaine d'images suffit en pratique pour obtenir des estimations suffisamment précises des distributions *a contrario*. De plus, la figure 3.5 montre qu'il n'est pas utile de rajouter plus d'images d'apprentissage, car l'influence sur les résultats obtenus est relativement marginale. La figure 3.5 montre finalement les ensembles de 10 images utilisés pour l'apprentissage dans chaque catégorie.

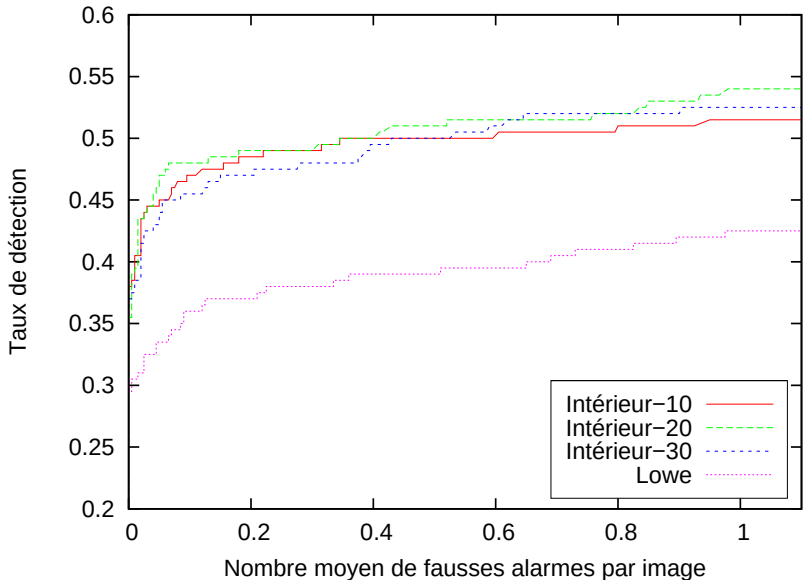


FIG. 3.5 – Influence du nombre d'images de l'ensemble d'apprentissage sur les performances de l'algorithme de détection. Le processus d'évaluation est détaillé dans la section 3.6, il s'agit de détecter chacun des 200 objets d'une base incrustés dans des images de fonds après avoir subi un certain nombre de transformations. Chaque courbe correspond à un ensemble d'apprentissage de respectivement 10, 20 et 30 images d'intérieur et représente le taux d'objets détectés en fonction du nombre de fausses alarmes moyen par image. Les résultats obtenus par l'approche originelle de [Low04] sont également représentés à titre de comparaison. Une dizaine d'images suffit pour estimer les distributions de façon suffisamment précise et les résultats ne s'améliorent pas significativement en rajoutant plus d'images.

La base de données utilisée est quant à elle constituée de 200 objets extraits aléatoirement de la base d'objets rigides ALOI [GBS05]. Plus de détails seront donnés sur cette base dans la section 3.6.

La figure 3.7 montre les distributions obtenues pour les variables $P_{H_0}^{Lowe}$, D_r, $P_{H_0}^{\mu}$, D_{sad} et $P_{H_0}^{*}$ pour les trois ensembles d'apprentissage. La variable la plus sensible aux images d'apprentissage est $P_{H_0}^{Lowe}$ en raison des nombreuses structures linéaires des images d'intérieur qui donnent souvent lieu à de multiples points SIFT proches et similaires. Les dépendances entre points sont donc plus importantes que dans les cas extérieurs. L'ordre de grandeur des variations reste cependant limité. Les distributions de ratio de distances D_r, la moyenne des ratios $P_{H_0}^{\mu}$ et la mesure de similarité globale $P_{H_0}(D_{sad})$ ont en revanche des comportements très similaires dans les trois ensembles d'apprentissage. Au final, la variable $P_{H_0}^{*}$ combinant toutes ces mesures a une distribution relativement indépendante du choix des images d'apprentissage. Pour s'assurer malgré tout que les distributions *a contrario* ne surestiment pas la significativité des différentes variables, nous utiliserons dans la suite l'ensemble d'apprentissage intérieur.

Remarque Les distributions empiriques apprises sont estimées à partir des valeurs observées dans le cas où aucun objet n'est présent. Aussi, les distributions obtenues ne couvrent pas les très petites valeurs des variables discriminantes, trop improbables sous l'hypothèse H_0, mais qui seront — nous l'espérons — observées lorsqu'un objet est présent. Pour obtenir une estimation des distributions pour ces valeurs extrêmes, nous avons recours ici encore à la théorie des grandes déviations, détaillée dans l'annexe C.

3.6 Évaluation

Pour évaluer les performances de notre algorithme de détection, nous exploitons la base d'objets ALOI [GBS05]. Cette base contient 1000 objets variés qui ont été photographiés sous divers points de vue et diverses illuminations sur un fond noir. Pour obtenir des temps de détection raisonnables, nous avons extrait aléatoirement un sous-ensemble de 200 objets pour nos expérimentations, représenté sur la figure 3.8. La base d'objets est constituée des photos de chacun des objets, de face. Pour pouvoir évaluer automatiquement le taux de réussite de l'algorithme, nous avons suivi une méthode similaire à [PL06] en incrustant artificiellement chacun des objets à détecter dans des images de test, après leur avoir appliqué un certain nombre de transformations. La configuration de base est la suivante :

- largeur ou hauteur maximale de l'objet de 100 pixels ;
- point de vue décalé de 25 degrés par rapport au point de vue de la photo dans la base ;
- rotation dans le plan de 15 degrés ;
- pas d'occultation ;
- bruit uniforme de 2% ajouté à l'image finale ;
- placement aléatoire dans l'image.

Les images transformées sont obtenues par interpolation bilinéaire. Les paramètres choisis rendent la détection relativement difficile. La figure 3.9 montre des exemples d'images obtenues

FIG. 3.6 – Ensembles d'apprentissage de 10 images chacun utilisés pour nos expériences. En haut : images d'intérieur. Au milieu : images "urbaines". En bas : images "rurales".

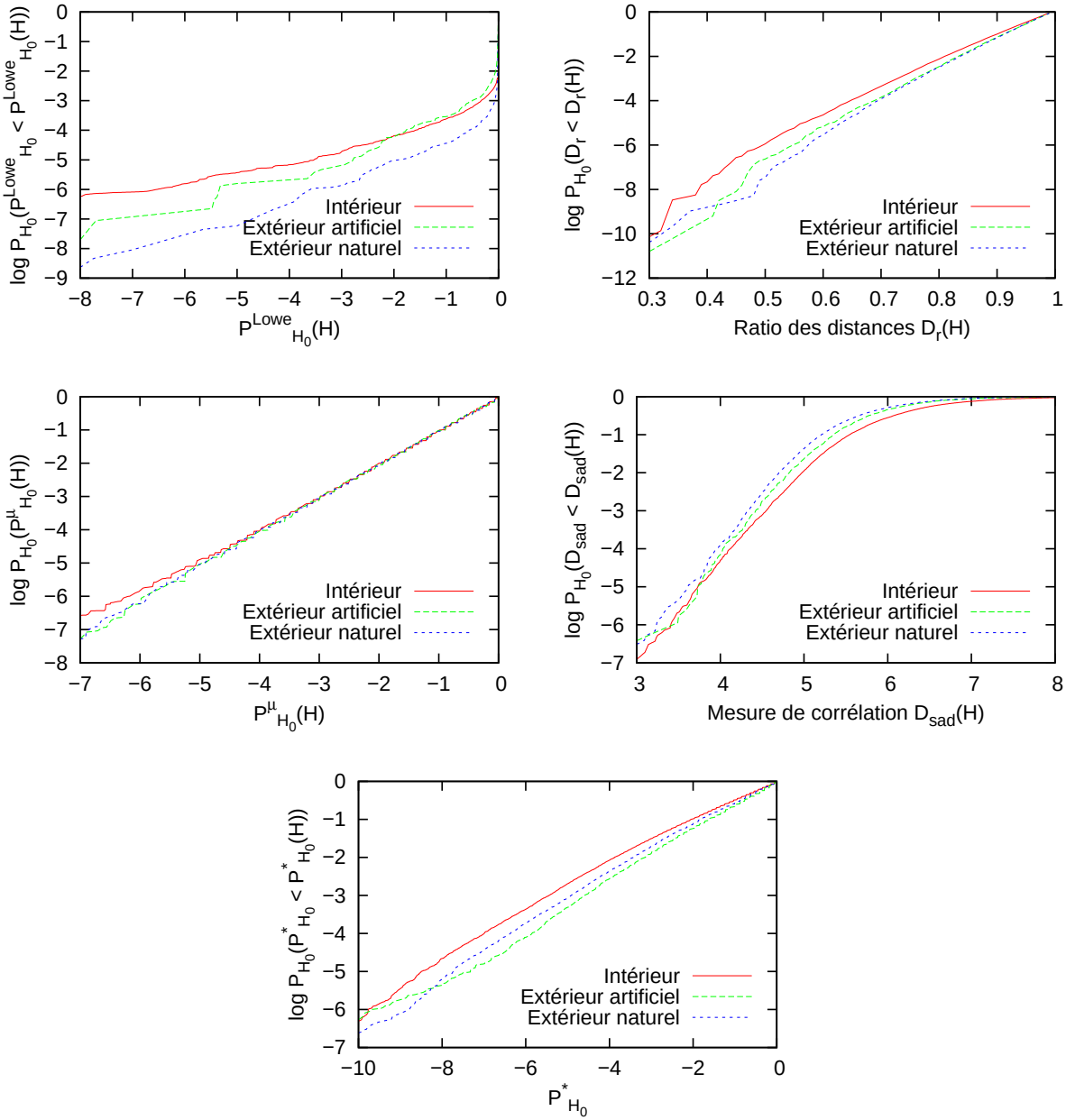


FIG. 3.7 – De gauche à droite et de haut en bas : distributions *a contrario* des variables $P^{Lowe}_{H_0}$, D_r, $P^{\mu}_{H_0}$, D_{sad} et $P^{*}_{H_0}$ pour chacun des trois ensembles d'apprentissage testés.

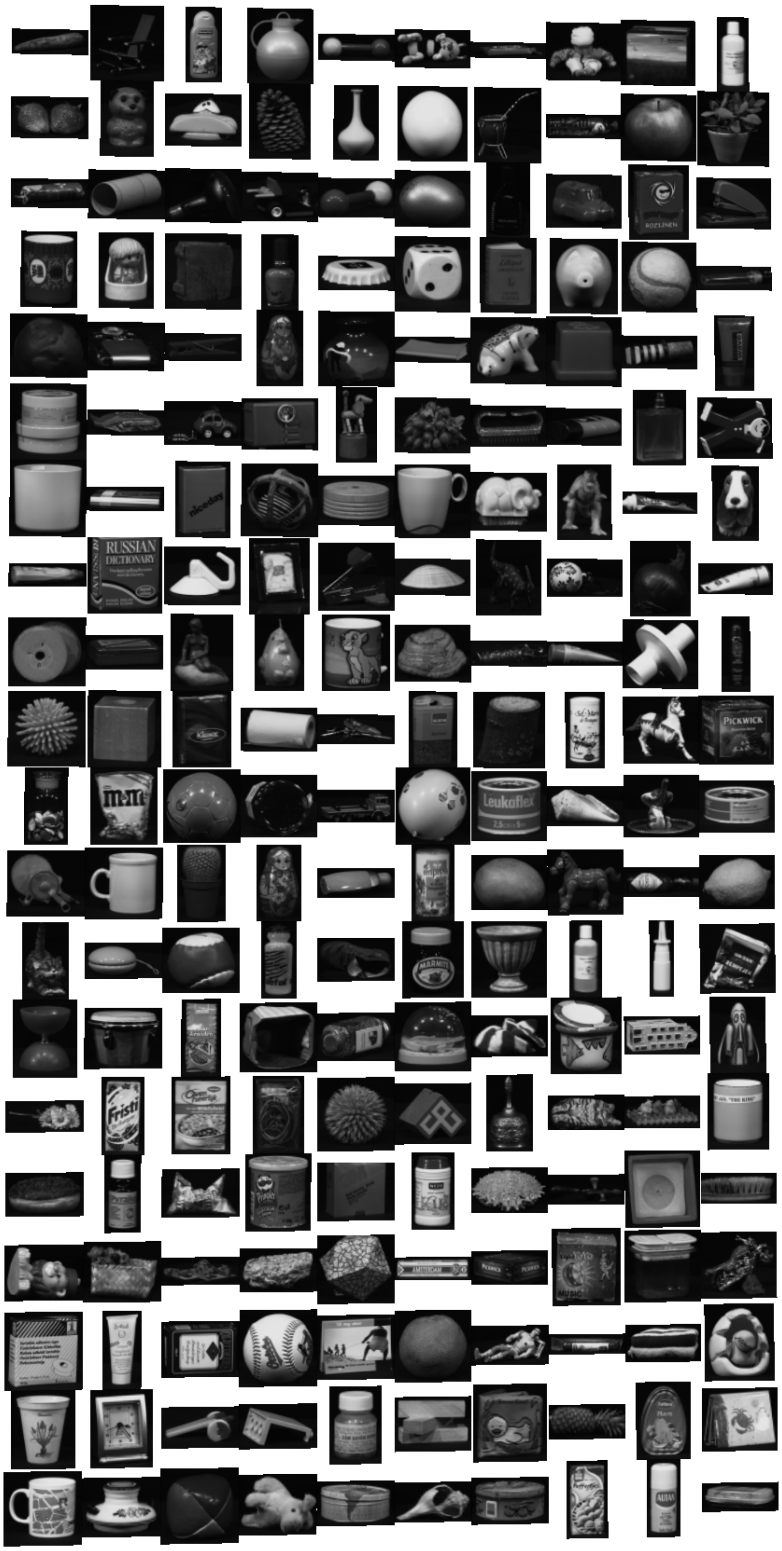


FIG. 3.8 – Vue de face des 200 objets extraits aléatoirement de la base ALOI [GBS05].

dans cette configuration. Un masque des objets a été découpé à la main pour permettre une incrustation sans fond. Ce masque n'est utilisé que pour la procédure d'évaluation. Un filtre gaussien est ensuite appliqué au masque, qui est utilisé comme indice de transparence alpha lors de l'insertion de l'objet. Ceci permet une intégration progressive des bords des objets et d'éviter un découpage trop brutal.

Pour chaque expérimentation, chacun des objets est incrusté dans une image de test choisie au hasard parmi les images de [TMFR03], [MFTM01], [SH05] et [FFIKP07] qui n'ont pas déjà été utilisées pour l'apprentissage. Au total, 200 images sont ainsi générées. Nous comparons systématiquement trois algorithmes : celui de [Low04] qui utilise uniquement la probabilité $P_{H_0}^{Lowe}$ et ne conserve que les associations dont le ratio des distances est supérieur à 0.8, le notre sans la variable D_{sad}, et le notre en utilisant toutes les variables discriminantes. Ceci nous permet d'isoler les gains provenant de la mesure de corrélation de ceux issus de l'analyse des mises en correspondance de points SIFT.

Les résultats sont analysés à l'aide de diagrammes montrant le taux d'objets correctement détectés parmi les 200 images en fonction du nombre moyen de fausses alarmes par image. Comme nous maîtrisons la génération des images de test, il est aisé de calculer ces deux grandeurs.

Dans la configuration standard, la figure 3.10 montre que le nombre de détections est significativement augmenté par rapport à [Low04] en acceptant toutes les mises en correspondance et en évaluant leur force à l'aide des variables $P_{H_0}^{Lowe}$ et $P_{H_0}^{\mu}$. Les résultats sont encore nettement améliorés si on utilise le terme de corrélation D_{sad}. Le comportement des algorithmes est ensuite comparé en faisant varier divers paramètres. La figure 3.11 étudie l'influence de la taille des objets incrustés dans l'image. La mesure de corrélation est encore plus utile lorsque les objets sont petits (50 pixels de large) et donc que peu de points SIFT peuvent être correctement associés. La figure 3.12 montre que les trois algorithmes sont relativement robustes vis-à-vis d'un bruit uniforme, et les gains apportés par notre approche restent constants. La figure 3.13 étudie l'influence du degré d'occultation des objets. La mesure D_{sad} y est naturellement sensible, et s'avère donc moins intéressante sous sa forme simple retenue en section 3.3.4 lorsque plus d'un quart de l'objet est occulté. Le même comportement se constate sur la figure 3.14 qui étudie des changements sévères d'illumination. Pour les cas extrêmes, l'apparence globale de l'objet est trop modifiée pour que la mesure D_{sad} reste utile. Cependant, dans tous les cas, les résultats de détection restent significativement améliorés par rapport à l'approche originelle de [Low04].

3.7 Discussion

À travers une application de détection d'objets, nous avons montré qu'il est également pertinent d'apprendre des distributions *a contrario* de haut niveau directement à partir d'images naturelles. L'apprentissage permet de combiner facilement des mesures hétérogènes et de prendre

FIG. 3.9 – Exemples d'incrustation artificielle d'un objet de la base sur une image naturelle. À gauche : photo de l'objet dans la base. Au milieu : objet incrusté dans une image de test. À droite : zoom sur l'incrustation de l'objet. Le changement de fond et les transformations appliquées aux objets rendent la détection relativement difficile, surtout en images d'intérieur.

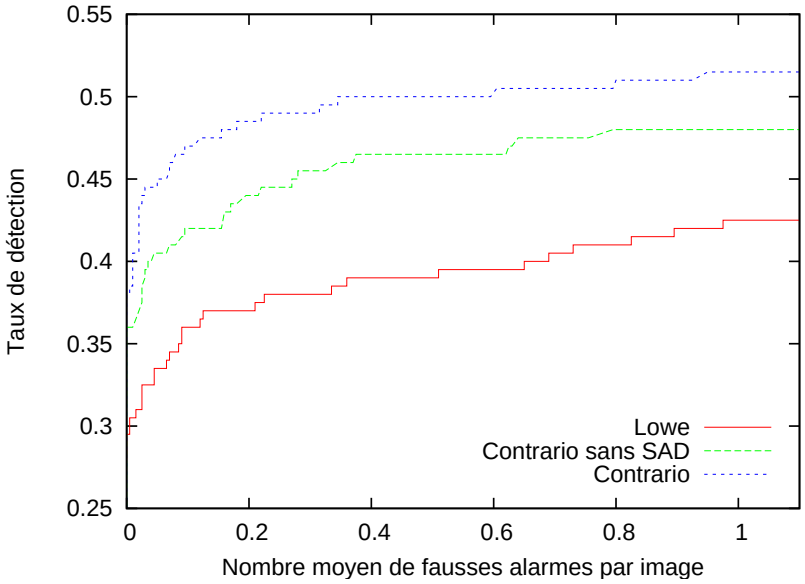


FIG. 3.10 – Taux de détection en fonction du nombre moyen de fausses alarmes par image pour des objets incrustés selon la procédure de base : dimension maximale de 100 pixels, bruit uniforme de 2%, changement de point de vue de 25 degrés, rotation dans le plan de 15 degrés, et pas d'occultation.

en compte des phénomènes difficiles à mesurer de façon analytique, comme les dépendances entre caractéristiques.

Les distributions sont relativement simples à apprendre. Seuls des exemples d'images ne contenant pas d'objets à détecter sont nécessaires, et les variables à apprendre sont choisies de façon à ce qu'un maximum de phénomènes soient déjà pris en compte. Par exemple, la variable $P_{H_0}^{Lowe}$ s'adapte déjà au nombre de points SIFT présents dans la zone de l'hypothèse ou au nombre de points SIFT du modèle, et la variable $P_{H_0}^{\mu}$ s'adapte au nombre de mises en correspondance compatibles. Les distributions de ces variables sont donc plus génériques et plus simples à apprendre que la distribution d'un vecteur brut directement constitué des variables μ, N_v, N_z, etc. Au final, une dizaine d'images d'apprentissage suffisent et le choix de ces images est peu sensible.

Nous avons également proposé un schéma pour la détection d'objets à partir de caractéristiques locales où les mises en correspondance ne sont pas filtrées précocement. Ceci permet de détecter plus d'objets quand l'apparence des objets est sensiblement altérée, et donc que les similarités entre descripteurs sont relativement faibles. Dans le cas d'un critère basé sur le ratio des distances avec le second plus proche voisin, cette approche permet également de gérer des bases plus grandes, où il est probable qu'un autre objet ait un point SIFT assez similaire. Enfin, en conservant toutes les mises en correspondance, des hypothèses sont générées même pour des objets n'ayant que quelques associations compatibles faibles. Ces hypothèses peuvent ensuite être renforcées via des mesures de similarité globales. Nous avons montré qu'un simple calcul de différences pixel à pixel permet déjà d'améliorer sensiblement les taux de détection.

