# Segmentation d'objets mobiles par fusion RGB-D et invariance colorimétrique

Julian Murgia

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

# Segmentation d'objets mobiles par fusion RGB-D et invariance colorimétrique

JULIAN MURGIA

IRTES sciences & ingénierie

Institut de Recherche sur les Transports, l'Énergie et la Société

SPIM

# Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

N° | X | X | X |

THÈSE présentée par

JULIAN MURGIA

pour obtenir le

Grade de Docteur de
l'Université de Technologie de Belfort-Montbéliard

Spécialité : **Informatique**

# Segmentation d'objets mobiles par fusion RGB-D et invariance colorimétrique

Unités de Recherche :
Institut de Recherche sur les Transports, l'Énergie et la Société (IRTES),
Laboratoire Systèmes et Transports (IRTES-SeT)

Soutenue publiquement le XX/XX/XX devant le Jury composé de :

| | | |
|---|---|---|
| DENIS HAMAD | Rapporteur | Professeur des Universités à l'ULCO |
| LOUAHDI KHOUDOUR | Rapporteur | Directeur de Recherche au CEREMA |
| ABDELMALIK TALEB-AHMED | Examinateur | Professeur des Universités à l'UVHC |
| CINDY CAPPELLE | Examinatrice | Maître de Conférences à l'UTBM |
| YASSINE RUICHEK | Directeur de thèse | Professeur des Universités à l'UTBM |
| CYRIL MEURIE | Encadrant | Chargé de Recherche à l'IFSTTAR |

# Remerciements

Les travaux de recherche présentés dans ce manuscrit ont été réalisés au sein du laboratoire Systèmes et Transports (SeT) de l'Institut de Recherche sur les Transports, l'Énergie et la Société (IRTES) rattaché à l'Université de Technologie de Belfort-Montbéliard (UTBM). Je tiens pour cela à remercier le directeur du SeT, M. Yassine RUICHEK, pour m'avoir accueilli au sein de ce laboratoire.

Je tiens à exprimer ma gratitude à M. Denis HAMAD, Professeur à l'Université de Littoral Côte d'Opale (ULCO), ainsi qu'à M. Louahdi KHOUDOUR, Directeur de Recherche au Centre d'études et d'Expertise sur les Risques, l'Environnement, la Mobilité et l'Aménagement (CEREMA), d'avoir accepté d'être les rapporteurs de ma thèse.

Je remercie également M. Abdelmalik TALEB-AHMED, Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis, d'avoir accepté de présider mon jury de thèse. Je souhaite également adresser un remerciement spécial à Mme Cindy CAPPELLE, Maître de Conférences à l'UTBM/IRTES-SeT, pour avoir accepté d'examiner cette thèse et pour les nombreux moments partagés avec elle ainsi que toute l'équipe Perception de l'Environnement et Navigation Autonome (PENA).

Je tiens à remercier sincèrement M. Yassine RUICHEK Professeur à l'Université de Technologie de Belfort-Montbéliard (France) et Directeur du laboratoire IRTES-SeT, pour avoir dirigé mon doctorat et guidé mes recherches.

J'adresse également mes remerciements les plus chaleureux à M. Cyril MEURIE, Chargé de Recherche à l'Institut Français des Sciences et Technologies des Transports, de l'Aménagement et des Réseaux (IFSTTAR) pour m'avoir initialement proposé cette thèse et pour m'avoir accordé sa confiance. Je lui exprime toute ma gratitude pour les nombreux conseils qu'il m'a octroyés, pour ses encouragements et pour son encadrement.

Je tiens par ailleurs à remercier tous mes collègues et amis doctorants et docteurs, dont l'ambiance a contribué fortement à ce travail : Frédérick ZANN, Baudouin DAFFLON, Jean-Michel CONTET, Franck GECHTER, Julien MOREAU, Dhouha ATTIA, Houssam SALMANE, Ali MOHAMMADI, Saeid MOOSAVI, Pablo GRUER, Yongliang QIAO, Youssef EL-MERABET, Yong FANG, You LI et Lijun WEI.