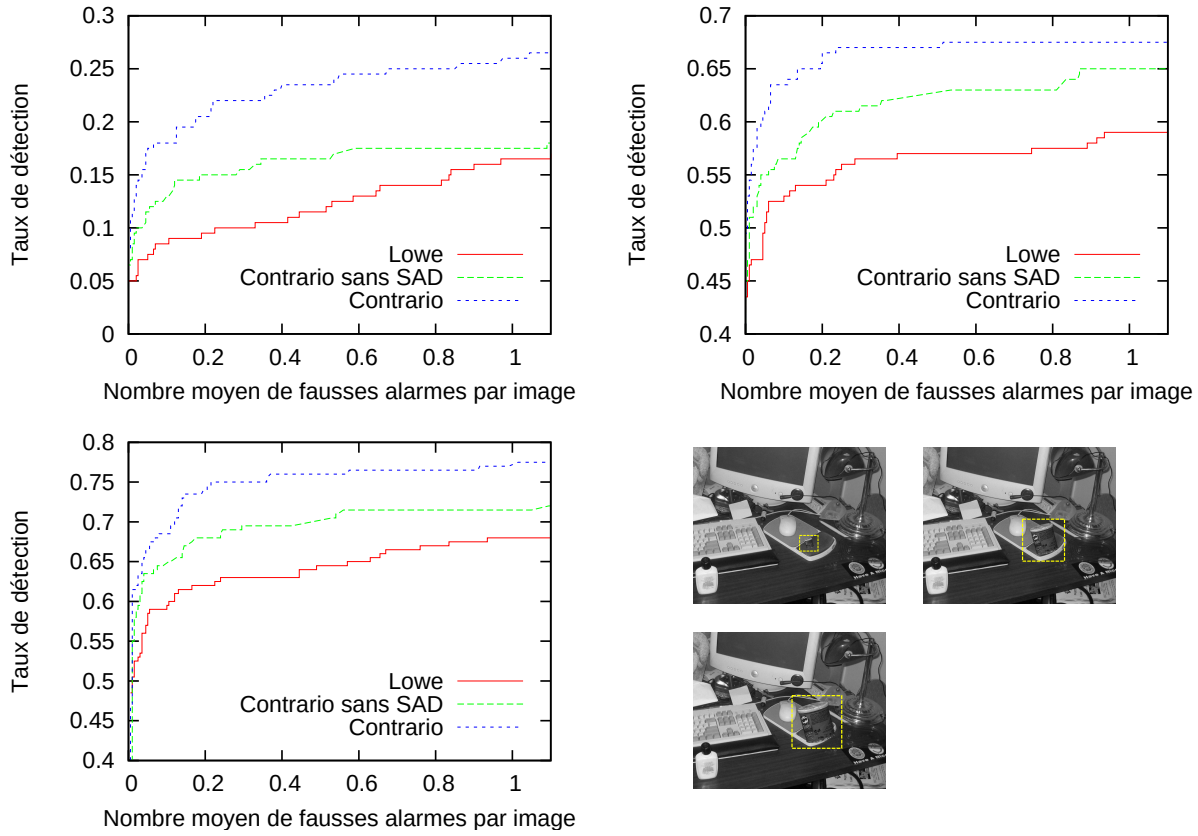


FIG. 3.11 – De gauche à droite et de bas en haut : taux de détection pour des objets incrustés avec une dimension maximale de 50, 150 et 200 pixels.

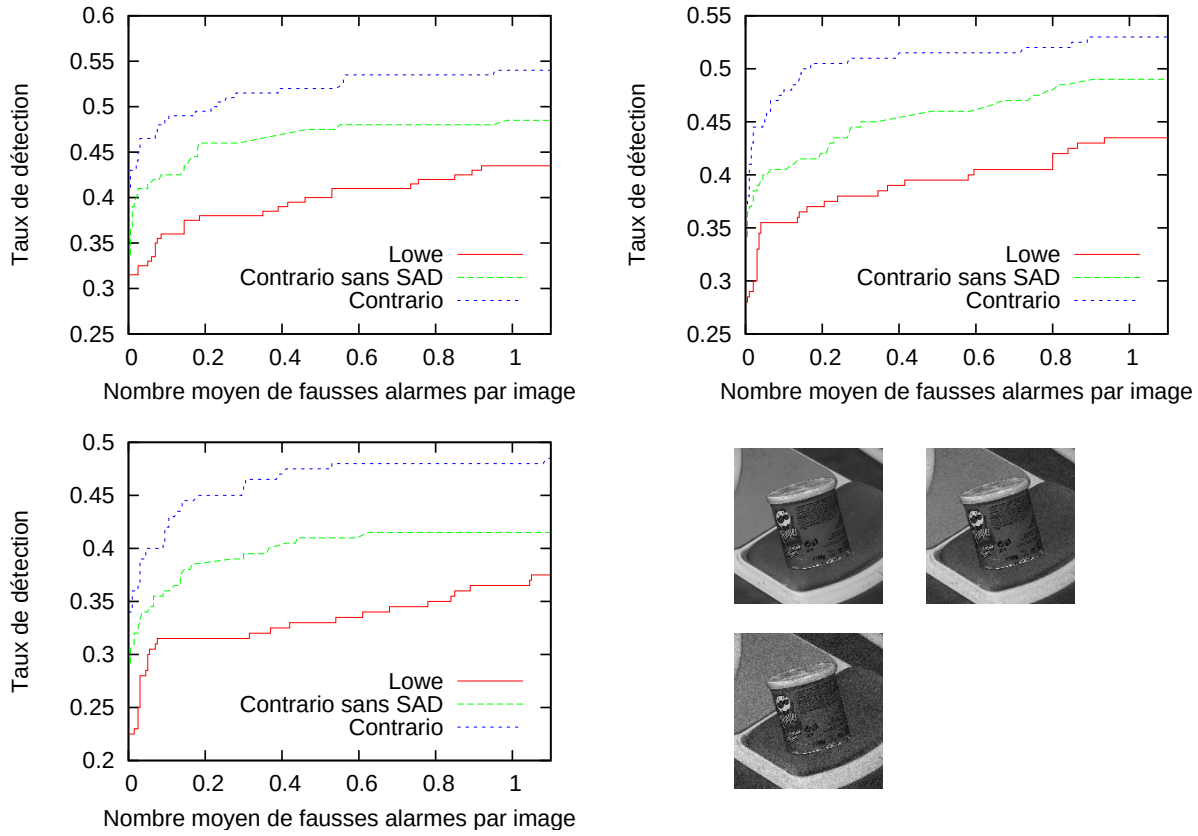


FIG. 3.12 – De gauche à droite et de bas en haut : taux de détection pour des objets incrustés avec un bruit additif de 0, 5, et 10 pour cent.

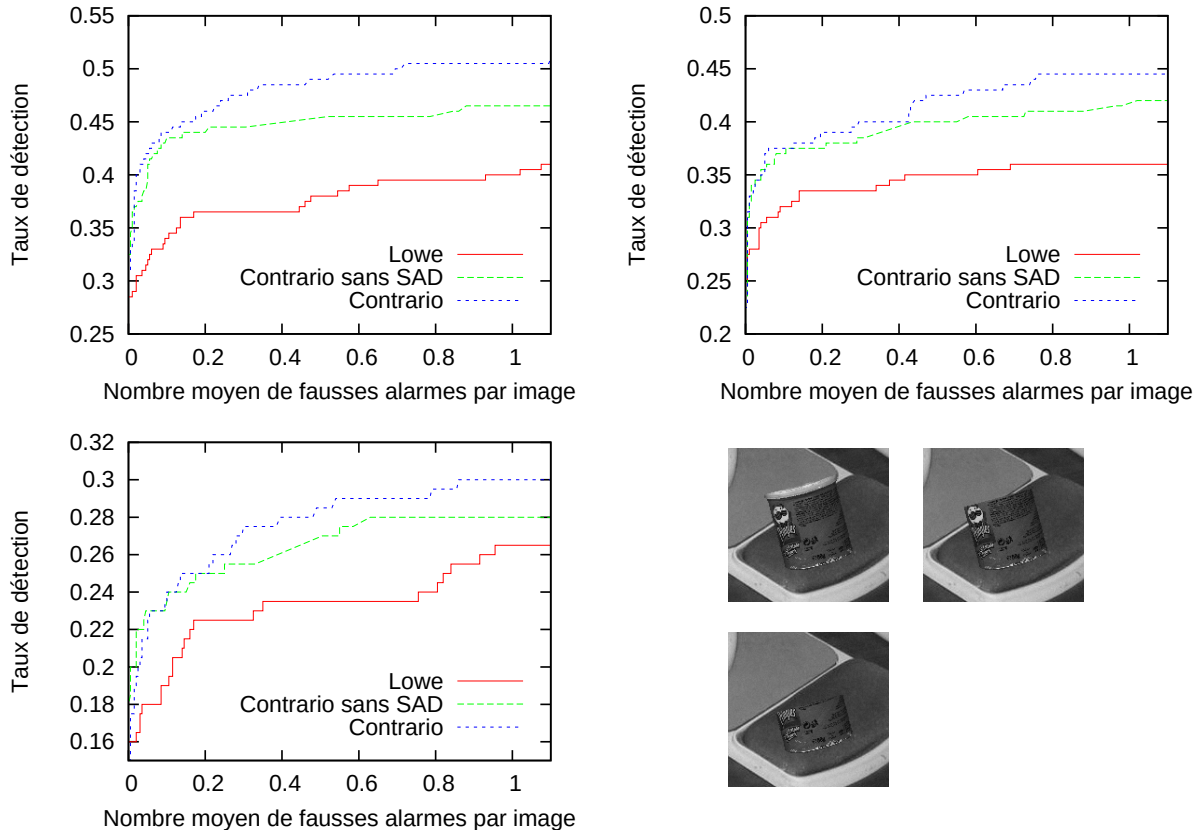


FIG. 3.13 – De gauche à droite et de bas en haut : taux de détection pour des objets incrustés avec un degré d'occultation de 10, 25 et 50 pour cent.

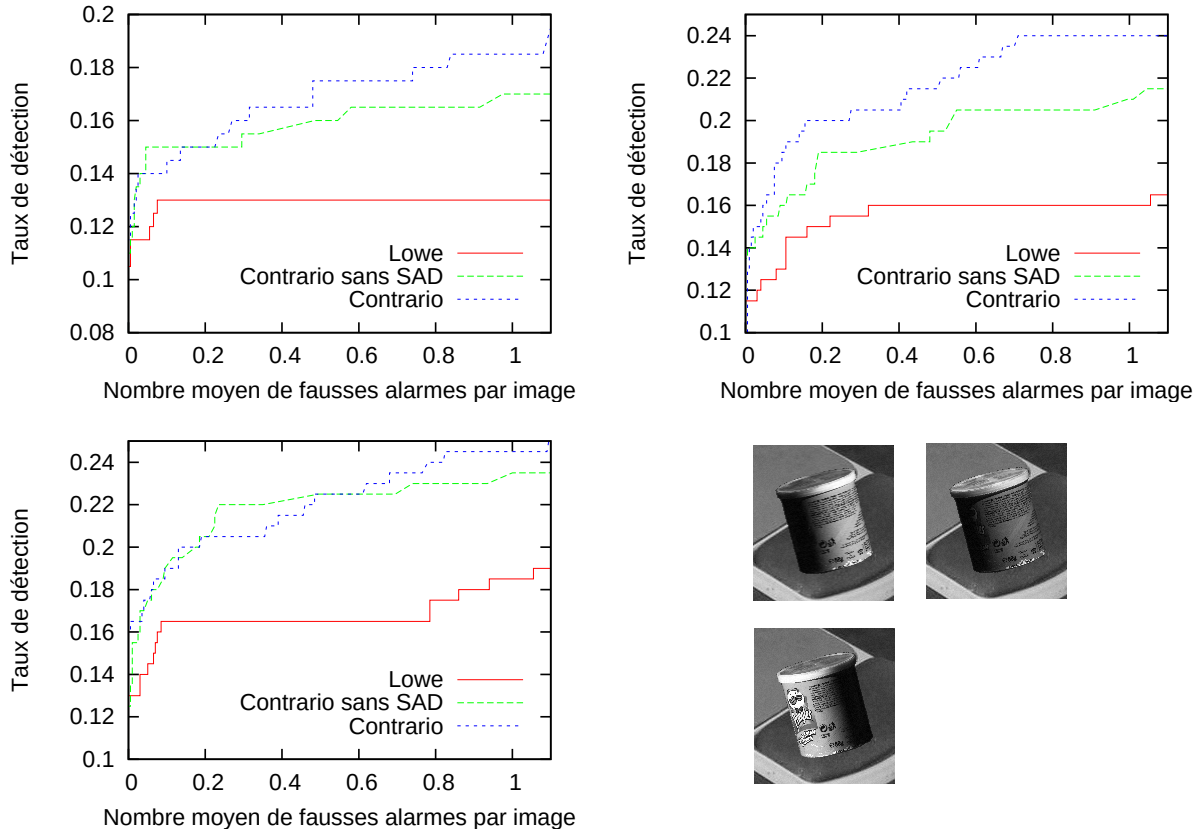


FIG. 3.14 – Taux de détection pour des objets incrustés avec trois changements d'éclairages différents. En haut à gauche : objets éclairés uniquement par la droite. En haut à droite : objets faiblement éclairés de face et par la droite. En bas à gauche : objets éclairés par la gauche.

Enfin, la méthode de décision proposée est incrémentale : la confiance dans une hypothèse peut être calculée à tout moment à partir des points SIFT déjà mis en correspondance. Les nouvelles associations ne pourront que venir renforcer la significativité de l'hypothèse. De même, le terme global de similarité D_{sad} ne peut que renforcer la significativité de l'hypothèse. S'il n'est pas calculé mais que l'hypothèse a déjà une *PFA* suffisamment faible, la décision de détection peut être prise. Cette propriété ouvre la voie à une implantation "anytime" de l'algorithme, où les décisions sont prises le plut tôt possible, sans attendre l'achèvement complet de l'analyse de l'image. C'est l'objet du chapitre suivant.

Chapitre 4

Algorithme “anytime” pour la détection d’objets *a contrario*

4.1 Introduction

L’approche *a contrario* utilise les variables discriminantes uniquement de façon positive, dans le sens où chaque variable peut éventuellement contribuer à augmenter la confiance dans la présence d’un objet, mais en aucun cas à la diminuer. Cette propriété est très intéressante, car elle permet de prendre des décisions à partir d’un sous-ensemble de variables discriminantes. En effet, considérons un ensemble $\mathbb{X}_n = \{X_1, \dots X_n\}$ de n variables discriminantes. On note $\mathbb{X} \leq \mathbb{X}(w)$ l’évènement $\{X_1 \leq X_1(w), X_2 \leq X_2(w), \dots, X_n \leq X_n(w)\}$ pour une observation w. La probabilité de fausse alarme associée est alors :

$$\mathit{PFA}_n(w) = P_{H_0}(\mathbb{X}_n \leq \mathbb{X}_n(w))$$

Supposons maintenant que seul un sous-ensemble $\mathbb{X}_k \subset \mathbb{X}_n$ de k variables soit disponible. On note $\mathbb{X}_k^C = \mathbb{X}_n \backslash \mathbb{X}_k$ l’ensemble des variables de $\mathbb{X}_n$ qui ne sont alors pas prises en compte. Puisque pour deux évènements A et B, on a toujours $P(A) \geq P(A, B)$, on peut montrer que la probabilité de fausse alarme calculée à partir de $\mathbb{X}_k$ est nécessairement supérieure ou égale à PFA_n :

$$\begin{aligned} \mathit{PFA}_k(w) &= P_{H_0}(\mathbb{X}_k \leq \mathbb{X}_k(w)) \\ &\geq P_{H_0}(\mathbb{X}_k \leq \mathbb{X}_k(w), \mathbb{X}_k^C \leq \mathbb{X}_k^C(w)) \\ &\geq \mathit{PFA}_n(w) \end{aligned}$$

Une conséquence directe de ce résultat est que si une *PFA* n’utilisant qu’un sous-ensemble de variables discriminantes passe sous le seuil de décision ε, alors la *PFA* utilisant toutes les variables discriminantes franchira également le seuil. Il est donc possible de produire une détection à partir du moment où un sous-ensemble de variables discriminantes est suffisamment significatif.

Comme le souligne [DeC02], la capacité à prendre des décisions à partir d’informations partielles ouvre la voie à des algorithmes de vision capables de fournir des résultats progressive-

ment au cours de leur exécution. Ces algorithmes peuvent alors être interrompus à tout moment pour satisfaire des contraintes de temps réel ou limité. Cette catégorie d’algorithmes, appelés “anytime” en intelligence artificielle, s’oppose à l’approche classique où l’on doit nécessairement attendre l’achèvement complet de tous les calculs pour obtenir les premiers résultats. Le caractère “anytime” d’un algorithme est particulièrement intéressant en vision artificielle, où la masse d’information est telle qu’il est souvent impossible de la traiter intégralement dans un temps raisonnable. Il est alors plus intéressant de rechercher à maximiser la pertinence des résultats dans le temps imparti.

Une deuxième propriété du cadre *a contrario* pousse dans le sens “anytime”. En effet, il permet de quantifier de façon homogène la pertinence de l’information portée par chacune des variables discriminantes, en mesurant à quel point les valeurs observées sont improbables sous une hypothèse de hasard. Comme l’a illustré [PRSM95], cette propriété est très intéressante pour un algorithme “anytime” car elle permet de se focaliser à tout moment sur les informations les plus significatives, et donc sur celles qui sont le plus susceptibles de mener à une détection. Le temps de calcul peut alors être réservé en priorité aux hypothèses les plus prometteuses afin de maximiser le nombre de détections dans un temps imparti.

Enfin, nous avons utilisé dans la section 3.3.3 du chapitre précédent un principe de maximalité, récurrent dans les approches *a contrario*, qui consiste à prendre des décisions à partir des évènements les plus significatifs de l’ensemble des données. Cette démarche assure également une croissance monotone de la confiance dans la présence d’un objet en fonction des nouvelles données disponibles. Ainsi, dans l’exemple de la détection d’objets du chapitre 3, le fait d’observer de nouvelles mises en correspondance compatibles avec une hypothèse peut uniquement augmenter la confiance dans la présence d’un objet. Cette propriété vient compléter l’intérêt de la prise de décision à partir d’un sous-ensemble de variables discriminantes, en autorisant également une prise de décision basée sur des variables discriminantes calculées à partir d’un sous-ensemble des données de l’image.

Ces éléments motivent une approche “anytime” basée sur le cadre *a contrario*. Nous faisons un premier pas dans cette direction dans ce chapitre où nous expérimentons une implantation “anytime” de la détection d’objets du chapitre 3. Nous proposons pour cela une architecture logicielle adaptée permettant de traiter en parallèle les étapes de bas et de haut niveau, afin de pouvoir atteindre l’étape de décision pour les objets les plus saillants bien avant l’achèvement de tous les calculs intermédiaires. Combinée avec les propriétés du cadre *a contrario* pour prioriser l’information et prendre des décisions à partir de données partielles, l’architecture proposée s’avère capable de détecter les objets d’une image de façon “anytime”.

Après un rapide tour d’horizon des approches “anytime” en vision, nous identifierons un certain nombre de propriétés architecturales utiles pour donner un caractère “anytime” à un algorithme de vision. Ces propriétés ont inspiré l’architecture que nous proposons, qui sera détaillée puis évaluée expérimentalement.

4.2 Algorithmes de vision “anytime”

Dès l’âge de 6 ans, un humain est capable de détecter plus de 10 000 catégories (bâtiment, vélo, visage, etc.) d’objets différents [Bie87], et continue à apprendre de nouvelles catégories pendant toute sa vie. Cumulé avec le nombre d’instances spécifiques de chaque catégorie qu’un humain peut reconnaître (la tour Eiffel, la Joconde, etc.), le nombre total d’objets distincts identifiables est énorme. Dans une simple image, chacun de ces objets (ou presque, selon le contexte) peut être présent sous différentes formes, à divers emplacements, ce qui rend l’information totale potentiellement analysable immense. Les ressources et le temps de traitement étant limités, il n’est pas possible, même pour le système visuel humain, d’analyser systématiquement l’ensemble de son champ visuel. Heureusement, cela n’est pas nécessaire, et nous avons notamment recours à des mécanismes d’attention visuelle pour se focaliser systématiquement sur les éléments les plus importants en fonction de la tâche courante et de leur saillance [TCKW$^+$95, IK01]. Ce principe nous permet de détecter en premier les objets les plus importants, puis de continuer progressivement à analyser les autres objets en fonction du temps disponible.

Appliquées à la vision artificielle, ces observations conduisent à ne pas chercher absolument à traiter toute l’information visuelle et toutes les hypothèses possibles le plus rapidement possible, mais plutôt à chercher comment maximiser le nombre et la pertinence des détections dans le temps imparti. Cette adaptation des algorithmes au temps disponible est très importante en robotique par exemple, et cela a ouvert un champ de recherche à part entière en intelligence artificielle : les algorithmes “anytime” [Zil96]. La propriété principale d’un algorithme “anytime” est que la qualité de son résultat s’améliore de façon progressive avec le temps de calcul disponible, et qu’il peut donc être interrompu à tout moment, comme l’illustre la figure 4.1. Un algorithme de détection “anytime” a plusieurs intérêts :

- il est parfaitement adapté à du temps réel ou du temps contraint, même variable (e.g. en robotique, le temps alloué aux tâches de vision peut dépendre de l’action courante du robot) ;
- même quand le temps est insuffisant pour analyser toute l’image, les structures les plus évidentes peuvent malgré tout être détectées ;
- en priorisant les calculs sur les données les plus prometteuses, il devient possible de calculer des caractéristiques complémentaires pour améliorer les taux de détection sans détériorer les performances dans les cas simples.

De nombreux algorithmes de vision présentent implicitement un caractère “anytime”. Par exemple, les approches qui optimisent itérativement une énergie améliorent en général leur résultat progressivement au cours du temps.

Mais de façon assez surprenante, relativement peu de travaux se sont intéressés explicitement à la vision “anytime” en cherchant à optimiser le profil de performance des algorithmes, c’est-à-dire à optimiser la qualité des résultats en fonction du temps écoulé. [HL97] propose un algorithme de rendu 3D qui s’améliore progressivement avec le temps. [DN95] extrait et traque

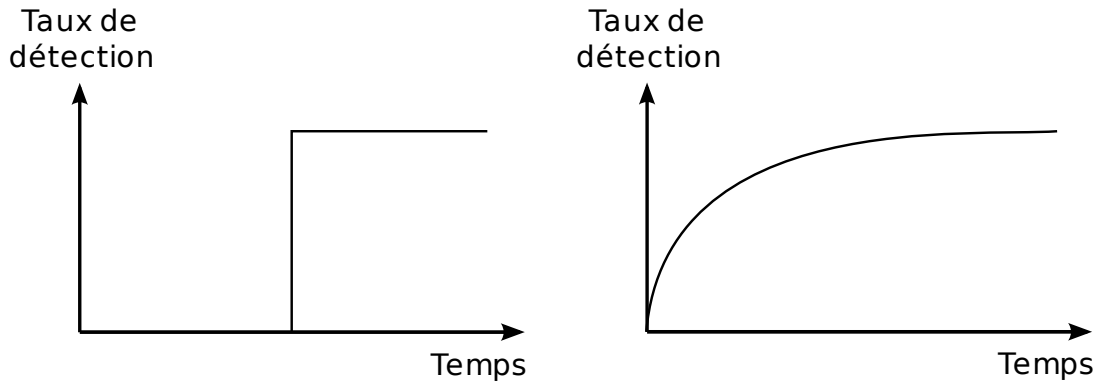


FIG. 4.1 – Le caractère “anytime” d’un algorithme se mesure par son profil de performance qui représente la qualité de la sortie en fonction du temps d’exécution. Un algorithme classique (à gauche), produit des résultats uniquement à la fin de son exécution. Appliqué à la détection d’objet, cela signifie que si l’algorithme est stoppé avant, aucune détection ne pourra être effectuée. À l’inverse, un algorithme “anytime” (à droite), produit progressivement des détections et, s’il est interrompu avant la fin de son exécution, les objets les plus évidents auront peut-être eu le temps d’être détectés.

des contours avec une précision qui augmente progressivement. [SB99, SB02] proposent un algorithme de détection d’obstacle qui affine progressivement une stratégie d’évitement optimale. [KFM06] modifie des filtres bas niveau pour leur donner un comportement “anytime”. Enfin, [BAP07] utilise des mesures de similarité dont la précision augmente progressivement pour le recalage d’images.

Mis à part ces quelques exemples, la plupart des efforts se sont concentrés sur l’optimisation globale du temps d’exécution, et le caractère “anytime” des algorithmes est alors le plus souvent accidentel. Comme nous allons le voir dans la section suivante, une perspective “anytime” motive pourtant des changements importants sur la manière de concevoir un algorithme de vision, et mérite donc d’être traitée explicitement.

4.3 Propriétés architecturales motivées par un comportement “anytime”

Nous avons vu en introduction un certain nombre de propriétés liées au traitement de l’information et à la prise de décision qui sont très utiles dans un cadre “anytime”, et pour lesquelles le cadre *a contrario* s’avère très adapté. Nous nous intéressons maintenant à des propriétés plus algorithmiques concernant les stratégies de recherche des objets.

Ces dernières sont généralement classifiées en trois catégories [HS93, BCFM00] : purement ascendantes (bottom-up), purement descendantes (top-down), et hybrides. De façon caricatu-

rale, les méthodes ascendantes partent de l'image pour progressivement en arriver à détecter des objets, alors que les méthodes descendantes partent d'une hypothèse de présence d'un objet et descendent le vérifier par des mesures sur l'image. Deux exemples très représentatifs sont la détection d'objets de [Low04] et celle de [VJ02]. Alors que [Low04] extrait des caractéristiques locales dans l'image pour ensuite les regrouper et formuler des hypothèses sur la présence d'un objet, [VJ02] part d'un modèle d'objet et teste toutes les positions dans l'image pour vérifier si l'objet est présent. Un parallèle peut être établi avec les méthodes de classification. Les méthodes discriminantes partent de caractéristiques mesurées directement sur les données pour en déduire directement le modèle correspondant. Elles sont donc ascendantes. À l'inverse, les méthodes génératives partent d'un modèle, et calculent la probabilité que les données aient été générées par le modèle testé. Comme nous allons le voir, la combinaison d'influences ascendantes/descendantes est très utile dans une perspective “anytime”.

Influences ascendantes. Pour obtenir un algorithme capable de privilégier les zones les plus importantes et les plus prometteuses de l'image, il est nécessaire de prendre en compte les données pour diriger la recherche, sous peine d'obtenir un temps de traitement constant. Par exemple, l'approche à base de fenêtres glissantes de [VJ02] évalue de façon descendante si un objet est présent dans chacune des positions possibles. Le temps d'exécution d'un tel algorithme augmente donc rapidement avec la taille de l'image, surtout si les objets sont recherchés sur différentes échelles et transformations. Il est impossible de détecter les objets les plus évidents en premier, puisque l'image est parcourue dans un ordre prédéfini. Il faut également tester chaque objet séparément, et la complexité augmente rapidement avec le nombre d'objets à détecter, même si des techniques de partage permettent de limiter l'augmentation du nombre de caractéristiques à calculer [TMF04]. Pour le moment, les seuls algorithmes temps réel sont limités à un seul type d'objet et nécessitent des optimisations avancées [LBHZ08, ZZS07, WDSS08]. Pour passer à l'échelle et pour obtenir un comportement “anytime”, la recherche doit donc être dirigée, au moins partiellement, par des indices fournis directement par l'image.

Parallélisme hiérarchique. Les influences ascendantes sont donc indispensables pour permettre un comportement “anytime”. Elles mènent souvent à des algorithmes hiérarchiques, où chaque étape traite des données de plus en plus abstraites avant de les transmettre à l'étape de niveau supérieur. Là aussi, la détection d'objets de [Low04] est un bon exemple. La première étape, de plus bas niveau, extrait les points SIFT de l'image. La seconde étape, juste au-dessus, les met en correspondance avec les points SIFT de la base d'objets. La troisième étape, encore au-dessus, regroupe les mises en correspondance cohérentes. Enfin, la dernière étape, tout en haut, prend les décisions. Pour qu'un tel algorithme puisse avoir un comportement “anytime” et traiter plus vite les hypothèses les plus prometteuses, il est essentiel d'introduire du parallélisme entre les différents niveaux de traitements. En effet, si chaque étape doit attendre l'achèvement complet de l'étape précédente pour commencer à travailler, aucune détection ne pourra être effectuée avant que les trois premières étapes aient traité toute l'image. Un tel parallélisme hiérarchique est présent dans les systèmes de vision biologiques, notamment pour que les étapes de plus haut niveau puissent influencer les étapes de plus bas niveau le plus tôt possible [Bar03, DRMFT04, LMRL98, TCKW$^+$95].

Influences descendantes. Le rôle des influences descendantes et ascendantes a été beaucoup discuté dans la littérature. Si les premiers modèles computationnels de vision étaient purement ascendants [Mar82], il est aujourd’hui clair que les tâches de bas niveau comme l’extraction de contours ou la segmentation ne peuvent pas être résolues de façon complète sans l’aide d’influences de plus haut niveau, ce qui pousse de plus en plus de travaux à combiner par exemple détection d’objets et segmentation en régions homogènes de façon intime [ZTY07, LLS08, ZD05, LW06, BSU04, Ull07, TCYZ05]. Les influences descendantes permettent également de prendre en compte le contexte de l’image, par exemple pour établir un *a priori* sur les positions des objets [Tor03, MTEF05] ou pour détecter les objets utiles à une tâche précise [NI02]. Enfin, de nombreuses approches utilisent des influences descendantes pour, à partir des hypothèses courantes, formuler des prédictions sur les emplacements des futures caractéristiques locales pertinentes [FTVG05, KMY06, MMP04, TL06]. Dans une perspective “anytime”, nous verrons que ce type de prédictions peut également servir à prioriser les caractéristiques locales susceptibles d’être pertinentes et donc à les faire remonter plus vite.

4.4 Choix d’une architecture adaptée

Pour autoriser un parallélisme hiérarchique, des influences ascendantes et descendantes, et une focalisation sur les données les plus prometteuses, une architecture logicielle adaptée est nécessaire. En effet, les approches classiques, généralement purement séquentielles, ne sont pas assez souples pour intégrer ce type de comportements.

Quoique non explicitement dans un but “anytime”, certains systèmes d’interprétation d’images des années 1980-1990 apportaient déjà des réponses à cette problématique. Par exemple, le système VISION [HR78] devenu par la suite le Schema System [DCB+89], permettait de combiner des influences ascendantes et descendantes en utilisant des modules experts distribués, chacun étant dédié à un élément particulier à détecter. Les systèmes SIGMA [MH85] et MESSIE II [San95] reposaient quant à eux sur un tableau noir central pour activer différents modules et leur permettre d’échanger des résultats et des prédictions. On peut également noter le système MAVI [BD94], basé sur une hiérarchie d’agents dédiés à des tâches spécifiques, chaque agent parent pouvant contrôler et faire des requêtes à ses agents fils. On se référera à [CL97] ou [Duc01] pour une étude bibliographique plus complète et détaillée. Ces systèmes étaient très ambitieux et cherchaient à intégrer et faire interagir un grand nombre de modules de vision pour analyser une scène, avec en contrepartie des procédures de communication, de contrôle et de planification de tâches très complexes.

Notre objectif est moins ambitieux que celui de ces systèmes à vocation généraliste. Nous proposons donc une architecture qui reprend certaines idées de ces travaux, mais de façon beaucoup plus simple. En effet, de façon assez similaire au système MAVI, nous proposons une architecture à base d’agents [FG97, Nwa96, WJ95]. Cette approche est en effet bien adaptée à des situations où différentes tâches doivent évoluer en parallèle, et a fait l’objet de divers travaux en vision [LT99, RBG+04, ZPG+04]. Contrairement à MAVI, nous proposons en revanche des agents totalement autonomes, ce qui permet d’assurer une meilleure modularité et de

s'affranchir de procédures de contrôle complexes. Chaque agent correspond alors à une tâche particulière, et s'exécute de façon autonome dans son propre fil d'exécution, tant qu'il a des données à traiter.

Les agents sont connectés entre eux de façon hiérarchique, et communiquent par messages. Les messages reçus par un agent sont stockés dans une mémoire tampon de façon asynchrone. Un agent envoyant un message n'a donc pas à attendre que les agents destinataires le traitent pour continuer à travailler. Ce mode de communication permet d'obtenir un système très souple assurant une indépendance forte entre les agents, qui n'ont pas besoin de savoir à quels agents ils sont connectés ni de qui ils reçoivent les messages. Chaque agent se contente de traiter les messages qu'il a reçus, et envoie des messages de résultat à ses agents parents, de façon ascendante, ou bien forge des messages contenant des prédictions à destination de ses agents fils, de façon descendante. Ce mécanisme à base de messages permet également d'intégrer facilement les notions de priorisation et de focalisation sur les données les plus significatives. Nous associons pour cela une priorité à chaque message. Les agents traitent alors les messages en attente par ordre de priorité. Dans notre cas, la priorité associée à chaque message sera déduite de la significativité *a contrario* des données qu'il transporte.

De plus, une mémoire centrale, rappelant les systèmes à base de tableau noir, permet aux agents de partager des données. Pour notre application, il s'agit des hypothèses courantes de présence d'objet. Chaque agent peut donc mettre à jour ou accéder aux données d'une hypothèse. En cas de mise à jour, il peut envoyer un message à ses agents parents pour signaler qu'un élément d'une hypothèse a été ajouté ou modifié. Ce mode de fonctionnement permet de ne faire circuler que des messages relativement légers entre les agents, et autorise plusieurs agents à travailler en parallèle sur différents aspects d'une même hypothèse.

Le modèle d'agent ainsi proposé est illustré dans la figure 4.2.

Chaque agent disposant de son propre fil d'exécution parallèle, l'architecture logicielle proposée peut donc tirer parti d'architectures matérielles multiprocesseurs. Pour autoriser une gestion plus souple du nombre de processus parallèles, nous ajoutons la possibilité de multiplier le nombre d'agents dédiés à une tâche en permettant de restreindre chaque agent à une zone limitée de l'image. Ce parallélisme spatial et l'adéquation avec une architecture matérielle à plusieurs processeurs seront discutés dans les sections 4.7 et 4.8.

4.5 Application à la détection d'objets

Les principes de la section précédente se concrétisent de façon naturelle pour l'algorithme de détection d'objets du chapitre 3. Les cinq types de traitements nécessaires dans l'algorithme de détection conduisent à introduire cinq types d'agents :

- *SiftExtractor* : ces agents extraient les points SIFT de l'image.
- *SiftMatcher* : ces agents mettent en relation les points SIFT extraits avec les points de la base.

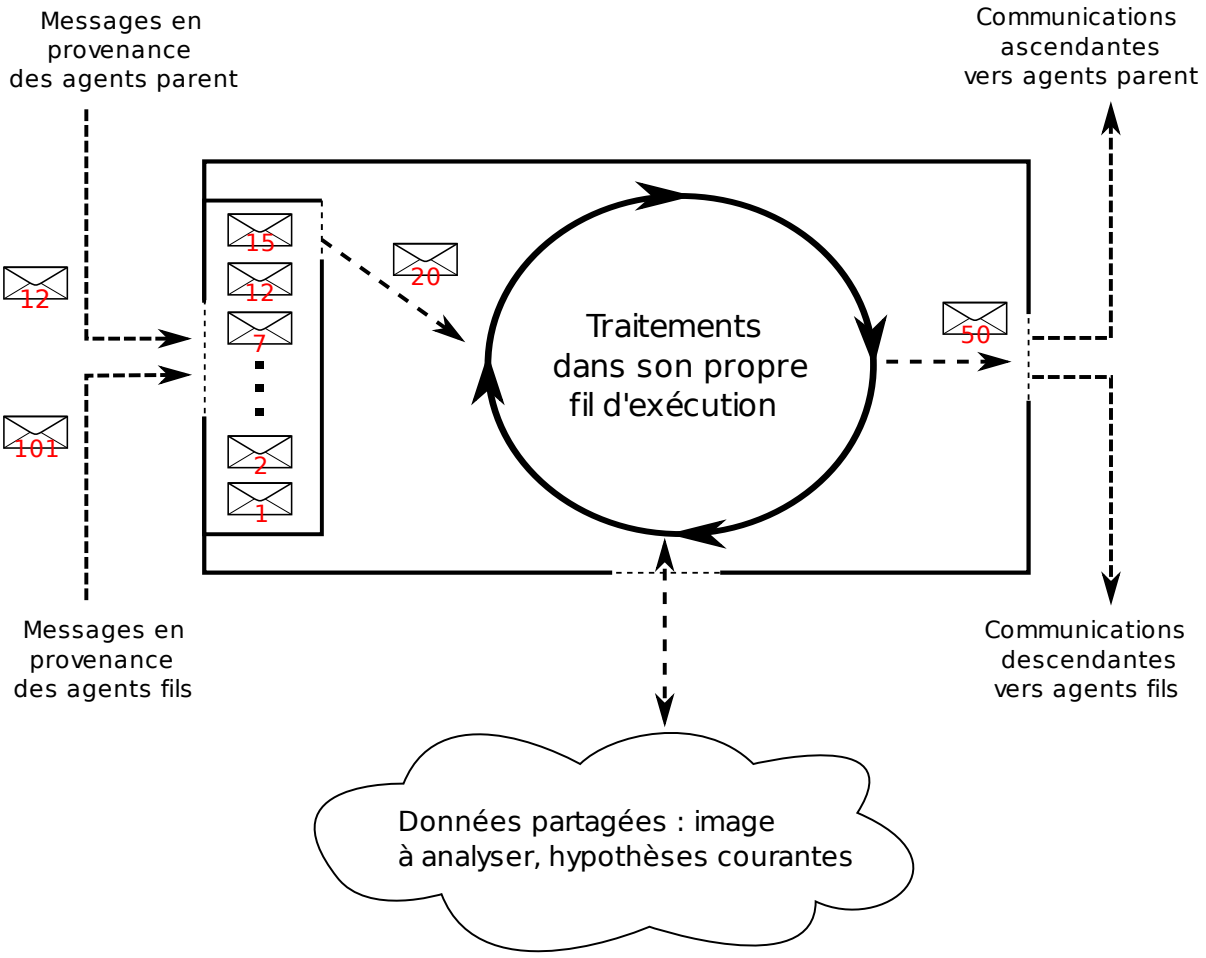


FIG. 4.2 – Vue structurelle d’un agent. Les messages en provenance des agents parents et des agents fils sont stockés par ordre de priorité dans une mémoire tampon asynchrone. Tant qu’il y a des messages, l’agent les traite dans son propre fil d’exécution. Il envoie alors les résultats sous forme de messages à ses parents ou à ses fils. En plus de sa mémoire propre, chaque agent a accès à des informations globales, telles que l’image en cours d’analyse ou les hypothèses courantes.

– *SiftClusterer* : cet agent regroupe les mises en correspondance compatibles pour formuler des hypothèses de pose.
– *SadComputer* : étant donné une hypothèse de pose, cet agent calcule D_{sad}, la somme des différences absolues entre la zone de l'image candidate et le modèle concernés (voir section 3.3.4).
– *Main* : cet agent regroupe les informations disponibles sur les hypothèses pour prendre une décision.

L'organisation des tâches au sein de l'algorithme de détection se traduit naturellement par une organisation hiérarchique des agents, représentée sur la figure 4.3.

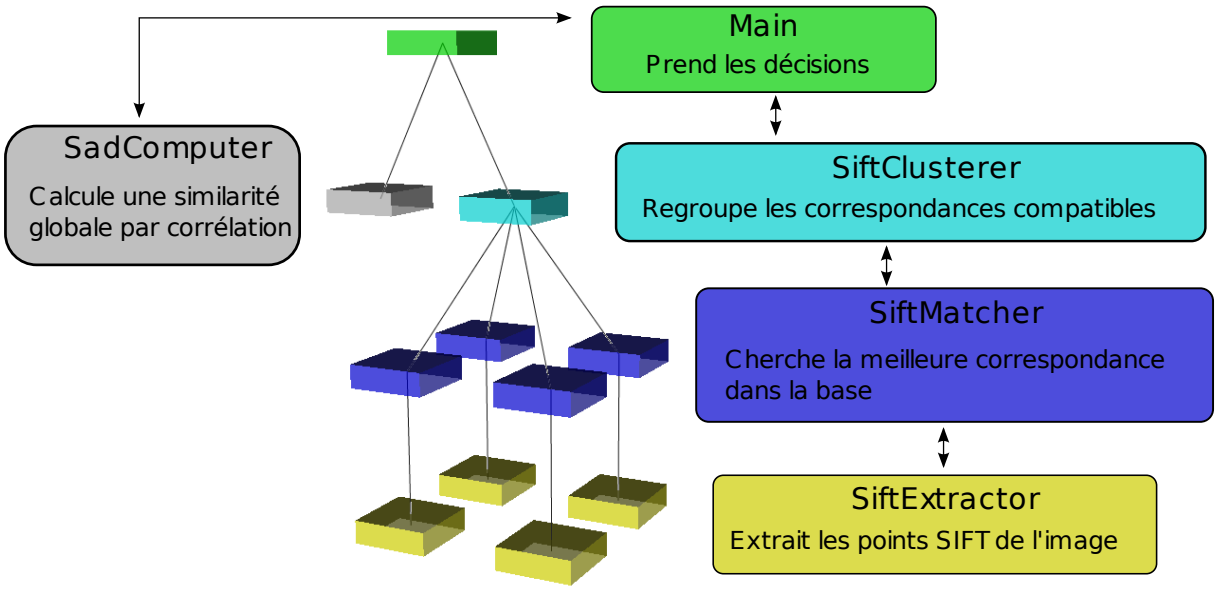


FIG. 4.3 – Architecture hiérarchique à base d'agents pour la détection d'objets. Les agents ont tous la même structure, seules les tâches effectuées changent. Chaque couleur correspond à un type d'agent. Ils s'exécutent tous indépendamment en parallèle, et communiquent par des messages asynchrones bidirectionnels. Les agents peuvent être restreints à une zone de l'image, ce qui permet de paralléliser les traitements spatialement. Ici, les agents *SiftExtractor* et *SiftMatcher* travaillent chacun sur un quart de l'image. L'architecture proposée peut ainsi mieux tirer parti d'une machine ayant plusieurs processeurs.