J'exprime également ma gratitude à M. Fadi DORNAIKA, Professeur à l'Université du Pays-Basque (Espagne) dont les conseils ont été précieux.

Enfin, j'adresse mes remerciements les plus affectueux à Emiline, dont la patience et la compréhension m'ont aidé à surmonter les difficultés autant que les montagnes qu'elle me fait et continuera j'espère de me faire grimper. Je remercie également mes parents et toute ma famille pour leur précieux soutien, ainsi que mes amis Mélanie, François-Xavier, Florine et Alexandre.

# Introduction

La vidéo-surveillance peut être définie par un "système mono ou multi-caméras doté d'un dispositif de transmission d'images, disposé dans un espace public ou privé pour le surveiller à distance". Généralement, de tels systèmes sont utilisés à des fins de sécurité et de prévention.

La démocratisation et les coûts de plus en plus bas de ces systèmes fait qu'ils sont de plus en plus utilisés dans le cadre privé et notamment pour la sécurisation des habitations. Dans ce cadre, des algorithmes intelligents de détection de mouvement viennent par exemple enrichir ledit système afin qu'il soit le plus autonome possible. Cependant, dans les environnements publics, cette surveillance est de par sa complexité habituellement effectuée par un ou plusieurs opérateurs humains, observant plusieurs moniteurs à la fois.

La monotonie de cette tâche et la fatigue visuelle qu'elle entraîne ne permet généralement pas à cet opérateur de garder l'attention nécessaire à la détection d'événements potentiellement dangereux pendant toute une journée. À cela s'ajoute le fait qu'il est difficile de focaliser son attention sur plusieurs scènes en même temps. C'est pourquoi il est nécessaire de développer des outils d'aide à la surveillance et à la décision comme la détection d'objets mobiles (véhicules, personnes, etc.) voire de reconnaître des mouvements et des situations potentiellement dangereuses afin de focaliser l'attention de l'opérateur qui peut alors interagir plus rapidement.

Dans le cadre des systèmes de transports intelligents, les systèmes d'aide à la conduite sont des outils destinés à assister le conducteur dans sa tâche de conduite, soit à des fins de sécurité par l'évitement de situations pouvant conduire à des accidents, soit à des fins d'amélioration du confort de conduite de l'usager. Ces systèmes sont de plus en plus automatisés et intégrés dans le véhicule. Nous pouvons par exemple citer dans le domaine de l'amélioration de la sécurité :

— l'ABS (Anti-Blocking System), l'ESP (Electronic Stability Program), le DST (Dynamic Steering Torque), le limitateur de vitesse, la détection de piétons, le système anti-collision, voire des systèmes à l'étude comme la détection de fatigue, etc.

— (des)activation automatique des feux et des essuie-glace, le régulateur de vitesse, la navigation par GPS, l'aide au stationnement, la détection de panneaux de signalisation, etc.

Cette liste n'est bien sûr pas exhaustive et l'avancement des technologies en la matière permet de voir naître de nouvelles fonctions régulièrement. Les systèmes d'aide à la conduite employant des techniques de vision assistée par ordinateur sont presque exclusivement actifs en milieu extérieur, où les conditions sont plus complexes à maîtriser. Par exemple, un nuage obstruant le cheminement des rayons lumineux du soleil vers la scène peut causer non-seulement des changements d'illumination sur l'environnement, mais aussi des changements de couleurs. La détection d'objets mobiles

est alors sujette à l'apparition de résultats indiquant la présence d'objets dans le champ de vision de la caméra, trompée par la présence de sur-illuminations momentanées, d'ombrages, de bruits ou de couleurs modifiées. Ces détections erronées peuvent conduire à des prises de décisions mettant en jeu la crédibilité des instances qui les utilisent, voire la sécurité des usagers dans les cas les plus sensibles. Il est donc primordial d'assurer la robustesse des systèmes d'aide à la vision en améliorant cette perception, ce que nous tentons de traiter dans cette thèse en parant par exemple à l'apparition de faux positifs dans les détections d'objets mobiles, aussi bien en milieu intérieur qu'extérieur, et en conditions difficiles.