4.5.1 Déroulement de la détection sur une image

La hiérarchie d'agents est initialisée par les agents *SiftExtractor* qui commencent par extraire les points SIFT de l'image. Pour chaque point extrait, un message est généré et envoyé aux agents parents, les *SiftMatcher*. Ces derniers recherchent pour chaque point le plus proche dans la base, et envoient un message pour chaque mise en correspondance à l'agent parent, le *SiftClusterer*, qui regroupe chaque nouvelle association avec les précédentes compatibles. Chaque fois qu'une mise en correspondance vient s'ajouter à une hypothèse, le *SiftClusterer*

envoie un message à l’agent *Main* contenant l’hypothèse mise à jour. Ce dernier évalue alors si la *PFA* de l’hypothèse est suffisamment faible pour produire une détection. Si ce n’est pas le cas, il envoie un message contenant l’hypothèse à l’agent *SadComputer*. Cet agent calcule la mesure de similarité globale D_{sad} pour chaque hypothèse reçue, et renvoie l’hypothèse mise à jour à l’agent *Main*, qui reconsidère alors l’hypothèse. L’agent *Main* peut également formuler des prédictions sur les emplacements des futurs points SIFT, qu’il envoie sous forme de messages descendants. Ces prédictions seront traitées par les agents *SiftExtractor* qui recherchent autour de la zone prédite si un point SIFT est similaire à la prédiction. Si un point est trouvé, un message avec une grande priorité est envoyé aux agents *SiftMatcher*, permettant alors une mise en correspondance prioritaire du point SIFT.

4.5.2 Priorité associée aux messages

Pour que les agents se focalisent en permanence sur les hypothèses les plus prometteuses, les messages sont traités par ordre de priorité. Chaque fois que c’est possible, la priorité d’un message est déterminée de façon inversement proportionnelle à la probabilité de fausse alarme associée à la donnée qu’il transporte. Par exemple, pour les messages transportant une hypothèse $\mathcal{H}$ la priorité sera donnée par $-\log \mathit{PFA}(\mathcal{H})$ (voir section 3.3.5). Ainsi, les données les plus significatives seront systématiquement traitées en premier à tous les niveaux.

4.6 Messages et traitements effectués par chaque agent

Nous présentons maintenant les traitements effectués par chaque agent de façon plus détaillée.

4.6.1 Les agents SiftExtractor

Dans un premier temps, ils extraient les points SIFT de l’image, qu’ils transmettent de façon ascendante aux *SiftMatcher*. En pratique, pour garantir que les points SIFT extraits correspondent à ceux de [Low04], nous utilisons le binaire fourni par l’auteur. Ces agents se contentent donc ici de forger des messages contenant chacun des points SIFT extraits par le programme externe.

La priorité associée à chaque message ne peut être déterminée à ce niveau que grâce à un *a priori* sur les zones où les objets peuvent être présents. Dans le cas général, cette priorité est fixée de façon aléatoire, pour ne pas dépendre de l’ordre de parcours de l’image, et entre -1 et 0, pour avoir une priorité inférieure aux messages de prédictions que nous détaillerons plus loin. Toutefois, pour des images en extérieur, il peut être intéressant d’utiliser des mesures de saillance bas niveau, comme les cartes de saillance de [IKN+98] qui mesurent le contraste entre une zone de l’image et son voisinage. Plus le contraste est élevé, plus la région a une valeur

de saillance élevée. Ce modèle est particulièrement pertinent pour des objets isolés sur un fond relativement homogène, comme le montre la figure 4.4. Nous expérimenterons l'utilisation de cette mesure de saillance sur un jeu de test spécifique composé d'images naturelles en extérieur dans la section 4.9.

Les agents *SiftExtractor* reçoivent également des requêtes descendantes initiées par l'agent *Main*. Il s'agit de prédictions sur l'emplacement de certains points SIFT. Chaque message de prédiction contient un point SIFT de la base et une transformation de similarité (orientation, échelle, position) supposée. L'agent *SiftExtractor* concerné recherche alors parmi les points SIFT de la zone quels sont ceux qui sont compatibles avec la transformation présumée. Chaque point compatible est ensuite envoyé aux agents *SiftMatcher* via un message ayant comme priorité la distance du χ^2 entre le point de l'image et le point du modèle. Cette distance étant nécessairement positive, ces messages seront toujours plus prioritaires que ceux dont la priorité a été fixée aléatoirement entre -1 et 0.

4.6.2 Les agents SiftMatcher

Chaque message reçu en provenance des agents *SiftExtractor* contient un point SIFT de l'image et les agents *SiftMatcher* se chargent de rechercher le plus proche dans la base. Ils envoient ensuite la mise en correspondance à l'agent parent, avec comme priorité $-\log P_{H_0}(D_r \leq D_r(k))$ avec D_r le ratio entre la distance avec le point le plus proche de la base et la distance avec le second point le plus proche (voir section 3.3.2).

4.6.3 L'agent SiftClusterer

Cet agent reçoit les mises en correspondance des *SiftMatcher*. Pour chaque mise en correspondance, il calcule sa cellule dans l'espace de Hough généralisé (section 3.2.3). Ensuite, si une hypothèse a déjà été créée pour cette cellule, il y ajoute la mise en correspondance, sinon il crée une nouvelle hypothèse dans la mémoire collective. L'agent met ensuite à jour la *PFA* de l'hypothèse en calculant le nouveau terme $P_{H_0}(P_{H_0}^{Lowe} \leq P_{H_0}^{Lowe}(\mathcal{H})) \times P_{H_0}(P_{H_0}^{\mu} \leq P_{H_0}^{\mu}(\mathcal{H}))$ (voir section 3.3.5). Il envoie alors un message contenant l'hypothèse $\mathcal{H}$ mise à jour à l'agent parent *Main*, en utilisant $-\log \mathit{PFA}(\mathcal{H})$ pour fixer la priorité du message.

4.6.4 L'agent Main

Cet agent prend les décisions. Les messages qu'il reçoit contiennent une hypothèse $\mathcal{H}$ qui a été mise à jour récemment. Il regarde si sa *PFA* est inférieure au seuil de décision. Si oui, l'hypothèse donne lieu à une détection immédiate. Sinon, si son terme $P_{H_0}(D_{sad} \leq D_{sad}(\mathcal{H}))$ n'a pas été calculé, il envoie un message contenant l'hypothèse à son agent fils *SadComputer*. La priorité du message est donnée ici aussi par $-\log \mathit{PFA}(\mathcal{H})$.

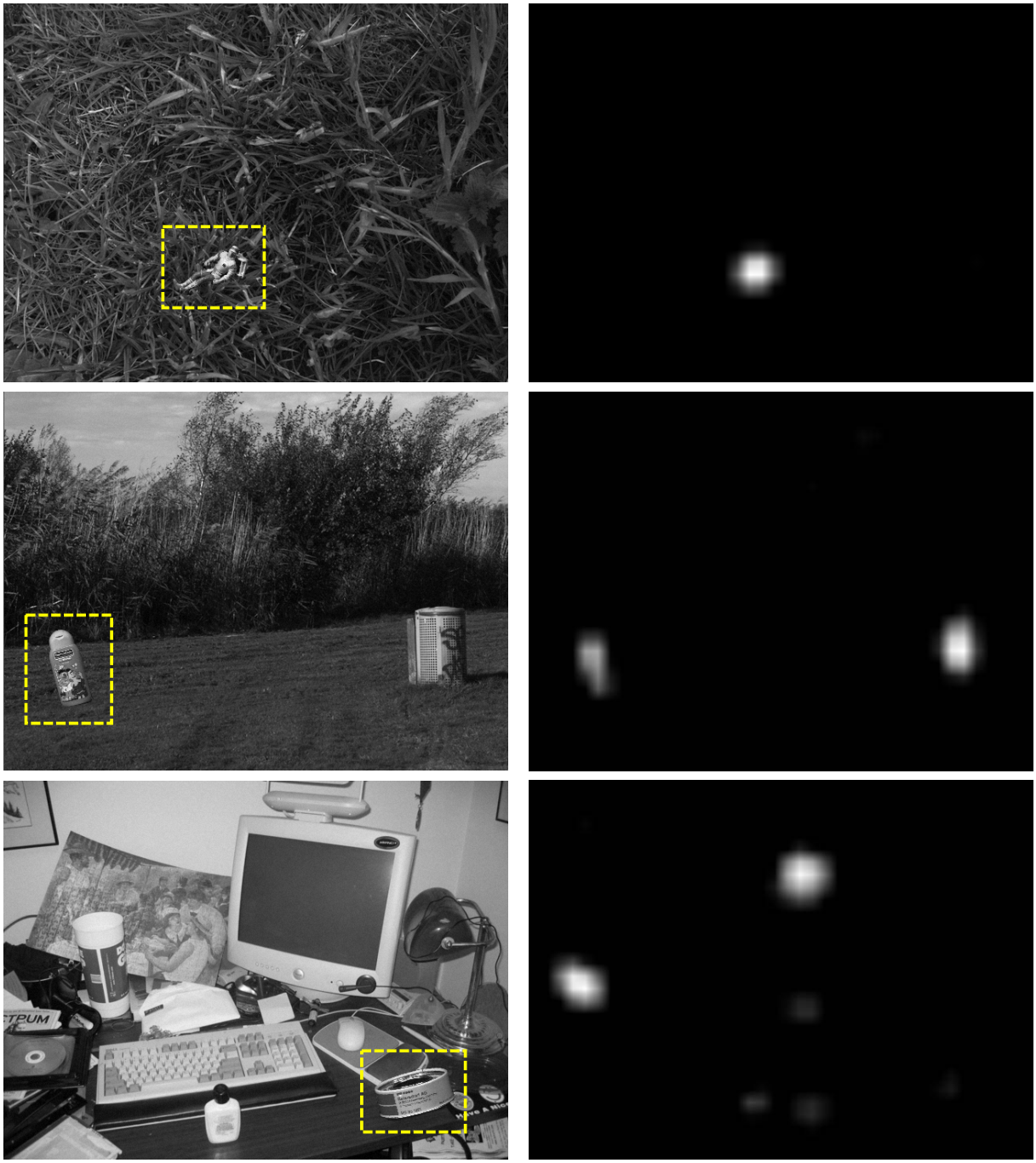


FIG. 4.4 – À gauche : images de test contenant un objet à détecter. À droite : cartes de saillance normalisées calculées par l’algorithme de [IKN+98], qui mesurent le contraste entre chaque région et sa zone environnante. Ces cartes sont très efficaces pour trouver *a priori* la position d’objets manufacturés dans des images en extérieur, car ils sont très différents de leur voisinage. En images d’intérieur, cette information est beaucoup moins utile, puisque les objets à détecter ne sont pas nécessairement plus saillants que les autres objets de la scène. Dans l’exemple du bas, les grandes zones sombres et contrastées associées à l’écran et au sac noir (à gauche) sont ainsi plus saillantes que l’objet artificiellement incrusté.

Il lance également des prédictions pour K points SIFT du modèle de l'hypothèse qui n'ont pas encore été mis en correspondance et pour lesquels des prédictions n'ont pas encore été émises. K est fixé empiriquement à 5 dans nos expériences. Pour choisir quels points du modèle vont donner lieu aux prédictions, ils sont classés par pouvoir discriminant et stabilité à l'aide d'une étape d'estimation hors-ligne préalable. Chaque modèle de la base est transformé par une rotation de 5 degrés, un ajout de bruit uniforme de 1% et une réduction d'échelle de $\frac{1}{2}$. Les points SIFT de chaque image sont ensuite mis en correspondance avec leur voisin le plus proche dans la base. À chaque point correctement associé est donné un score inversement proportionnel au ratio des distances D_r avec le second point le plus proche de la base. Les autres points SIFT ont un score nul. À la fin, chaque point a un score de répétabilité qui permet de les ordonner et de rechercher en priorité les points stables les plus discriminants.

4.6.5 L'agent SadComputer

Cet agent reçoit des hypothèses à analyser. Il calcule simplement le terme $P_{H_0}(D_{sad} \leq D_{sad}(\mathcal{H}))$ de la probabilité de fausse alarme de chaque hypothèse, puis renvoie l'hypothèse mise à jour à l'agent *Main*, avec comme priorité $-\log PFA(\mathcal{H})$.

4.7 Parallélisme spatial

Toutes les tâches ne sont pas aussi coûteuses en temps de calcul. Il peut alors être souhaitable d'équilibrer la répartition de la charge en multipliant le nombre d'agents dédiés à certaines tâches. Ceci permet également d'augmenter le nombre de fils d'exécution parallèles, et donc de tirer parti plus efficacement des architectures multiprocesseurs, comme nous le verrons dans la section 4.8.

Nous proposons pour cela d'ajouter un parallélisme spatial aux traitements en associant à chaque message une localisation ponctuelle. Pour les messages contenant un point SIFT, la localisation est définie par les coordonnées du point dans l'image. Pour les messages contenant une hypothèse, nous choisissons le centre du plus petit rectangle englobant les points SIFT de l'hypothèse. Il devient alors possible d'augmenter le nombre d'agents dédiés à une même tâche en les dupliquant, puis en restreignant chacun d'entre eux à une zone précise. Ils ne traitent alors que les messages dont la localisation appartient à leur zone.

Nous avons utilisé cette possibilité pour les tâches d'extraction et de mise en correspondance des points SIFT, qui sont les plus coûteuses. Ainsi dans l'architecture de la figure 4.3, nous avons utilisé quatre agents *SiftMatcher* et quatre agents *SiftExtractor* qui traitent chacun un quart de l'image, sans superposition. Ceci permet d'allouer une grande partie de la puissance de calcul à ces étapes de bas niveau.

4.8 Adéquation avec une architecture multiprocesseurs

Il est aujourd’hui généralement admis que l’évolution des architectures généralistes ne passera plus par l’augmentation de la fréquence de processeurs monolithiques, qui atteint ses limites, mais par la multiplication du nombre d’unités de traitement. L’inflexion des acteurs industriels est déjà réelle. La majorité des ordinateurs personnels commercialisés s’appuient sur des processeurs à deux voire quatre coeurs. Certaines consoles de jeux comme la Xbox 360 de Microsoft ou la Playstation 3 de Sony embarquent respectivement trois et huit coeurs de processeurs. Pour des applications graphiques, des architectures à plusieurs dizaines de processeurs sont devenues la norme. Des projets comme l’architecture Larrabee d’Intel [SCS+08] ou Tesla de NVIDIA [LNOM08] laissent penser que les architectures plus généralistes s’appuieront également sur plusieurs dizaines voire centaines de processeurs d’ici quelques années.

L’architecture logicielle proposée tire naturellement parti d’architectures matérielles à plusieurs processeurs scalaires. La division des tâches en agents permet une parallélisation implicite en ayant recours à des fils d’exécutions séparés (threads). Chaque agent s’exécutant dans son propre fil d’exécution, le nombre de processus d’exécutant en parallèle est égal au nombre d’agents. Nous reposons alors sur le système d’exploitation pour répartir les fils d’exécution sur chaque processeur. Pour tirer pleinement parti des possibilités d’une architecture matérielle à N coeurs, il faut cependant s’assurer qu’au moins N fils d’exécution sont actifs à tout moment, et donc qu’au moins N agents ont des messages à traiter.

Nous assurons cette propriété en surdimensionnant légèrement le nombre total d’agents par rapport au nombre de processeurs. La contrepartie est une perte théorique d’efficacité lorsque plus de N agents sont actifs en même temps à cause des changements de contexte pour passer d’un fil d’exécution à l’autre. Les systèmes d’exploitation actuels rendent cependant cette perte négligeable en pratique si le nombre de fils reste raisonnable. Nous obtenons ainsi de bonnes performances et un taux d’occupation des processeurs élevé avec l’architecture à 11 agents de la figure 4.3 implantée sur une machine avec 4 processeurs.

4.9 Évaluation du comportement “anytime”

L’architecture proposée implante les concepts présentés dans la section 4.3. Le flux ascendant basé sur les points SIFT permet de formuler rapidement des hypothèses, qui peuvent être complétées de façon descendante par l’agent *SadComputer*. Les influences descendantes servent également à prédire les futurs emplacements des caractéristiques locales pour accélérer le traitement des hypothèses prometteuses. Les agents s’exécutant en parallèle, le parallélisme hiérarchique est géré implicitement. Ainsi, des hypothèses peuvent être étudiées par l’agent *Main* bien avant que tous les points SIFT de l’image aient été mis en correspondance. Couplée avec les priorités associées aux messages, cette propriété permet à l’architecture de se focaliser à tout moment sur les hypothèses les plus prometteuses et donc de réussir à détecter les objets les plus saillants le plus rapidement possible.

Nous évaluons l'intérêt de notre approche en comparant le profil de performance que nous obtenons par rapport à celui de l'algorithme de [Low04]. Le protocole est identique à celui de la section 3.6, les objets d'une base de 200 objets sont artificiellement incrustés dans des images de test avec un changement de point de vue de 25 degrés, une hauteur ou largeur de 100 pixels, un bruit uniforme de 2% et une rotation planaire de 15 degrés. Nous n'utilisons en revanche que des images de test de taille 640×480 pour ne pas biaiser les temps de calcul. Les seuils de détection sont fixés dans tous les cas de façon à obtenir un taux moyen de fausses alarmes par image inférieur à 10^{-1}. Les temps sont mesurés sur un Intel Core 2 à quatre coeurs cadencés à 2.4 Ghz. Par souci d'équité et pour ne mesurer que les gains liés aux aspects “anytime”, la recherche des plus proches voisins dans la base a été parallélisée pour l'algorithme de [Low04] de façon à tirer pleinement parti des quatre processeurs. Dans les deux cas, Les points SIFT sont extraits par le binaire fourni par David Lowe. Cette implantation à vocation démonstrative est relativement lente (entre 1 et 2 secondes pour une image 640×480), le temps d'extraction n'est donc pas comptabilisé dans nos graphiques pour ne se focaliser que sur les traitements qui ne sont pas strictement identiques. Pour donner un ordre d'idée, des implantations rapides à base de GPU (Graphics Processing Unit) ou tirant parti de processeurs multicoeurs permettent aujourd'hui d'extraire les points SIFT d'une image 640×480 en moins de 40 ms [Bjö07, ZCZX08].

La figure 4.5 (gauche) montre les profils de performance obtenus pour des objets ayant une taille maximale de 100 pixels (configuration standard de la section 3.6) incrustés sur des images d'intérieur. Nous testons notre algorithme complet, et une version où les prédictions de points SIFT de l'agent *Main* ont été désactivées. Notre approche a globalement plus de calculs à faire, car les mises en correspondance sont toutes analysées et la mesure de corrélation D_{sad} est calculée pour chaque hypothèse. Le caractère “anytime” permet cependant de réduire très significativement les temps de détection par rapport à l'algorithme de [Low04]. Notons que ce dernier n'a pas un profil de performance en forme de marche brutale, car les temps de traitement dépendent du nombre de points SIFT de chaque image. Aucune détection ne peut être espérée en moins de 400 ms pour [Low04], alors que 50% des objets détectables ont déjà été détectés par notre approche pendant ce même délai ; quand après un peu plus d'une seconde [Low04] a détecté 50% des objets détectables, notre approche en a détecté 90%. Grâce au parallélisme hiérarchique, les premières mises en correspondance peuvent être regroupées très tôt, bien avant que tous les points SIFT de l'image aient été associés. Les premières détections arrivent donc très tôt, surtout si les points SIFT manquants sont prédits par l'agent *Main*. Ces détections correspondent aux objets qui contiennent beaucoup de points SIFT similaires à ceux de leur modèle dans la base.

Les gains sont encore plus significatifs lorsque les objets sont plus saillants, la figure 4.5 (droite) montre les profils de performance quand les objets incrustés ont une taille de 200 pixels. Les temps d'exécution de [Low04] sont inchangés, alors que ceux de notre algorithme sont nettement diminués, plus de 75% des objets sont cette fois détectés en moins de 400 ms.

Dans des images d'extérieur, la figure 4.6 montre que les cartes de saillance de [IKN+98] fournissent un *a priori* très intéressant sur la significativité d'un point SIFT (section 4.6.1). Cette information est très facile à intégrer dans notre architecture et permet une amélioration très nette du temps d'exécution.

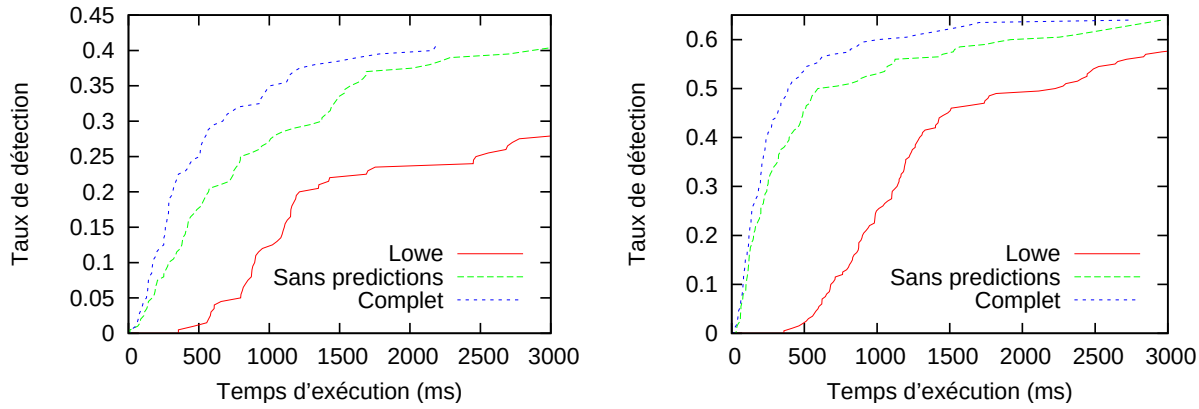


FIG. 4.5 – Profil de performance pour notre algorithme de détection, avec ou sans prédictions, comparé avec l’algorithme de [Low04]. Les objets sont incrustés sur des images d’intérieur de taille 640x480. Les objets sont incrustés avec une largeur ou hauteur maximale de 100 pixels à gauche, 200 pixels à droite.