Les travaux réalisés durant cette thèse se sont déroulés dans ce contexte global de vision assistée par ordinateur, et plus particulièrement dans le cadre de problématiques de vidéo-surveillance opérables en tous lieux (intérieur et extérieur) et à tout moment de la journée. Néanmoins, nous nous sommes limités à l'utilisation d'une seule caméra fixe qui, nous le verrons par la suite, pourra néanmoins fournir en plus de l'information de couleur, une information sur la distance des objets présents dans la scène.

Nous souhaitons dans cette thèse, surmonter un certain nombre de difficultés bien connues de la littérature dans ce domaine, à savoir :

1. les changements d'illumination (globale ou locale),
2. la similarité des couleurs de la scène et des objets mobiles,
3. la nature dynamique de certaines régions comme c'est le cas ici pour les feuillages d'arbres par exemple,

Pour ce faire, nous apportons notre contribution dans le domaine de la détection d'objets mobiles en tous lieux et à tout moment de la journée en proposant une méthode de fusion couleur et profondeur intégrée à l'algorithme de segmentation fond/forme Codebook, couplée à l'utilisation d'invariant colorimétrique adapté.

Dans le chapitre 1, nous présenterons le concept d'invariance colorimétrique et nous introduisons un certain nombre d'invariants colorimétriques et d'espaces de représentation couleur. Nous poursuivons avec un état de l'art des méthodes de segmentation d'objets mobiles et décrivons de façon plus détaillée l'algorithme que nous avons choisi pour la suite de nos travaux : le Codebook.

Le deuxième chapitre de ce mémoire présente la stratégie initiale, basée sur les deux concepts cités précédemment : l'invariance colorimétrique et la détection des objets mobiles par méthode de détection fond/forme. Nous y décrivons par ailleurs notre protocole de test, à savoir les bases de données vidéo utilisées et la méthode d'évaluation des résultats. Ensuite, nous présentons les premiers résultats obtenus par la méthode proposée : le Codebook couplé à l'utilisation d'invariants colorimétriques. Ces résultats serviront de base de comparaison pour les tests décrits dans le chapitre suivant.

Dans le troisième chapitre, nous proposons d'intégrer une information supplémentaire dans la méthode de détection d'objets mobiles pour classer les pixels de l'image, à savoir la profondeur. Afin d'augmenter la robustesse de notre système nous commençons d'abord par décrire la méthode retenue pour le calcul de cette information de profondeur, dépendant du capteur employé, puisque

nous le rappelons, nous visons à proposer une méthode fonctionnant à la fois en intérieur et en extérieur. Nous proposons ensuite une stratégie de fusion booléenne des informations couleur et profondeur au sein du même algorithme : le Codebook RGB-D couplé là encore à l'utilisation d'invariant colorimétrique adapté. Enfin, nous terminons ce chapitre en proposant une fusion par logique floue de ces deux informations permettant d'améliorer dans certaines conditions nos résultats globaux.

Pour finir, nous dresserons une conclusion générale de nos travaux et exposerons des perspectives à court et moyen termes pour poursuivre dans l'amélioration de nos résultats.

# Table des matières

# Table des figures

# Liste des tableaux

# Chapitre 1

# Invariance colorimétrique et segmentation fond/forme



Le présent chapitre introduit les différents concepts mis en oeuvre dans nos travaux : le principe de l'invariance colorimétrique est d'abord décrit en détails avec une présentation d'un certain nombre de méthodes et d'espaces de représentation couleur possédant des propriétés invariantes ; ensuite, nous présenterons les objectifs et les problématiques rencontrées habituellement dans le cadre de la détection d'objets mobiles. Un état de l'art des méthodes de segmentation fond/forme donnera un aperçu des différentes méthodes de la littérature et pour les plus importantes, de leurs approches respectives pour résoudre ces problématiques. L'accent sera mis sur la méthode dite "Codebook" qui a été utilisée dans ces travaux.