4.10 Discussion

Nous avons montré que le cadre statistique *a contrario*, combiné avec une architecture logicielle adaptée, peut donner lieu à un algorithme de détection “anytime”. Cela a été expérimenté fructueusement à travers l’application de détection d’objets du chapitre 3. L’algorithme obtenu diminue significativement les temps de détection pour les objets les plus saillants et peut être interrompu à tout moment pour satisfaire des contraintes de temps réel ou limité. En se focalisant à tout moment sur les hypothèses les plus prometteuses, notre approche permet également d’améliorer les taux de détection en rendant inutiles les seuillages précoces et en rendant calculatoirement acceptable l’ajout de post-traitements descendants.

Il est d’autre part intéressant de constater que les propriétés architecturales motivées par un comportement “anytime” recoupent parfois les propriétés motivées par l’augmentation des taux de détection. La combinaison des influences ascendantes / descendantes est par exemple utile dans les deux cas. En effet, les prédictions de points SIFT utilisées ici pour accélérer les temps de traitements, peuvent également être utilisées pour mesurer des similarités locales additionnelles [KMY06].

L’aspect “anytime” de l’algorithme pourrait être amélioré de diverses manières. Par exemple, le décalage temporel entre les caractéristiques locales est basé uniquement sur l’ordonnancement des caractéristiques par la saillance de l’information qu’elles portent. D’autres sources de décalage temporel sont possibles et s’intègreraient facilement. Par exemple, dans [Jol03], les régions les plus saillantes sont également les plus rapides à extraire, car elles ont moins d’ambiguïtés locales. Elles peuvent donc être analysées plus tôt que les autres par les étapes de plus haut niveau.

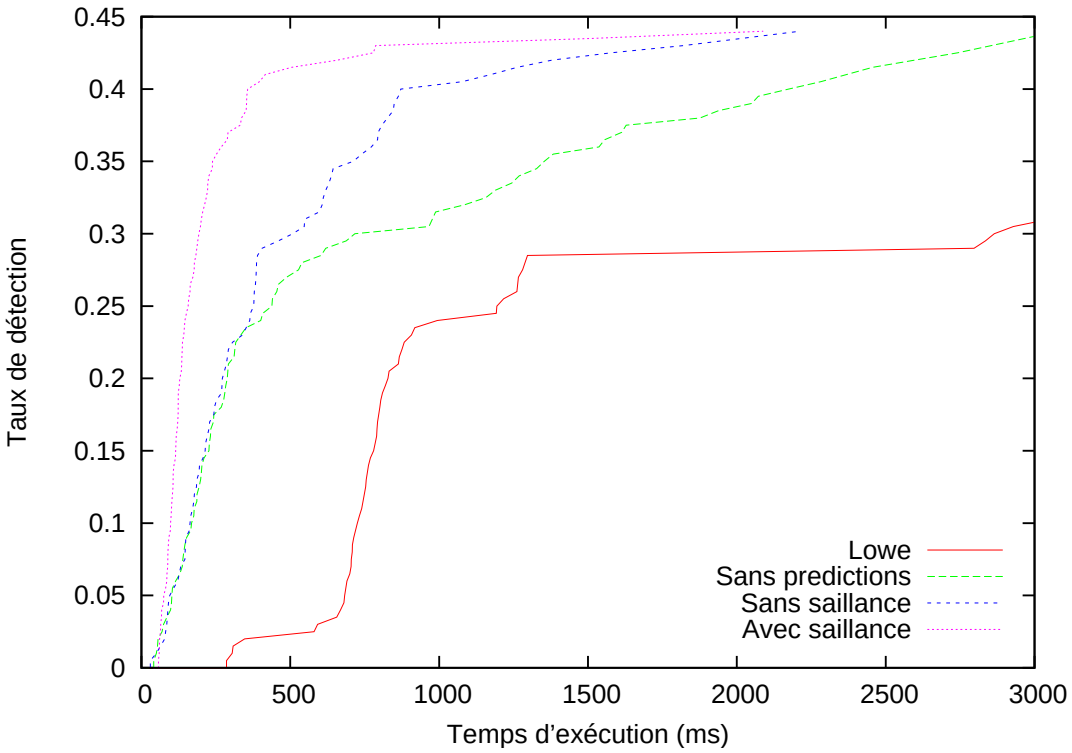


FIG. 4.6 – Profil de performance pour notre algorithme de détection comparé avec l'algorithme de [Low04]. Notre algorithme est décliné en quatre versions : avec ou sans prédictions et en utilisant ou pas les cartes de saillance de [IKN+98] pour estimer la priorité *a priori* des points SIFT. Les objets sont incrustés sur des images d'extérieur de taille 640×480 avec une largeur ou hauteur maximale de 100 pixels. Sur les images d'extérieur, les cartes de saillance permettent de trouver *a priori* l'emplacement de la plupart des objets, et donc de les détecter beaucoup plus rapidement, même si le calcul initial de la carte prend environ 40 ms.

Un autre aspect intéressant que nous avons laissé de côté dans ce travail est le choix dynamique des actions ou caractéristiques à calculer en fonction de l’état courant des connaissances. Grâce à la combinaison d’influences ascendantes et descendantes, les étapes de plus haut niveau pourraient optimiser les traitements en demandant aux étapes de bas niveau de calculer les données les plus utiles. On retrouve cette démarche dans plusieurs travaux, soit dans un but d’augmentation des taux de détection [MM01, PFS05] ou bien dans un but “anytime” [HL97, ZPG+04]. Notre architecture permettrait assez facilement d’intégrer ce type de comportement.

Concernant l’architecture logicielle proposée, l’organisation et la multiplication des agents pour équilibrer les charges de calcul et exploiter le parallélisme de l’architecture matérielle sous-jacente est pour le moment empirique, et statique : le nombre d’agents et donc de fils d’exécution n’évolue pas au cours du temps. De nombreuses améliorations sont possibles à ce niveau. Dans un premier temps, les agents pourraient disposer non plus d’un seul, mais de plusieurs fils d’exécution pour traiter plusieurs messages en parallèle. La charge pourrait alors être dynamiquement répartie en allouant plus ou moins de fils d’exécution à chaque agent, par exemple en fonction du nombre de messages en attente et du nombre de processeurs disponibles sur la machine. Cette répartition de la charge pourrait également faire l’objet d’une optimisation plus globale par apprentissage.

Finalement, pour valider la portée de la combinaison du cadre *a contrario* avec l’architecture logicielle proposée, il serait intéressant de l’appliquer à d’autres algorithmes de vision, en particulier à la détection de catégories d’objets, où la combinaison d’influences ascendantes/descendantes et l’utilisation d’informations partielles sont probablement encore plus utiles que pour la détection d’instances d’objets.

Conclusion

Après avoir mis en évidence dans le premier chapitre un certain nombre de limitations du cadre analytique *a contrario* existant, nous avons illustré à travers trois applications comment un recours à de l'apprentissage pouvait permettre d'assouplir les conditions d'utilisation et augmenter la portée de l'approche statistique *a contrario*. En particulier, nous avons montré que l'apprentissage permet d'utiliser simplement des mesures discriminantes multiples afin de capturer plusieurs propriétés des éléments à détecter. Les procédures de parcours de candidats peuvent également être moins rigides et s'appuyer sur les mesures discriminantes pour aiguiller leur choix et n'analyser que les candidats les plus prometteurs.

Après un travail de validation préliminaire sur une application de détection de segments, nous avons montré qu'un apprentissage *a contrario* à partir d'images de bruit blanc permettait d'apporter une solution efficace au problème de décision en segmentation d'image. En ne conservant que des régions adjacentes dont les différences ne sont pas statistiquement susceptibles d'apparaître dans du bruit blanc, nous avons proposé une méthode générique pour résoudre les problèmes de sur-segmentation d'algorithmes existants. La méthode ne comporte aucun paramètre libre autre que le nombre moyen de fausses alarmes tolérées. Les résultats obtenus confirment l'intérêt de l'approche *a contrario*, qui, même en n'utilisant aucun *a priori* quantitatif sur la scène, s'avère capable de discriminer de façon robuste les régions correspondant à un phénomène particulier de la scène.

Nous avons ensuite considéré une application de plus haut niveau, la détection d'instances d'objets à partir d'une base de photos. En nous appuyant sur la souplesse apportée par l'apprentissage *a contrario*, nous avons pu proposer un algorithme de détection combinant des mesures de similarité complémentaires, permettant d'améliorer significativement les taux de détection de l'approche originelle de [Low04]. Ici encore, le seul paramètre libre est le nombre moyen de fausses alarmes tolérées par image.

L'apprentissage n'est pas incompatible avec les calculs analytiques. Au contraire, nous avons utilisé à de nombreuses reprises des calculs analytiques pour approcher les distributions *a contrario* mises en jeu. L'apprentissage se limite alors à estimer les phénomènes difficiles à quantifier comme les dépendances entre mesures discriminantes ou l'influence des heuristiques d'exploration. L'apprentissage est donc considérablement simplifié : dans le cas de la détection d'objets, des estimations fiables ont pu être obtenues à partir de seulement dix images d'apprentissage, choisies aléatoirement.

Nous avons finalement exploité dans le chapitre 4 un autre aspect de l'approche *a contrario*, sa capacité à intégrer des informations partielles de façon monotone pour augmenter la confiance en la présence d'objets. Couplé avec une architecture logicielle adaptée, cette propriété nous a permis de proposer une implantation "anytime" de la détection d'instances d'objets. Nous avons pour cela introduit une architecture à base d'agents autonomes, capable de mener en parallèle les différentes étapes de traitement pour produire les premières détections bien avant l'achèvement de toutes les étapes intermédiaires. En autorisant à la fois des influences descendantes, sous forme de prédictions, et en se focalisant à tout moment sur les données les plus prometteuses, nous avons pu obtenir un algorithme de détection avec un bon profil de performance, c'est-à-dire avec un taux de détection qui augmente très rapidement en fonction du temps alloué.

Un certain nombre d'améliorations possibles liées à chaque application ont été évoquées dans les différents chapitres. Nous présentons ici des perspectives plus globales.

Premièrement, les chapitres 3 et 4 ont fait émerger des pistes pour construire une méthodologie complète de conception et d'implantation d'algorithmes de détection. Un processus souple a en effet été proposé pour intégrer des mesures discriminantes au sein d'un système de détection d'objets. Pour chaque mesure, un nouvel agent est introduit, et une approximation analytique de sa probabilité de fausse alarme peut éventuellement être fournie. Son intégration dans le système est ensuite relativement automatique, l'estimation précise de sa probabilité de fausse alarme et sa combinaison avec les autres mesures étant assurées par l'apprentissage *a contrario*. Il serait intéressant de valider et de généraliser cette démarche sur d'autres applications, pour obtenir, à terme, un système de vision plus générique pour la détection.

Un autre aspect, important à développer sur le long terme, est la question de l'intégration d'*a priori* dans le processus de détection. Poursuivant la démarche de [DMM08], nous avons cherché dans cette thèse à détecter des éléments en n'utilisant que le minimum d'informations *a priori* sur les structures à détecter. Si l'expérience montre que des résultats probants peuvent être obtenus dans cette voie, il est clair que pour obtenir des performances similaires au système visuel humain, il reste nécessaire d'intégrer de l'*a priori* sur le monde réel, comme l'illustre la figure 4.7. Dans ce but, nous pensons que l'approche *a contrario* peut jouer un rôle important en fournissant une première quantification perceptuelle de l'information, relativement universelle, qui pourrait ensuite être pondérée par un *a priori* sur l'image. Elle jouerait alors un rôle similaire aux calculs analytiques dans cette thèse. Nous les avons en effet utilisés à plusieurs reprises pour fournir une approximation des probabilités de fausse alarme, et donc simplifier considérablement l'apprentissage, qui peut alors se focaliser uniquement sur les phénomènes non pris en compte. De même, l'utilisation du cadre *a contrario* pour fournir une première approximation permettrait aux étapes suivantes de ne se focaliser que sur l'intégration de l'*a priori*.

Cet *a priori* pourrait éventuellement être déduit du contenu de l'image lui-même par les étapes de plus haut niveau, comme le propose [Tor03]. Notre architecture à base d'agents permettrait d'intégrer tous ces éléments de façon unifiée, l'*a priori* pouvant être utilisé à la fois pour pondérer les probabilités de fausse alarme associées aux différentes mesures, et pour orienter la recherche de candidats sous forme de prédictions, comme cela a déjà été – rudimentairement – expérimenté pour les points SIFT.

FIG. 4.7 – Illustration de l'intérêt de l'*a priori* en détection. Les deux images (extraites de [Tor03]) contiennent un même groupe de pixels. Selon son emplacement et son orientation, nous l'identifions comme une voiture ou un piéton. Seul un *a priori* sur le contenu de la scène et sur la nature des objets permet de distinguer ces deux cas.

Annexe A

Détection de segments significatifs sur rétine artificielle

Nous nous intéressons ici à l'implantation d'algorithmes de détection *a contrario* de primitives bas niveau pour des systèmes de vision embarqués et autonomes. La plupart des travaux *a contrario* existants requièrent une grande puissance de calcul, car un grand nombre de candidats doivent être analysés dans l'image. Par exemple, l'algorithme de détection de segments de la section 2.2 nécessite un Intel Core 2 cadencé à 2.4 Ghz pour pouvoir traiter plusieurs images par seconde. Le même constat peut être fait pour les autres algorithmes de détection *a contrario* bas niveau de la littérature, allant de presque 20 images par seconde pour le plus rapide [GJMR08], jusqu'à plusieurs dizaines de secondes par image pour les plus lents [DMM00b, Jak07], sur la même architecture. Pour s'approcher de performance temps réel sur une architecture classique, il est donc nécessaire de recourir à des processeurs puissants et cadencés à des fréquences élevées. Ces solutions ne sont pas toujours acceptables pour des systèmes autonomes, car elles consomment trop d'énergie. À titre d'exemple, un Intel Core 2 T9400 cadencé à 2.4 Ghz consomme de l'ordre de 35 Watts. Pour un Intel Quad Core cadencé à 3.3 Ghz, la consommation monte jusqu'à 150 Watts.

Ce coût énergique et calculatoire s'explique par la nature des traitements à effectuer. Les primitives de bas niveau sont généralement définies directement au niveau pixellique, et correspondent à des groupes de pixels partageant certaines propriétés. Les algorithmes de détection pour ce genre de primitives partagent un certain nombre de caractéristiques :

- Confrontation à une masse importante d'information. L'entrée est l'image brute.
- Candidats spatialement répartis. Les groupes de pixels à détecter peuvent être n'importe où dans l'image.
- Traitements par candidat limités : pour chaque groupe de pixels, il est seulement nécessaire de calculer quelques grandeurs.
- Indépendance des traitements entre candidats : l'analyse d'un candidat ne dépend pas de l'analyse d'un autre.

L'architecture scalaire classique est largement sous-optimale pour ce type de calculs. En effet, si les processeurs scalaires sont capables d'effectuer des calculs complexes rapidement, ils

ne traitent qu'une seule donnée à la fois. Le temps de calcul est donc directement dépendant du nombre de données. De plus, pour chaque donnée, les traitements sont relativement courts. Le processeur passe donc la plupart de son temps à charger de nouvelles données, et ces transferts sont à la fois coûteux en temps et en énergie.

Il semble plus naturel de s'orienter vers du parallélisme massif de type SIMD (Single Instruction Multiple Data), qui permet d'appliquer un traitement simple à un grand nombre de données différentes de façon efficace. L'utilisation de GPU (Graphics Processing Unit) s'est beaucoup développée ces dernières années et permet d'atteindre de très bonnes performances pour ce genre d'applications. Mais leur consommation d'énergie les rend également inadaptés à des systèmes de vision autonomes. À titre d'exemple, la Geforce 8800 GTS utilisée dans [Bjö07] pour une extraction en temps réel de points SIFT consomme de l'ordre de 250 Watts. Ce constat est amené à évoluer dans le futur, divers projets cherchant à intégrer des GPU dans des systèmes à faible consommation. Citons par exemple le projet Tegra de NVIDIA, qui propose un système sur puce intégrant un GPU avec une consommation typique de quelques Watts. Cette consommation s'obtient cependant, pour le moment, au prix d'un bridage significatif du GPU.

Nous nous intéressons dans ce chapitre à un autre type d'architecture, spécialement conçue pour la vision artificielle sur systèmes autonomes : les rétines artificielles. En cherchant à rapprocher à l'extrême données et unités de traitement, les rétines artificielles sont parvenues ces dernières années à obtenir un bien meilleur rapport puissance de calcul / consommation d'énergie pour des applications de vision de bas ou moyen niveau. L'idée générale consiste à réduire au maximum les transferts d'information en élaborant des capteurs d'images qui embarquent une unité de calcul simple directement au sein de chaque pixel. Les systèmes ainsi obtenus sont massivement parallèles, et l'image peut être traitée directement au sein du capteur, sans transferts de données. Pour les algorithmes adaptés, ce parallélisme massif et la réduction des transferts d'information permettent d'obtenir de très bonnes performances pour une consommation d'énergie très faible, de l'ordre de quelques dizaines de milliwatts. Le rapprochement capture-traitement ouvre également des perspectives intéressantes en terme d'intégration de système de vision sur puce.

Certaines rétines sont de plus capables de calculer très rapidement des grandeurs globales, qui permettent par exemple d'implanter du contrôle automatique de gain en ajustant les temps de capture en fonction de la luminosité de l'image [NDBG97]. Cette possibilité est également très intéressante pour l'implantation d'un algorithme de détection *a contrario* car elle permet de calculer efficacement des variables conditionnantes globales.

Pour ces raisons, nous avons expérimenté ce type d'architecture pour implanter un algorithme de détection *a contrario*. Il existe différents types de rétines artificielles, on se référera à [Elo05, Gie05, Moi00, Pai01] pour des études bibliographiques détaillées. Si les premières rétines étaient dédiées et le plus souvent analogiques, la majorité des projets actifs vise aujourd'hui à développer des rétines programmables, analogiques, numériques, ou les deux. Parmi les numériques, il s'en trouve de purement synchrones ou d'autres asynchrones. Au sein des rétines synchrones [PMB99, KKI04], tous les processeurs effectuent la même instruction à chaque tic d'horloge. Chaque processeur ne pouvant généralement communiquer

que de façon locale avec ses voisins immédiats, il faut alors n tics d'horloge pour échanger une donnée, par propagations successives, avec un pixel situé à une distance n. Il est donc relativement coûteux de calculer des grandeurs régionales. Pour pallier ce problème, plusieurs architectures autorisant également des communications asynchrones ont été proposées [DM02, Gie05, GBM06, LD06, WKKI05, YWKI07]. Cependant, pour le moment, aucune de ces architectures asynchrones n'a donné lieu à un prototype de taille raisonnable. Parmi les rétines synchrones, la plus grande réalisée à ce jour est la rétine Pvlsar34, développée à l'ENSTA par Thierry Bernard et qui a une taille de 200×200 pixels, suffisante pour une utilisation industrielle. Elle propose également un additionneur analogique qui permet d'estimer des mesures globales sur l'image rapidement, c'est pourquoi nous avons mené nos expérimentations sur ce modèle de rétine en utilisant comme application support la détection de segments de la section 2.2.

A.1 La rétine Pvlsar34

La rétine Pvlsar34 est une machine composée de 40 000 cellules organisées selon une grille 200×200, comme le montre la figure A.1. Chaque cellule contient, d'une part, un capteur photosensible et un convertisseur analogique/numérique pour l'acquisition d'image, et d'autre part un processeur booléen ainsi que 48 bits de mémoire pour les traitements. Tous les processeurs exécutent simultanément la même instruction, mais sur leurs propres données, il s'agit donc d'une machine massivement parallèle de type SIMD. Les processeurs sont connectés de façon 4-connexe : chaque processeur partage un bit de mémoire avec chacun de ses quatre voisins. Les processeurs peuvent donc interagir de façon locale.

La rétine peut être vue comme une machine permettant d'appliquer des fonctions booléennes aux bits d'une image. Elle est particulièrement adaptée à des opérations locales devant être appliquées sur tous les pixels : elle a été utilisée pour implanter en temps réel du filtrage morphologique [Man00], du calcul de squelettes [MBPL02], de l'extraction de points d'intérêts locaux [RM03] ou bien des cartes de saillance [RM07]. La capacité mémoire de chaque pixel permet de stocker plusieurs images successives, des algorithmes temporels ont donc également été proposés, pour détecter du mouvement dans [Ric06] ou pour créer une interface homme/machine visuelle dans [NMB06].