# 1 Les invariants colorimétriques

Dans le cadre d'applications de vidéo-surveillance se voulant opérables de jour comme de nuit, en intérieur ou extérieur, etc, nous sommes confrontés à un certain nombre de difficultés telles que des changements de luminosité, des ombres et des changements de couleurs qui perturbent la détection fine et le suivi des objets mobiles. De ce fait, nombre d'auteurs utilisent des approches basées sur des invariants colorimétriques pour atténuer ces perturbations comme par exemple les ombres portées [24, 25, 23, 44, 70, 62] ou plus généralement lisser les images afin d'en produire de nouvelles images dont la scène ne serait théoriquement pas impactée par des changements de couleurs ou de luminosité, et aussi d'aider à la reconnaissance d'objets [78, 16, 30, 35, 34, 36, 58, 37].

Dans cette thèse, nous avons suivi la même ligne directrice, et employé largement le concept d'invariance colorimétrique afin de limiter les effets indésirables observés dans la segmentation des objets mobiles au cours de journées complètes. Cette section introduit la problématique à l'origine de laquelle plusieurs opérateurs d'invariance colorimétrique ont été développés dans la littérature, ainsi qu'un certain nombre d'espaces couleur possédant des propriétés invariantes.

## 1.1 Problématique

Il est généralement admis qu'une scène change visuellement au fil d'une journée, que ce soit en termes de couleurs, d'illumination, etc. : les conditions météorologiques changeantes, les occultations du soleil par des nuages, la saison de l'année, sont autant de facteurs qui provoquent d'importantes variations d'une même scène. Mais d'autres facteurs peuvent également modifier la couleur perçue, outre les changements de l'intensité de la source lumineuse. Nous pouvons aussi citer :

— La direction d'observation ;

— Les changements de la géométrie des objets ;

— La direction de la source lumineuse ;

— Les changements dans la distribution spectrale de l'énergie lumineuse.

En effet, un objet de couleur verte garde la même couleur verte, qu'il soit photographié à midi en été ou le soir sous un soleil rougeoyant. Pourtant dans ces deux cas, la couleur perçue ne sera pas le même vert : si à midi on l'observera d'un vert brillant, l'objet apparaîtra le soir d'un vert beaucoup plus nuancé. Néanmoins notre cerveau, et plus précisément notre système visuel, est capable de s'adapter à ce type de changements, nous permettant ainsi de reconnaître un objet quelle que soit la perception que nous en avons. Ce concept s'appelle l'invariance colorimétrique. Il s'agit de considérer que la couleur d'un objet est dissociable de la couleur de la lumière qui l'illumine.

En vision par ordinateur, la prise en compte de cette problématique est une nécessité. En effet, de nombreux travaux nécessitent une reconnaissance robuste de certains éléments de l'image qui peuvent être fortement impactés par leur aspect colorimétrique. A titre d'exemple, les applications d'extraction et de reconnaissance de panneaux de signalisation routiers basées sur le facteur couleur [56] rencontrent des échecs lorsque le temps est pluvieux ou brumeux. La nécessité de pré-traiter les images avant l'extraction des éléments qu'elles contiennent est réelle. Ainsi, la modification des couleurs de l'image pour la faire apparaître telle qu'elle serait, éclairée par un illuminant *canonique* (par exemple, blanc) est une solution. Sans pour autant être exhaustif, d'autres applications comme la segmentation d'images aériennes pour la caractérisation de toitures [18], la segmentation

d'images couleur de cytologie bronchique pour la détection de cellules suspectes en cytopathologie [54] ou encore la caractérisation de l'environnement de réception de signaux satellitaires [51] font appel à l'invariance colorimétrique ou aux espaces de représentation couleur pour améliorer leurs résultats.