La rétine est pilotée par un processeur scalaire basse consommation, parfois appelé "cortex", qui envoie les instructions à exécuter. Le cortex peut également extraire des informations de la rétine, soit sous forme d'image binaire correspondant à un plan de bit particulier, soit sous forme de descripteur scalaire global correspondant à la somme des bits d'un plan particulier. Il n'est pas possible d'accéder directement aux valeurs des pixels de la rétine, la sortie sous forme d'image doit être effectuée séquentiellement par paquets de 8 pixels. Elle est donc coûteuse et n'est généralement utilisée qu'une seule fois à la fin des traitements ou bien pour le débogage des algorithmes. En revanche, les sommes globales sur l'image sont très rapides à calculer, ouvrant la voie à des algorithmes adaptatifs : le cortex peut envoyer des instructions à la rétine, effectuer des mesures globales sur l'image, puis adapter les instructions qui suivent en fonction

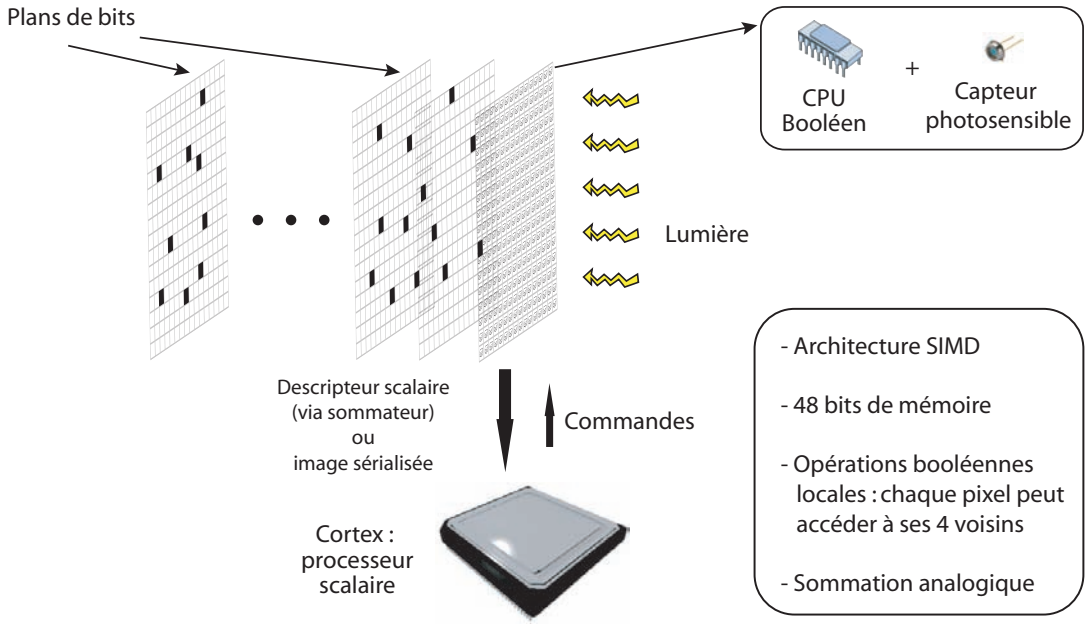


FIG. A.1 – Système de vision rétinien à base de Pvlsar34. La rétine est une grille de 200×200 cellules contenant à la fois un capteur photosensible pour l'acquisition d'image et un processeur booléen pour effectuer des traitements directement au coeur du pixel. La mémoire de 48 bits par cellule peut être vue globalement comme une succession de 48 plans de bits. De par son parallélisme SIMD, la rétine effectue donc des opérations globales sur des images binaires : à chaque bit de la mémoire est appliqué le même traitement au même moment. Des interactions locales sont possibles car les processeurs sont interconnectés de façon 4-connexe. La rétine est pilotée par un processeur scalaire externe, appelé "cortex", qui se charge de séquencer les instructions à envoyer à la rétine. Il peut également récupérer le résultat des calculs rétiniens en extrayant un plan de bits particulier ou bien le résultat de la sommation de tous les bits d'un plan.

de ces mesures. Nous allons utiliser ici ce rebouclage pour adapter l'extraction des segments à certaines variables conditionnantes globales, et ainsi ne conserver que les segments significatifs d'une image.

A.2 Application à la détection de segments

Nous rappelons tout d'abord les grandes étapes de l'algorithme de la section 2.2 :

1. Extraire huit images binaires correspondant à huit cônes de direction différents. Dans chacune de ces images, un pixel est allumé si le point correspondant dans l'image originale a une direction locale de gradient incluse dans le cône.
2. Pour chacune de ces images binaires, l'épaisseur des segments est réduite à un pixel dans la direction orthogonale en ne conservant que les pixels situés sur un maximum local des intensités de gradient.
3. Ne conserver que les segments suffisamment significatifs selon la variable discriminante choisie.

Les étapes 1. et 2. sont adaptées à un mode de calcul SIMD et sont facilement implantables sur Pvlsar34. L'étape 3. est en revanche plus délicate. Trois variables discriminantes ont été proposées dans la section 2.2. Le calcul d'une propriété telle que la longueur L, le minimum μ ou la moyenne λ de contraste sur un segment demande L itérations, puisque les pixels ne peuvent communiquer que de proche en proche. C'est donc assez coûteux. De plus, pour les critères de minimum et de moyenne de contraste, les *PFA* font intervenir la longueur du segment et la distribution G des intensités de gradient sur l'image. Comme le seuil de significativité est alors différent pour chaque segment, la *PFA* de chaque segment doit être calculée directement au sein de la rétine. Ceci n'est pas envisageable car les processeurs devraient alors conserver en mémoire G, L et μ ou λ, ce qui n'est pas possible avec une mémoire de 48 bits.

Le critère de longueur est en revanche envisageable, car il est alors possible de ne prendre en compte qu'une seule variable conditionnante, globale. Le seuil de significativité est alors le même pour tous les segments, et la procédure de sélection devient beaucoup plus simple. C'est ce qui a motivé son étude statistique dans le chapitre 2. Le critère de significativité pour un segment S est alors donné par le calcul suivant :

$$\mathit{PFA}_L(S) = P_{H_0}(L \geq L(S) \mid p_b) < \frac{\varepsilon}{N_s}$$

avec p_b la densité de pixels blancs et N_s le nombre total de segments dans l'image de direction du segment. Pour chaque image de direction localisée, le cortex peut donc déterminer la longueur minimale L_{min} assurant l'ε-fiabilité en fonction de p_b et N_s. Il suffit alors de ne conserver que les segments de l'image de longueur supérieure à L_{min}. L'algorithme complet finalement implantable sur Pvlsar34 est donné dans la figure 5. Nous en détaillons maintenant quelques points délicats.

Algorithme 5 : Algorithme de détection de segments adapté à Pvlsar34. $A \wedge B$ correspond au "et" binaire bit à bit entre les images A et B, et $A \vee B$ au "ou" binaire. Toutes les opérations sont menées sur la rétine, sauf celles qui sont précédées de la mention "(Cortex)". Le calcul de N_s et la suppression des segments trop courts seront détaillés dans les sections A.2.2 et A.2.3.

Données : Image I
Sortie : Image de segments S

$G \leftarrow$ gradient de I ;
$\{D_i\}_{i \in \{1...8\}} \leftarrow$ images de direction déduites de G ;
pour chaque *seuil de gradient* g_k **faire**
  Calculer le masque binaire M_k tel que $M_k(x, y) = 1$ si $|G(x, y)| \geq g_k$;
  pour chaque *image de direction* D_i **faire**
    $D_i^k \leftarrow D_i \wedge M_k$;
    Localiser D_i^k sur les crêtes de gradient ;
    (Cortex) Calculer $p_b = \frac{1}{200 \times 200} \sum_{x,y} D_i^k(x, y)$;
    $E_i^k \leftarrow$ extrémités des segments de D_i^k ;
    (Cortex) Calculer $N_s = \sum_{x,y} E_i^k(x, y)$;
    (Cortex) Calculer la longueur minimale L_{min} pour qu'un segment soit significatif, en fonction de p_b et N_s ;
    Supprimer les segments de longueur inférieure à L_{min} dans D_i^k ;
    $S \leftarrow S \vee D_i^k$;
  fin
fin

A.2.1 Calcul des seuils de gradient

L'algorithme de détection est appliqué sur une succession d'images de direction où seuls les pixels de plus en plus contrastés sont conservés. Ainsi, pour chaque pourcentage δ, il faut être capable de déterminer le seuil g_{min} sur l'intensité de gradient qui permet de ne conserver que les $\delta\%$ de pixels les plus contrastés. Sur une architecture classique, g_{min} peut être déduit rapidement en utilisant la distribution des intensités de gradient sur l'image.

Sur la rétine, la distribution du gradient peut être calculée par le cortex en ayant recours à des seuillages successifs et des sommations globales. Pour chaque valeur possible g, il suffit de calculer une image binaire I_g où un pixel vaut 1 si son intensité de gradient est supérieure ou égale à g. $P(G \geq g)$ est alors estimée en comptant le nombre de pixels dans I_g grâce au sommateur analogique. Cette procédure est cependant coûteuse, car s'il y a N valeurs possibles pour les intensités de gradient, il faut appliquer N seuils et sommations successifs.

C'est pourquoi nous utilisons une version simplifiée dans nos expérimentations, en utilisant une série de $\log_2(N)$ seuils $g_k = 2^k$ avec k compris entre 0 et $\log_2(N) - 1$. Les puissances de 2 permettent de déterminer très efficacement si une intensité de gradient est suffisamment grande à l'aide de processeurs booléens : une intensité est supérieure à 2^k si au moins un des $\log_2(N) - k$ bits de poids fort vaut 1.

A.2.2 Calcul du nombre de segments candidats

À partir d'une image de direction, le nombre de segments candidats peut être calculé en comptant le nombre d'extrémités, comme le montre la figure A.2 pour la direction horizontale. Les pixels extrémités peuvent être extraits par une transformée morphologique en tout-ou-rien [Ser83], déduite de la définition de segment de la section 2.2.2.

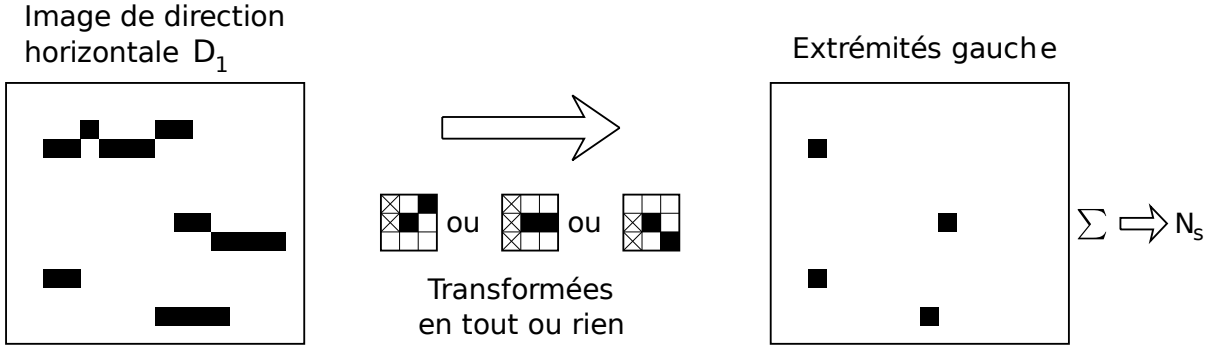


FIG. A.2 – Calcul du nombre de segments candidats dans une image de direction localisée. La première extrémité de chaque segment (ici, l'extrémité gauche) est tout d'abord extraite grâce à des transformées morphologiques en tout-ou-rien. Le nombre de segments N_s peut ensuite être directement estimé grâce au sommateur analogique.

A.2.3 Élimination des segments trop courts

Les segments de longueur inférieure au seuil L_{min} calculé par le cortex peuvent être supprimés de façon itérative sans explicitement calculer la longueur de chaque segment. Soit D_i l'image de direction concernée. L'idée est de supprimer itérativement les extrémités des segments de D_i. Après L_{min} itérations, tous les segments de longueur inférieure à L_{min} ont alors disparu. On appelle I_L l'image obtenue. L'image de direction ne contenant plus que les segments suffisamment grands correspond finalement à l'ouverture morphologique par reconstruction [Ser83] de D_i par I_L, comme le montre la figure A.3.

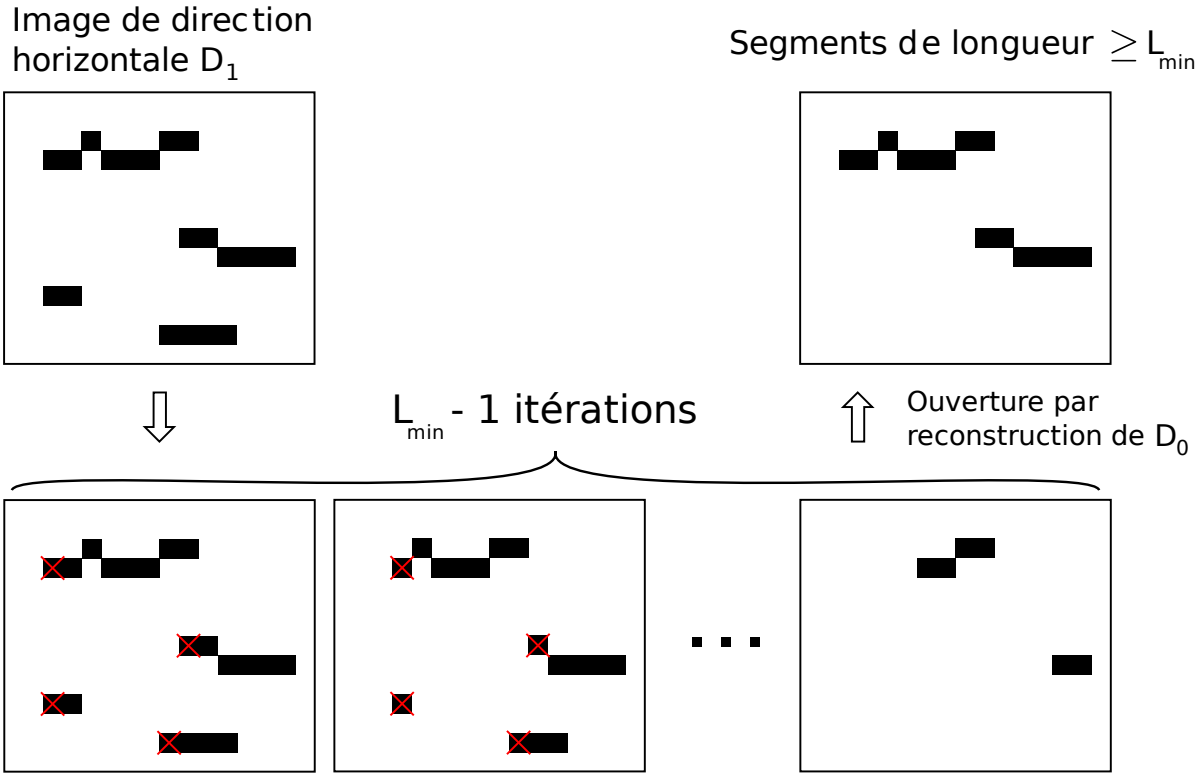


FIG. A.3 – Étape de suppression des segments de longueur inférieure au seuil L_{min} déterminé par le cortex. $L_{min} = 5$ dans l'exemple. Les segments sont érodés itérativement par leur première extrémité. Après $L_{min} - 1$ itérations, seuls les segments de longueur supérieure ou égale à L_{min} sont encore (partiellement) présents. Avec une ouverture morphologique par reconstruction de l'image de direction originale, on obtient finalement une image ne contenant que les segments originaux dont la longueur est supérieure ou égale à L_{min}.

A.3 Résultats et discussion

Pour un seul seuil de gradient, le temps de traitement obtenu sur Pvlsar34 est d'environ 40 ms, soit 20 images par seconde. Si les six seuils sont utilisés, environ 5 images par seconde peuvent être traitées, soit approximativement les mêmes performances que celles d'un

Intel Core 2 avec quatre coeurs cadencés à 2.4 Ghz. Le principal intérêt de l'architecture dans cette application est donc la réduction de la consommation énergétique de plusieurs ordres de grandeur, ainsi que la réduction de l'encombrement.

Les temps de calcul obtenus peuvent toutefois paraître décevants en regard du parallélisme massif mis en oeuvre. Il y a deux explications à cela. La première, c'est que les processeurs booléens sont cadencés à seulement 5 Mhz pour économiser le maximum d'énergie. La seconde rappelle les limitations d'une architecture purement synchrone, qui n'est réellement efficace que lorsque tous les pixels font un travail effectif à chaque instruction. Ce n'est pas toujours le cas ici, car les segments sont des primitives régionales. En particulier, pour le rognage itératif qui supprime les segments trop courts, seuls les pixels situés aux extrémités des segments font réellement un travail utile à chaque itération. Tous les autres ne changent pas l'état de leur mémoire. Nous avons donc atteint ici les limites du modèle synchrone.

En revanche, des performances bien meilleures sont à attendre des générations de rétine à venir, qui permettent de combiner calculs synchrones et asynchrones. Par exemple, l'architecture proposée par [Gie05] permet de calculer des sommes régionales sur des ensembles de pixels connectés sans synchronisation externe, et donc très rapidement. Le calcul de variables discriminantes ou conditionnantes régionales pourrait ainsi être beaucoup plus efficace. Dans [YWKI07], l'évolution des processeurs élémentaires et l'introduction d'une mémoire partagée permettent également d'envisager des calculs de *PFA* directement au sein de la rétine, capables de prendre en compte efficacement des variables conditionnantes globales et régionales. Ceci ouvre des perspectives très intéressantes pour l'implantation temps réel d'algorithmes de détection *a contrario* de primitives régionales sur systèmes de vision autonomes.

Annexe B

Preuve de la proposition 3

Soit $W = \{w_1, w_2, \ldots, w_n\}$ l'ensemble de tous les couples de régions possibles dans une image $N \times M$. $\#W$, le cardinal de W, est très grand et non calculable analytiquement, mais il ne dépend que de N et M. Nous considérons qu'une image est une observation d'un certain modèle, par exemple d'un modèle de bruit blanc uniforme. Par extension, chaque région d'une image est vue comme une observation d'un certain modèle régional. Nous introduisons tout d'abord quelques notations :

- Si W est extrait d'une image qui est le résultat du modèle *a contrario* où les pixels sont indépendants et identiquement distribués (i.i.d.), on le note par $H_0(W)$. Si de plus les pixels sont uniformément distribués dans le modèle sous-jacent, on le note par $H_0^U(W)$.
- Si un couple de régions w_i est le résultat du modèle *a contrario* où les pixels sont indépendants et identiquement distribués (i.i.d.), on le note par $H_0(w_i)$. Si de plus les pixels sont uniformément distribués dans le modèle sous-jacent, on le note par $H_0^U(w_i)$.
- $\mathcal{H}(w_i)$ signifie que le couple de régions w_i est analysé par l'heuristique $\mathcal{H}$.
- $S_\delta(w_i)$ signifie que le couple de régions w_i est significativement différent d'après S_δ.
- L'espérance d'une variable aléatoire X est notée $\mathbb{E}(X)$.
- $\#\Omega$ est le cardinal d'un ensemble Ω.
- L'espérance du nombre de fausses alarmes produites par l'algorithme $\mathcal{A}$ sur W est notée :

$$\mathbb{E}_W = \mathbb{E}\left(\#\{w_i \in W; \mathcal{H}(w_i) \text{ et } S_\delta(w_i) \text{ et } H_0(w_i)\}\right)$$

  Il s'agit de l'espérance du nombre de couples de régions analysés par $\mathcal{H}$ et considérés significativement différents par S_δ, alors qu'ils sont en réalité le résultat du modèle *a contrario*.

Nous rappelons maintenant les conditions de la proposition 3 :

1. $\mathcal{A}$ est ε-fiable pour des images de bruit blanc uniforme :

$$\mathbb{E}_W \mid H_0^U(W) < \varepsilon$$

2. $P(S_\delta(w_i) \mid H_0(w_i)) \leq P(S_\delta(w_i) \mid H_0^U(w_i))$
3. $P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0(w_i)) \leq P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0^U(w_i))$

4. $P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0(W)) = P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0(w_i))$ et en particulier $P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0^U(W)) = P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0^U(w_i))$

Nous cherchons à démontrer que sous ces conditions, $\mathcal{A}$ est ε-fiable pour des images arbitraires, i.e. que $\mathbb{E}_W < \varepsilon$. Soit X^i la variable de Bernoulli qui vaut 1 quand une fausse alarme se produit, i.e. quand $\mathcal{H}(w_i)$, $S_\delta(w_i)$ et $H_0(w_i)$ se produisent simultanément. En utilisant la linéarité de l'espérance :

$$\begin{aligned} \mathbb{E}_W \mid H_0^U(W) &= \mathbb{E}\left(\sum_{i=1}^{\#W} X^i \mid H_0^U(W)\right) \\ &= \sum_{i=1}^{\#W} P_{H_0^U}^{X^i} \end{aligned}$$

où $P_{H_0^U}^{X^i} = P(X^i = 1 \mid H_0^U(W)) = P(\mathcal{H}(w_i), S_\delta(w_i), H_0(w_i) \mid H_0^U(W))$.

W est extrait d'une image de bruit blanc uniforme, nous savons donc que tous les couples de régions w_i sont localement issus d'un modèle de bruit blanc uniforme, et donc du modèle *a contrario* :

$$\begin{aligned} P_{H_0^U}^{X^i} &= P(\mathcal{H}(w_i), S_\delta(w_i) \mid H_0^U(W)) \\ &= P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0^U(W)) \times P(S_\delta(w_i) \mid H_0^U(W)) \end{aligned}$$

En utilisant les conditions (4) puis (3) :

$$\begin{aligned} P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0^U(W)) &= P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0^U(w_i)) \\ &\geq P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0(w_i)) \end{aligned}$$

Étant donné le modèle d'un couple de régions, les différences observées ne dépendent pas du modèle global de l'image :

$$P(S_\delta(w_i) \mid H_0^U(W)) = P(S_\delta(w_i) \mid H_0^U(w_i))$$

En utilisant la condition (2) :

$$P(S_\delta(w_i) \mid H_0^U(w_i)) \geq P(S_\delta(w_i) \mid H_0(w_i))$$

Et donc :

$$P_{H_0^U}^{X^i} \geq P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0(w_i)) \times P(S_\delta(w_i) \mid H_0(w_i))$$

On développe maintenant le calcul de $\mathbb{E}_W$ dans le cas général :

$$\mathbb{E}_W = \sum_{i=1}^{\#W} P(X^i = 1)$$

Avec :

$$\begin{aligned} P(X^i = 1) &= P(\mathcal{H}(w_i), S_\delta(w_i), H_0(w_i)) \\ &= P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0(w_i)) \\ &\quad \times P(S_\delta(w_i) \mid H_0(w_i)) \\ &\quad \times P(H_0(w_i)) \end{aligned}$$

La probabilité *a priori* $P(H_0(w_i))$ est inconnue, mais inférieure à 1, donc :

$$\begin{aligned} P(X^i = 1) &< P(\mathcal{H}(w_i) \mid S_\delta(w_i), H_0(w_i)) \times P(S_\delta(w_i) \mid H_0(w_i)) \\ &< P_{H_0^U}^{X^i} \\ &< P(X^i = 1 \mid H_0^U(W)) \end{aligned}$$

On en déduit que $\mathbb{E}_W < \mathbb{E}_W \mid H_0^U(W)$ et donc, d'après la condition (1), $\mathbb{E}_W < \varepsilon$. $\square$

Ceci nous permet de conclure que les quatre conditions de la proposition 3 sont suffisantes pour garantir l'ε-fiabilité que nous recherchons.

Annexe C

Estimation de queues de distributions *a contrario* empiriques

Nous avons proposé à plusieurs reprises dans les chapitres 2 et 3 d'estimer la distribution d'une variable discriminante X sous l'hypothèse *a contrario* H_0 de façon empirique. Pour chaque variable X, il s'agit alors d'estimer $P_{H_0}(X \leq x)$ ou $P_{H_0}(X \geq x)$ à partir d'un échantillon de valeurs observées sur un ensemble d'images d'apprentissage. Pour les valeurs "raisonnables" de X, cette estimation peut être réalisée de façon classique. Soit $\Omega = \{X_1, X_2, \ldots, X_n\}$ l'ensemble des valeurs de X observées pendant l'étape d'apprentissage. L'estimation empirique de $P_{H_0}(X \leq x)$ (le cas $P_{H_0}(X \geq x)$ est similaire) est alors donnée par :

$$\hat{P}_{H_0}(X \leq x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{X_i \leq x}$$

avec $\mathbb{1}_{X_i \leq x}$ la fonction indicatrice qui vaut 1 si la valeur X_i est inférieure ou égale à x.

Cette estimation n'est cependant satisfaisante que pour les valeurs de x typiques sous l'hypothèse *a contrario*, et donc observables pendant l'étape d'apprentissage. Plus les valeurs de x se rapprochent des valeurs minimales de Ω, moins l'estimation est fiable, et pour des valeurs de x inférieures à la plus petite des valeurs de Ω, la distribution empirique devient même nulle. Ceci est problématique car les variables discriminantes ont précisément été choisies pour que leurs valeurs en présence d'un objet soient significativement plus petites que celles qui peuvent apparaître par hasard. Il est donc nécessaire de fournir une estimation de $P_{H_0}(X \leq x)$ pour des valeurs de x très faibles.

Une solution classique dans ce cas consiste à modéliser la distribution de probabilité de X sous H_0 par une loi paramétrique, par exemple de type gaussienne, exponentielle, etc. Les paramètres sont alors estimés à partir de Ω. Il est toutefois difficile de déterminer des modèles paramétriques adaptés à chaque variable, et le choix du modèle introduit nécessairement un *a priori* discutable.

Il n'est cependant pas indispensable de modéliser toute la distribution de X, seule la queue gauche de la distribution est problématique ici. Les propriétés de la queue d'une distribution sont l'objet d'un champ de recherche particulier, appelé théorie des valeurs extrêmes ou théorie des grandes déviations [Cas88], très utilisée en finance ou en hydrologie pour estimer la probabilité d'occurrence d'évènements rares comme des inondations ou des krachs boursiers. Cette théorie permet notamment de modéliser le comportement de la queue d'une distribution avec très peu d'*a priori* sur la distribution sous-jacente, c'est donc l'approche que nous avons choisie.

Le résultat fondamental de cette théorie est analogue au théorème central-limite pour la valeur moyenne d'un échantillon. Soit un échantillon de n valeurs tirées indépendamment selon la loi d'une variable X. Il est bien connu en probabilité classique que la loi de la moyenne de l'échantillon tend vers une loi normale quand n tend vers $+\infty$, quelle que soit la loi de X. De même, il a été prouvé [ESM97] que la loi de la valeur maximale de l'échantillon ne peut converger que vers trois familles de lois, celle de Gumbel, celle de Fréchet ou celle de Weibull. Ce théorème nécessite quelques conditions sur la loi de X que nous ne détaillons pas ici, mais qui sont vérifiées pour la plupart des distributions. Il suffit intuitivement qu'elles soient suffisamment régulières. La famille "attractrice" dépend de la distribution de X et plus précisément du comportement de la queue de la distribution de X. La famille de Gumbel regroupe les distributions à queue exponentielle, parfois qualifiée de "modérée" (loi normale, binomiale, exponentielle, ...), la famille de Fréchet regroupe les distributions à queue large (loi de Student, loi log-gamma, ...), et celle de Weibull les distributions à queue étroite avec une limite supérieure finie (loi uniforme, loi beta, ...).

Il découle de ce résultat que pour la plupart des distributions, il n'y a que trois familles de modèles possibles pour sa queue. [ERS99] propose une méthode simple, automatique et générale pour estimer le meilleur modèle pour la queue d'une variable X à partir d'un échantillon. Le principe est le suivant. Soit u un seuil supérieur marquant le "début" de la queue gauche de X. Son choix sera discuté plus loin. Pour toute valeur de X inférieure à u, on note $y = u - X$ l'écart entre X et u. La fonction de répartition complémentaire $F_u(y)$ des écarts, sachant que X est inférieure à u, s'écrit :

$$F_u(y) = P(u - X \geq y \mid X \leq u)$$

avec $0 \leq y \leq u - x_0$ où x_0 représente la limite inférieure gauche de X, possiblement infinie. $F_u(y)$ représente la probabilité qu'une valeur de X inférieure à u en soit éloignée d'une distance supérieure à y. Elle est équivalente à la distribution de X pour les valeurs inférieures à u, car on peut passer facilement d'une formulation à l'autre :

$$P(X \leq x \mid X \leq u) = F_u(u - x)$$

L'intérêt de raisonner avec F_u est que l'on peut alors prouver [ERS99] que lorsque u tend vers la limite inférieure de X, F_u converge vers la fonction de répartition complémentaire de la loi de Pareto généralisée $G_{\xi,\beta}$:

$$G_{\xi,\beta}(y) = \begin{cases} \exp(\frac{-y}{\beta}) & \text{si } \xi = 0, \\ \\ (1 + \xi\frac{y}{\beta})^{\frac{-1}{\xi}} & \text{si } \xi \neq 0. \end{cases}$$

où $\beta > 0$, $y \geq 0$ si $\xi \geq 0$ et $0 \leq y \leq -\frac{\beta}{\xi}$ si $\xi < 0$.

Le paramètre de forme ξ permet de reproduire les trois comportements possibles pour la queue d'une distribution : $\xi = 0$ correspond à une distribution de X à queue exponentielle, $\xi > 0$ à une distribution à queue large, et $\xi < 0$ à une distribution à queue étroite avec une limite inférieure finie égale à $u + \frac{\beta}{\xi}$. Le paramètre β est quant à lui un paramètre d'échelle.

Si nous parvenons à estimer les paramètres de $G_{\xi,\beta}(y)$, nous obtenons directement une estimation de $P(X \leq x)$ pour les valeurs de X inférieures à u. En pratique, [ERS99] propose d'estimer les paramètres de $G_{\xi,\beta}(y)$, et donc de la queue de X, en utilisant les plus petits échantillons de Ω. Il suffit pour cela de choisir u suffisamment grand pour qu'il existe suffisamment d'observations X_i dans Ω qui soient inférieures à u. On peut alors déduire une variable d'écart $Y_i = u - X_i$ pour chaque $X_i < u$, et on peut déterminer ξ, β en maximisant la vraisemblance des écarts Y_i pour la distribution de Pareto généralisée. Cette procédure estime donc un modèle de la queue de X à partir des échantillons de Ω inférieurs à u. L'estimation peut être partiellement contrainte si l'on sait que X est bornée par une limite inférieure x_0 finie. C'est le cas pour le ratio de distances D_r et pour la mesure de corrélation D_{sad} du chapitre 3. Dans ce cas, on sait que ξ doit être inférieur strictement à 0 et que β doit être égal à $(x_0 - u) \times \xi$ pour que la limite soit respectée. Aucun autre *a priori* n'est nécessaire sur la distribution de X.

Il n'y a pas de méthode immédiate pour choisir le meilleur seuil u. Il doit être suffisamment grand pour que l'estimation des paramètres soit fiable, et suffisamment petit pour que l'approximation par la loi de Pareto généralisée soit bonne. Pour notre application nous avons empiriquement fixé u au premier pourcentile de Ω, c'est à dire à la valeur $X_i \in \Omega$ maximale telle que $\hat{P}_{H_0}(X \leq X_i)$ soit inférieure à $\frac{1}{100}$. Nous utilisons donc les 1% des échantillons les plus faibles pour estimer ξ et β, ce qui représente quelques centaines de valeurs pour un ensemble d'apprentissage de 10 images. Cette valeur de u s'avère également suffisante en pratique pour que l'approximation par la loi de Pareto soit bonne.

Finalement, la distribution *a contrario* de X peut être estimée pour toutes les valeurs de X en combinant l'estimateur empirique et le modèle $G_{\xi,\beta}$:

$$P_{H_0}(X \leq x) = P_{H_0}(X \leq x \mid X \leq u) \times P_{H_0}(X \leq u) + P_{H_0}(X \leq x \mid X > u) \times P_{H_0}(X > u)$$

$P_{H_0}(X \leq x \mid X \leq u)$ est calculé par $G_{\xi,\beta}(u - x)$, et les autres termes sont calculés par l'estimateur empirique $\hat{P}_{H_0}$. Ceci revient à utiliser l'estimateur empirique pour les valeurs de x supérieures à u, et le modèle $G_{\xi,\beta}$ pour les valeurs inférieures à u.

La figure C.1 montre les estimations ainsi obtenues pour les variables discriminantes du chapitre 3. Nous observons les trois familles de modèles possibles pour la queue d'une distribution. Pour D_r et D_{sad} (haut), il s'agit d'une distribution à queue étroite car la limite inférieure de la variable est finie. Pour $P_{H_0}^{Lowe}$ et $P_{H_0}^{\mu}$ (au milieu), les distributions ont une queue assez large ce qui se traduit par un paramètre de forme ξ estimé positif. Enfin, la queue de $P_{H_0}^{*}$ (en bas) est estimée exponentielle, donc linéaire sur l'échelle logarithmique, ce qui correspond au paramètre $\xi = 0$.

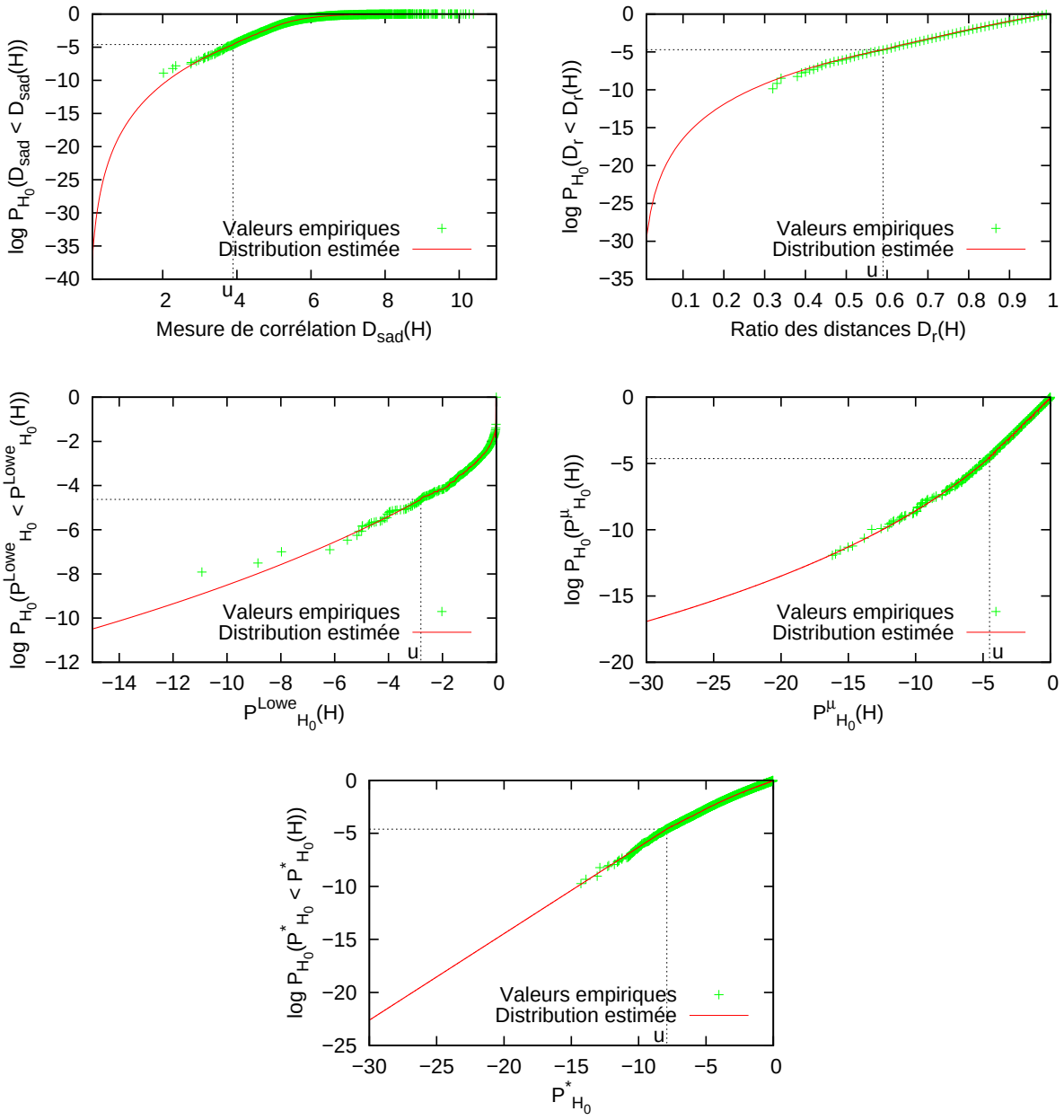


FIG. C.1 – De gauche à droite et de haut en bas : distributions empiriques et distributions estimées pour les variables $P^{Lowe}_{H_0}$, D_r, $P^{\mu}_{H_0}$, D_{sad} et $P^{*}_{H_0}$ sur l'ensemble d'apprentissage en intérieur. Pour chaque variable, la distribution estimée est égale à la distribution empirique jusqu'à la valeur u où le logarithme de la distribution vaut $\log(\frac{1}{100}) \simeq -4.6$ (en pointillé). En deçà, la distribution est estimée grâce à la loi de Pareto généralisée.

Les estimations obtenues sont visuellement satisfaisantes, confirmant l'hypothèse que le seuil u choisi est suffisamment faible pour que l'approximation par la loi de Pareto généralisée soit bonne. Il faut cependant relativiser l'importance des erreurs d'estimation. Une bonne précision n'est nécessaire que pour les valeurs de X encore susceptibles d'être le résultat du hasard, c'est à dire légèrement plus faibles que les valeurs minimales observées dans Ω. Au-delà, les valeurs de X seront quasi systématiquement associées à la présence d'un objet, et une estimation plus grossière est alors suffisante.

Bibliographie

[ACV07] A. Auclair, L. D. Cohen, and N. Vincent. How to use SIFT vectors to analyze an image with database templates. In *5th International Workshop on Adaptive Multimedia Retrieval, Paris, France*, 2007.

[ADV03] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 :502–507, 2003.

[AFDMar] A. Angeli, D. Filliat, S. Doncieux, and J.A. Meyer. Real-time visual loop-closure detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008 (to appear).

[AO07] Tomasz Adamek and Noel E. O'Connor. *Semantic Multimedia*, chapter Stopping Region-Based Image Segmentation at Meaningful Partitions, pages 15–27. Springer, 2007.

[Bal81] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2) :111–122, 1981.

[BAP07] R. Brooks, T. Arbel, and D. Precup. Anytime similarity measures for faster alignment. *Computer Vision and Image Understanding*, 110(3) :378–389, 2007.

[Bar03] M. Bar. A cortical mechanism for triggering top-down facilitation in visual object recognition. *Journal of Cognitive Neuroscience*, 15(4) :600–609, 2003.

[BCFM00] J. Batlle, A. Casals, J. Freixenet, and J. Martì. A review on strategies for recognizing natural objects in colour images of outdoor scenes. *Image and Vision Computing*, 18(6-7) :515–530, 2000.

[BD94] O. Boissier and Y. Demazeau. MAVI : a multi-agent system for visual integration. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 731–738, 1994.

[BFVG06] H. Bay, B. Fasel, and L. Van Gool. Interactive museum guide : fast and robust recognition of museum objects. In *Proceedings of the First International Workshop on Mobile Vision*, 2006.

[BHR86] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4) :425–455, 1986.

[Bie87] I. Biederman. Recognition-by-components : A theory of human image understanding. *Psychological Review*, 94(2) :115–147, 1987.

[Bjö07] M. Björkman. CUDA implementation of SIFT, 2007.

[BL97] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1006, 1997.

[BM93] S. Beucher and F. Meyer. The morphological approach to segmentation : the watershed transformation. *Mathematical Morphology in Image Processing*, pages 433–481, 1993.

[BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 :509–522, 2002.

[BP93] R. Brunelli and T. Poggio. Face recognition : features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10) :1042–1052, 1993.

[BSU04] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Workshop*, volume 4, pages 46–46, 2004.

[Cas88] E. Castillo. *Extreme value theory in engineering*. Academic Press, 1988.

[CC02] S. Chambon and A. Crouzil. Evaluation et comson de mesures de correlation robustes aux occultations. *Rapport de recherche*, 2002.

[CC04] S. Chambon and A. Crouzil. Towards correlation-based matching algorithms that are robust near occlusions. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, pages 20–23, 2004.

[CDD+07] F. Cao, J. Delon, A. Desolneux, P. Musé, and F. Sur. A unified framework for detecting groups and application to shape recognition. *Journal of Mathematical Imaging and Vision*, 27(2) :91–119, 2007.

[CL94] Y.L. Chang and X. Li. Adaptive image region-growing. *IEEE Transactions on Image Processing*, 3(6) :868–872, 1994.

[CL97] D. Crevier and R. Lepage. Knowledge-based image understanding systems : A survey. *Computer Vision and Image Understanding*, 67(2) :161–185, 1997.

[CM02] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24 :603–619, 2002.

[DCB+89] B.A. Draper, R.T. Collins, J. Brolio, A.R. Hanson, and E.M. Riseman. The Schema system. *International Journal of Computer Vision*, 2(3) :209–250, 1989.

[DDLP07] J. Delon, A. Desolneux, J.L. Lisani, and A.B. Petro. A non parametric approach for histogram segmentation. *IEEE Transactions on Image Processing*, 16(1) :253–261, 2007.

[DeC02] D. DeCarlo. Towards real-time cue integration by using partial results. *Lecture Notes in Computer Science*, 2353 :327–342, 2002.

[Des00] A. Desolneux. *Evènements significatifs et applications à l'analyse d'image*. PhD thesis, ENS-Cachan, 2000.

[DHS01] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley New York, 2001.

[DM02] B. Ducourthial and A. Mérigot. Parallel asynchronous computation for image analysis. *Proceedings of the IEEE*, 90(7) :1218–1229, July 2002.

[DMM00a] A. Desolneux, L. Moisan, and J-M. Morel. Meaningful Alignments. *International Journal of Computer Vision*, 40(1) :7–23, 2000.