Depuis plusieurs années, la problématique de l'invariance colorimétrique est un sujet de recherche qui prend de l'importance. De nombreuses méthodes ont vu le jour et sont généralement découpées selon les deux étapes successives suivantes :

1. L'estimation de l'illuminant de la scène ;
2. L'adaptation chromatique : la correction des couleurs de l'image pour obtenir une nouvelle image contenant la même scène éclairée par un Illuminant canonique.

L'étape d'estimation de l'illuminant consiste à déterminer les caractéristiques chromatiques de la couleur. C'est une étape clé du concept d'invariance colorimétrique, car plus l'estimation de l'illuminant est précise, plus l'image corrigée sera valide. Plusieurs approches ont été utilisées dans la littérature pour la détermination de l'illuminant d'une scène et sont généralement classées parmi les 3 catégories suivantes :

1. Approches statiques ;
2. Approches par apprentissage ;
3. Approches combinant les deux précédentes.

L'étape de correction des couleurs de l'image est commune à la plupart des invariants colorimétriques. Une fois la couleur de l'illuminant déterminée, il devient possible de transformer les images afin de leur donner un aspect chromatique tel qu'elles apparaîtraient si elles avaient été prises avec une scène éclairée par un illuminant canonique. Cette tâche est usuellement réalisée à l'aide du modèle diagonal, ou *modèle de von Kries* :

$$I^c = \Lambda^{u,c} I^u \tag{1.1}$$

où $I^u$ est l'image obtenue sous une source d'illumination inconnue, et $I^c$ est l'image transformée comme si elle était prise sous l'Illuminant canonique. $\Lambda^{u,c}$ est la matrice diagonale de mappage, soit :

$$\Lambda^{u,c} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} = \begin{pmatrix} \frac{L_R^c}{L_R^u} & 0 & 0 \\ 0 & \frac{L_G^c}{L_G^u} & 0 \\ 0 & 0 & \frac{L_B^c}{L_B^u} \end{pmatrix} \tag{1.2}$$

où $L^u$ est la source de lumière inconnue et $L^c$ est la source de lumière canonique.

Malgré la variété des méthodes d'invariance colorimétrique, il n'est pas possible de définir une approche globale. En effet, la variété des situations et des caractéristiques à prendre en compte en termes d'invariance colorimétrique sont conséquentes. L'angle de vue de la caméra, les reflets, les orientations des surfaces, les intensités lumineuse et chromatique sont des exemples d'éléments variables qu'une seule approche ne peut gérer à la fois. C'est pourquoi, plusieurs méthodes sont souvent considérées dans un même traitement : Bianco *et al.* [5] tentent par exemple de déterminer l'invariant colorimétrique le plus intéressant à l'aide d'une forêt d'arbres décisionnels. Par simplification, il est généralement considéré qu'une scène n'est illuminée que par un seul illuminant et supposé que l'illumination est uniforme sur toute l'image, ce qui n'est évidemment pas toujours le cas. Néanmoins, plusieurs travaux considèrent l'existence et l'estimation de un à deux illuminants différents au sein d'une même image [43, 16, 37, 63].

Dans ce chapitre, nous avons cherch  pr senter un certain nombre d'invariants colorim triques et d'espaces de repr sentation couleur, afin de r pondre  la probl matique de d tection et de suivi d'objets mobiles en tous lieux et  tout moment de la journ e. Pour ce faire, nous avons choisi de classer les invariants et espaces que nous utiliserons ensuite dans ce m moire en di rentes cat gories et de les d tailler ci-apr s : la figure 1.1 r capitule le classement de ces di rentes m thodes. La figure 1.3 illustre un exemple d'application d'invariants pour une image donn e.