[DMM00b] A.M. Desolneux, L.M. Moisan, and J.-M. Morel. Meaningful Alignments. *International Journal of Computer Vision*, 40(1) :7–23, 2000.

[DMM01] A. Desolneux, L. Moisan, and J.-M. Morel. Edge detection by Helmholtz principle. *Journal of Mathematical Imaging and Vision*, 14(3) :271–284, 2001.

[DMM03] A. Desolneux, L. Moisan, and J.-M. Morel. A grouping principle and four applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(4) :508–513, 2003.

[DMM08] A. Desolneux, L. Moisan, and J-M. Morel. *From Gestalt Theory to Image Analysis : A Probabilistic Approach*. Springer, Interdisciplinary Applied Mathematics Series, Vol. 34, 2008.

[DN95] J. Denzler and H. Niemann. Evaluating the performance of active contour models for real-time object tracking. In *Proceedings of the Asian Conference on Computer Vision*, volume 2, pages 341–345, 1995.

[DRMFT04] A. Delorme, G. A. Rousselet, M. J. Mace, and M. Fabre-Thorpe. Interaction of top-down and bottom-up processing in the fast visual analysis of natural scenes. Technical report, Cognitive Brain Research 19, 2004.

[DRMS07] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM : Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6) :1052–1067, 2007.

[Duc01] E. Duchesnay. *Agents situés dans l'image et organisés en pyramide irrégulière*. PhD thesis, Université de Rennes, 2001.

[EKJ07] S. Ekvall, D. Kragic, and P. Jensfelt. Object detection and mapping for service robot tasks. *Robotica*, 25(02) :175–187, 2007.

[Elo05] A. Elouardi. *Évaluation des rétines électroniques pour une définition architecturale d'un système monopuce (SoC) dédié à la vision embarquée*. PhD thesis, Université Paris-Sud, UFR Scientifique d'Orsay, 2005.

[ERS99] P. Embrechts, S.I. Resnick, and G. Samorodnitsky. Extreme value theory as a risk management tool. *North American Actuarial Journal*, 3 :30–41, 1999.

[ESM97] P. Embrechts, H. Schmidli, and T. Mikosch. *Modelling of extremal events in insurance and finance*. Springer Verlag, 1997.

[Ete92] A. Etemadi. Robust segmentation of edge data. In *Proceedings of the International Conference on Image Processing and its Applications*, pages 311–314, 1992.

[FA91] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9) :891–906, 1991.

[FB81] M.A. Fischler and R.C. Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395, 1981.

[FDM+92] O.D. Faugeras, R. Deriche, H. Mathieu, N. Ayache, and G. Randall. The depth and motion analysis machine. *World Scientific Series In Machine Perception And Artificial Intelligence*, pages 143–175, 1992.

[FFIKP07] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we perceive in a glance of a real-world scene. *Journal of Vision*, 7(1) :10, 2007.

[FFJS08] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1) :36–51, 2008.

[FG97] S. Franklin and A. Graesser. Is it an agent, or just a program ? A taxonomy for autonomous agents. *Lecture Notes in Computer Science*, 1193 :21–36, 1997.

[FH04] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2) :167–181, 2004.

[Fil07] D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2007.

[FL07] P.E. Forssen and D.G. Lowe. Shape descriptors for maximally stable extremal regions. In *Proceedings of the 11th International Conference on Computer Vision*, pages 1–8, 2007.

[FSP06] G. Fritz, C. Seifert, and L. Paletta. A mobile vision system for urban detection with informative local descriptors. In *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*, page 30, 2006.

[FTVG05] V. Ferrari, T. Tuytelaars, and L. Van Gool. *Towards Category-Level Object Recognition*, chapter Simultaneous object recognition and segmentation by image exploration. Springer, 2005.

[GB04] V. Gies and T.M. Bernard. Statistical solution to watershed over-segmentation. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 1863–1866, 2004.

[GBM06] V. Gies, T.M. Bernard, and A. Merigot. Asynchronous regional computation capabilities for digital retinas. In *Proceedings of the International Workshop on Computer Architecture for Machine Perception and Sensing (CAMP)*, pages 7–11, 2006.

[GBS05] J.M. Geusebroek, G.J. Burghouts, and A.W.M. Smeulders. The Amsterdam library of object images. *International Journal of Computer Vision*, 61(1) :103–112, 2005.

[Gie05] V. Gies. *Asynchronisme dans les rétines artificielles*. PhD thesis, Université Paris-Sud, UFR Scientifique d'Orsay, 2005.

[GIM99] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529, 1999.

[GJ08] R. Grompone and J. Jakubowicz. On computational gestalt detection thresholds. *Journal of Physiology*, To appear, 2008.

[GJMR08] R. Grompone, J. Jakubowicz, J-M. Morel, and G. Randall. LSD : A line segment detector. *CMLA preprint*, (2008-15), January 2008.

[GM06] B. Grosjean and L. Moisan. A-contrario detectability of spots in textured backgrounds. *MAP5 preprint*, (2006-12), 2006.

[GS97] C.M. Grinstead and J.L. Snell. *Introduction to Probability*. American Mathematical Society, 1997.

[HL97] E. Horvitz and J. Lengyel. Perception, attention, and resources : A decision-theoretic approach to graphics rendering. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*, pages 238–249, 1997.

[HLO99] D.P. Huttenlocher, R.H. Lilien, and C.F. Olson. View-based recognition using an eigenspace approximation to the hausdorff measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 951–955, 1999.

[HR78] A.R. Hanson and E.M. Riseman. VISIONS : A computer system for interpreting scenes. *Computer Vision Systems*, pages 303–333, 1978.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15 :50, 1988.

[HS93] R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision Volume 2*. Addison-Wesley, 1993.

[IK88] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44(1) :87–116, 1988.

[IK01] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Reviews Neuroscience*, 2(3) :194–204, 2001.

[IKN+98] L. Itti, C. Koch, E. Niebur, et al. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11) :1254–1259, 1998.

[IPG+07] L. Igual, J. Preciozzi, L. Garrido, A. Almansa, V. Caselles, and B. Rouge. Automatic Low Baseline Stereo In Urban Areas. *Inverse Problems and Imaging*, 1(2) :319–348, 2007.

[Jak07] J. Jakubowicz. *La recherche d'alignements dans les images digitales et ses applications à l'imagerie satellitaire*. PhD thesis, ENS-Cachan, 2007.

[Jol03] J. Jolion. Stochastic pyramid revisited. *Pattern Recognition*, 24(8) :1035–1042, 2003.

[JZ88] C.X. Ji and Z.P. Zhang. Stereo match based on linear feature. In *Proceedings of the 9th International Conference on Pattern Recognition*, pages 875–878, 1988.

[KFM06] W.W. Kywe, D. Fujiwara, and K. Murakami. Scheduling of image processing using anytime algorithm for real-time system. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 3, pages 1095–1098, 2006.

[KH08] J.S. Kim and K.S. Hong. A new graph cut-based multiple active contour algorithm without initial contours and seed points. *Machine Vision and Applications*, 19(3) :181–193, 2008.

[KKI04] T. Komuro, S. Kagami, and M. Ishikawa. A dynamically reconfigurable SIMD processor for a vision chip. *IEEE Journal of Solid State Circuits*, 39(1) :265–268, 2004.

[KMY06] I. Kokkinos, P. Maragos, and A. Yuille. Bottom-up & top-down object detection using primal sketch features and graphical models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1893–1900, 2006.

[KS04] Y. Ke and R. Sukthankar. PCA-SIFT : A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 506–513, 2004.

[KZB04] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. *Lecture Notes in Computer Science*, 3021 :228–241, 2004.

[LBHZ08] C.H. Lampert, M.B. Blaschko, T. Hofmann, and S. Zurich. Beyond sliding windows : Object localization by efficient subwindow search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[LD06] A. Lopich and P. Dudek. Architecture of a VLSI cellular processor array for synchronous/asynchronous image processing. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 3618–3621, 2006.

[Lin94] T. Lindeberg. Scale-space theory : a basic tool for analyzing structures at different scales. *Journal of Applied Statistics*, 21(1) :225–270, 1994.

[LLS08] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3) :259–289, 2008.

[LMRL98] T. S. Lee, D. Mumford, R. Romero, and V. A. Lamme. The role of the primary visual cortex in higher level vision. *Vision Research*, 38 :2429–2454, 1998.

[LNOM08] E. Lindholm, J. Nickolls, S. Oberman, and J. Montrym. NVIDIA Tesla : a unified graphics and computing architecture. *IEEE Micro*, pages 39–55, 2008.

[LO07] H. Ling and K. Okada. An efficient Earth Mover's Distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5) :840–853, 2007.

[Low99] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[Low01] D.G. Lowe. Local feature view clustering for 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 682–688, 2001.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[LT99] J. Liu and Y.Y. Tang. Adaptive image segmentation with distributed behavior-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7) :544–551, 1999.

[LW88] Y. Lamdan and H.J. Wolfson. Geometric hashing : A general and efficient model-based recognition scheme. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 238–249, 1988.

[LW06] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. *Lecture Notes in Computer Science*, 3954 :581–594, 2006.

[Man00] A. Manzanera. *Vision Artificielle Rétinienne*. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2000.

[Mar82] D. Marr. *Vision*. Freeman, W.T., 1982.

[MBPL02] A. Manzanera, T. M. Bernard, F. Prêteux, and B. Longuet. nD skeletons : a unified mathematical framework. *Electronic Imaging*, 11(1) :25–37, 2002.

[MCUP04] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10) :761–767, 2004.

[MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th International Conference on Computer Vision*, volume 2, pages 416–423, 2001.

[MH85] T. Matsuyama and V. Hwang. SIGMA : A framework for image understanding : integration of bottom-up and top-down analyses. In *Proceedings of the 9th International Joint Conference on Artifical Intelligence*, pages 908–915, 1985.

[MM01] S. Minut and S. Mahadevan. A reinforcement learning model of selective visual attention. In *Proceedings of the fifth international conference on Autonomous agents*, pages 457–464, 2001.

[MMP04] P. Moreels, M. Maire, and P. Perona. Recognition by probabilistic hypothesis construction. In *Proceedings of the European Conference on Computer Vision*, pages 55–58. Springer, 2004.

[Moi00] A. Moini. *Vision Chips*. Kluwer Academic Publishers, 2000.

[MP07] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. *International Journal of Computer Vision*, 73(3) :263–284, 2007.

[MS89] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications in Pure and Applied Mathematics*, 42(5) :577–685, 1989.

[MS04] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3) :201–218, 2004.

[MS05] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, 2005.

[MSC+06] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J-M. Morel. An a contrario decision method for shape element recognition. *International Journal of Computer Vision*, 69(3) :295–315, 2006.

[MTEF05] K. Murphy, A. Torralba, D. Eaton, and W. Freeman. *Towards Category-Level Object Recognition*, chapter Object Detection and Localization Using Local and Global Features. Springer, 2005.

[NDBG97] Y. Ni, F. Devos, M. Boujrad, and J. H. Guan. Histogram-equalization-based adaptive image sensor for real-time vision. *IEEE Journal of Solid State Circuits*, 32(7) :1027–1036, 1997.

[NI02] V. Navalpakkam and L. Itti. A goal oriented attention guidance model. *Lecture Notes in Computer Science*, 2525 :453–461, 2002.

[NMB06] P. Nadrag, A. Manzanera, and N. Burrus. Smart retina as a contour-based visual interface. In *Proceedings of the Distributed Smart Cameras Workshop (DSC'06)*, 2006.

[NN04] R. Nock and F. Nielsen. Statistical region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11) :1452–1458, 2004.

[Nwa96] H.S. Nwana. Software agents : An overview. *Knowledge Engineering Review*, 11(3) :205–244, 1996.

[Pai01] F. Paillet. *Intégration et évaluation de Rétines Artificielles Numériques Programmables de hautes performances*. PhD thesis, Université Pierre et Marie Curie, 2001.

[PBD06] Y. Pan, J.D. Birdwell, and S.M. Djouadi. Bottom-up hierarchical image segmentation using region competition and the mumford-shah functional. In *Proceedings of the 18th International Conference on Pattern Recognition*, volume 2, pages 117–121, 2006.

[PFS05] L. Paletta, G. Fritz, and C. Seifert. Q-learning of sequential attention for visual object recognition from informative local descriptors. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 649–656, 2005.

[PL06] D.H. Parks and M.D. Levine. The mcgill object detection suite. In *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*, page 47, 2006.

[PMB99] F. Paillet, D. Mercier, and T. M. Bernard. Second generation programmable artificial retina. In *Proceedings of the IEEE ASIC/SOC Conference*, pages 304–309, 1999.

[PNSK05] T. Pang-Ning, M. Steinbach, and V. Kumar. Introduction to data mining, 2005.

[Pra01] W. K. Pratt. Digital image processing, 2001.

[PRSM95] T. Pun, C. Rauber, S. Startchik, and R. Milanese. Transforming an image into dataflows of relevant primitives for objects location, reconstruction and indexing. *Vision Interface*, 95 :15–19, 1995.

[PW08] O. Pele and M. Werman. Robust real time pattern matching using bayesian sequential hypothesis testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8) :1427–1443, 2008.

[RBG+04] V. Rodin, A. Benzinou, A. Guillaud, P. Ballet, F. Harrouet, J. Tisseau, and J. Le Bihan. An immune oriented multi-agent system for biological image processing. *Pattern Recognition*, 37(4) :631–645, 2004.

[RGD07] J. Rabin, Y. Gousseau, and J. Delon. A contrario matching of local descriptors. *Archives ouvertes HAL*, (00168285), 2007.

[Ric06] J. Richefeu. *Motion detection and analysis in digital retina-based vision systems*. PhD thesis, Université Paris VI, 2006.

[RM03] J. Richefeu and A. Manzanera. A morphological dominant points detection and its cellular implementation. In *Proceedings of ISSPA 2003*, volume 2, pages 181–184, 2003.

[RM07] T. Ridène and A. Manzanera. Mécanismes d'attention visuelle sur rétine programmable. In *Traitement et Analyse de l'Information : Méthodes et Applications (TAIMA'07)*, pages 301–306, 2007.

[RMlHM07] A. Robin, L. Moisan, and S. le Hegarat-Mascle. An unsupervised approach for subpixelic land-cover change detection. In *Proceedings of the International Workshop on the Analysis of Multi-temporal Remote Sensing Images*, pages 1–6, 2007.

[RP98] V. Rehrmann and L. Priese. Fast and robust segmentation of natural color scenes. In *Proceedings of the 3rd Asian Conference on Computer Vision*, pages 598–606, 1998.

[San95] F. Sandakly. *MESSIE-II : Contribution à la mise en œuvre d'une architecture à base de connaissances pour l'interprétation de scènes 2D et 3D*. PhD thesis, Université de Nice-Sophia Antipolis, 1995.

[Sap90] G. Saporta. *Probabilités, analyse des données et statistique*. Editions Technip, 1990.

[SB99] K. Sobottka and H. Bunke. Any-time behavior for obstacle tracking. In *Proceedings of the International Conference on Intelligent Transportation Systems*, pages 368–373, 1999.

[SB02] K. Sobottka and H. Bunke. Investigating anytime algorithms for future distance warning systems. *Real-Time Imaging*, 8(1) :61–71, 2002.

[SCS+08] L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerman, R. Cavin, et al. Larrabee : a many-core x86 architecture for visual computing. In *Proceedings of SIGGRAPH*, volume 27, 2008.

[Ser83] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, 1983.

[SH05] A. Stein and M. Hebert. Incorporating background invariance into feature-based object recognition. In *Proceedings of the 7th IEEE Workshop on Applications of Computer Vision*, 2005.

[Sha08] S.K. Shah. Performance modeling and algorithm characterization for robust image segmentation. *International Journal of Computer Vision*, To appear, 2008.

[SM00] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, 2000.

[TCKW+95] J.K. Tsotsos, S.M. Culhane, W.Y. Kei Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78(1-2) :507–545, 1995.

[TCYZ05] Z. Tu, X. Chen, A. L. Yuille, and S. Zhu. Image parsing : Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2) :113–140, 2005.

[TK95] C.J. Taylor and D.J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11) :1021–1032, 1995.

[TL06] T. Tamminen and J. Lampinen. Sequential monte carlo for bayesian matching of objects with occlusions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6) :930–941, 2006.

[TMF04] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features : efficient boosting procedures for multiclass object detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, pages 762–769, 2004.

[TMFR03] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. pages 273–280, 2003.

[Tor03] A. Torralba. Contextual priming for object detection. *International Journal of Computer Vision*, 53(2) :169–191, 2003.

[TZ02] Z. Tu and S.C. Zhu. Image segmentation by data-driven markov chain monte carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5) :657–673, 2002.

[Ull07] S. Ullman. Object recognition and segmentation by a fragment-based hierarchy. *Trends in Cognitive Sciences*, 11(2) :58–64, 2007.

[VCB06] T. Veit, F. Cao, and P. Bouthemy. An a contrario decision framework for region-based motion detection. *International Journal of Computer Vision*, 68(2) :163–178, 2006.

[VJ02] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2) :137–154, 2002.

[VT02] R.C. Veltkamp and M. Tanase. Content-based image retrieval systems : A survey. *Technical Report*, 2002.

[WDSS08] C. Wojek, D. Dorkó, A. Schulz, and B. Schiele. Sliding-windows for rapid object class localization : A parallel technique. In *Proceedings of the 30th DAGM Symposium*, 2008.

[WHMM06] L. Wolf, X. Huang, I. Martin, and D. Metaxas. Patch-based texture edges and segmentation. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 481–493. Springer, 2006.

[WJ95] M. Wooldridge and N.R. Jennings. Intelligent agent : theory and practice. *Knowledge Engineering Review*, 10 :115–152, 1995.

[WKKI05] Y. Watanabe, T. Komuro, S. Kagami, and M. Ishikawa. Parallel extraction architecture for image moments of numerous objects. In *Proceedings of the 7th International Workshop on Computer Architecture for Machine Perception*, pages 105–110, 2005.

[WR97] H.J. Wolfson and I. Rigoutsos. Geometric hashing : An overview. *IEEE Computational Science & Engineering*, pages 10–21, 1997.

[WWWC04] R. Wojciechowski, K. Walczak, M. White, and W. Cellary. Building virtual and augmented reality museum exhibitions. In *Proceedings of the 9th international conference on 3D Web technology*, pages 135–144, 2004.

[XAB07] N. Xu, N. Ahuja, and R. Bansal. Object segmentation using graph cuts based active contours. *Computer Vision and Image Understanding*, 107(3) :210–224, 2007.

[XM08] X. Xie and M. Mirmehdi. MAC : Magnetostatic Active Contour Model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4) :632–646, 2008.

[YWKI07] K. Yamaguchi, Y. Watanabe, T. Komuro, and M. Ishikawa. Design of a Massively Parallel Vision Processor based on Multi-SIMD Architecture. *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 3498–3501, 2007.

[ZCZX08] Q. Zhang, Y. Chen, Y. Zhang, and Y. Xu. SIFT implementation and optimization for multi-core systems. *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, pages 1–8, 2008.

[ZD05] L. Zhao and LS Davis. Closely coupled object detection and segmentation. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, volume 1, pages 454–461, 2005.

[Zil96] S. Zilberstein. Using Anytime Algorithms in Intelligent Systems. *AI Magazine*, 17(3) :73, 1996.

[ZMLS07] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories : A comprehensive study. *International Journal of Computer Vision*, 73(2) :213–238, 2007.

[ZPG+04] Q. Zhou, D. Parrott, M. Gillen, D. M. Chelberg, and L. Welch. Agent-based computer vision in a dynamic, real-time environment. *Pattern Recognition*, 37(4) :691–705, 2004.

[ZTY07] S. Zheng, Z. Tu, and A.L. Yuille. Detecting object boundaries using low-, mid-, and high-level information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[ZY96] S.C. Zhu and A. Yuille. Region competition : Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9) :884–900, 1996.

[ZZS07] W. Zhang, G. Zelinsky, and D. Samaras. Real-time accurate object detection using multiple resolutions. In *Proceedings of the 11th IEEE International Conference on Computer Vision*, pages 1–8, 2007.

Publications de l'auteur

- Image segmentation by a contrario simulation
  Nicolas Burrus and Thierry M. Bernard and Jean-Michel Jolion
  Journal of Pattern Recognition, À paraître

- Bottom-up and top-down object matching using asynchronous agents and a contrario principles
  Nicolas Burrus and Thierry M. Bernard and Jean-Michel Jolion
  6th International Conference on Computer Vision Systems (ICVS 2008), Mai 2008

- Segmentation d'image par simulations a contrario
  Nicolas Burrus, Thierry M. Bernard et Jean-Michel Jolion
  16e congrès francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle, Janvier 2008

- Smart retina as a contour-based visual interface
  Paul Nadrag and Antoine Manzanera and Nicolas Burrus
  Distributed Smart Cameras Workshop (DSC'06), Octobre 2006

- Adaptive Vision Leveraging Digital Retinas : Extracting Meaningful Segments
  Nicolas Burrus and Thierry M. Bernard
  Advanced Concepts for Intelligent Vision Systems International Conference (ACIVS'06), Septembre 2006