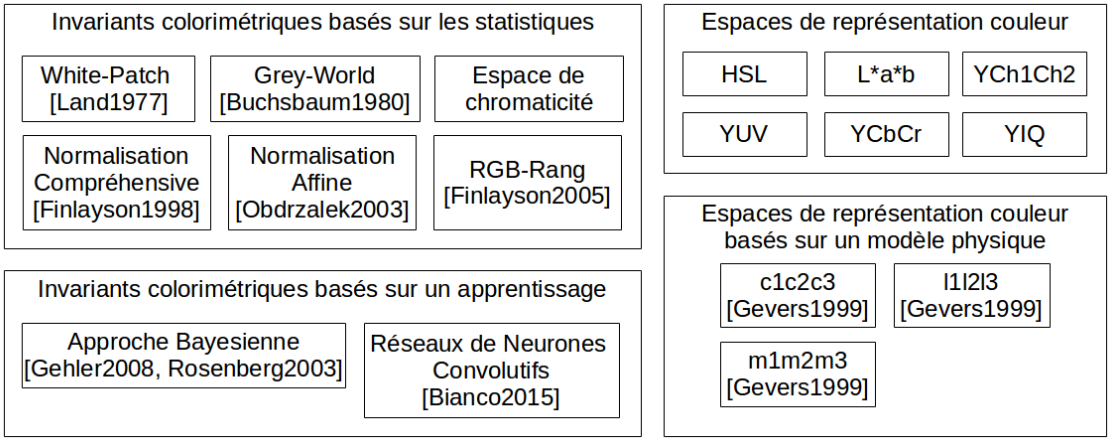



FIGURE 1.1 – Cat gories de m thodes d'invariance colorim trique et d'espaces de repr sentation couleur.

## 1.2 Invariants colorimétriques basés sur les statistiques

La d finition de la couleur per ue d pend de plusieurs facteurs. Elle est d finie comme le r sultat d'une r flexion d'un rayon lumineux sur une surface. Chacun de ces deux l ments a ses caract - ristiques propres, qui sont g n ralement inconnues. La plupart des m thodes emploient le mod le de r flexion lambertien qui d finit une image $I = (I_R, I_G, I_B)^T$ pour une surface lambertienne une position $x$ comme suit :

$$I_c = \int_w E(\lambda, x) S(\lambda, x) \rho_c(\lambda) \mathrm{d}\lambda \tag{1.3}$$

o  $c$ repr sente la composante couleur et $E(\lambda, x)$, $S(\lambda, x)$ et $\rho_c(\lambda)$ sont respectivement la distribution spectrale de l'illuminant, la r flectance de la surface et la sensitivit  de la cam ra. Si l'on consid re $\omega$ comme  tant le spectre visible de la lumi re, l' quation devient :

$$L(x) = \begin{pmatrix} L_R(x) \\ L_G(x) \\ L_B(x) \end{pmatrix} = \int_\omega E(\lambda, x) \rho(\lambda) \mathrm{d}\lambda \tag{1.4}$$

L'estimation de la chromaticit  de la source lumineuse pour une seule image est un probl me ind termin  car $E(\lambda, x)$ et $\rho(\lambda) = (\rho_R, \rho_G, \rho_B)^T$ sont tous deux inconnus. Ainsi, on simplifie le plus souvent les propri t s statistiques des illuminants et de la r flectance des surfaces. De plus, il est g n ralement admis que l'illumination est uniforme sur toute l'image (c'est- -dire que $E(\lambda)$ est constant), ce qui est en r alit  tr s improbable.

Le mod le de formation des images sur lequel se basent beaucoup de m thodes est le mod le

diagonal. Celui-ci est décrit par la transformation suivante :

$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{1.5}$$

L'équation 1.5 se trouve être à la base de beaucoup de méthodes dites statiques. Ces dernières s'appuient sur l'idée que la couleur observée $(R', G', B')^T$ d'un objet est fonction de sa couleur $(R, G, B)^T$ et de la couleur de l'illuminant $(\alpha, \beta, \gamma)^T$. L'objectif est donc de déterminer les composantes de la couleur de l'illuminant, afin de procéder à l'inverse de cette transformation et ainsi, obtenir une image obtenue avec un Illuminant canonique (de couleur blanche par exemple). Cet illuminant peut être déterminé en utilisant des valeurs statistiques calculées à partir des couleurs présentes dans l'image, comme la moyenne, le minimum, le maximum, la variance, etc.

Les approches statiques utilisent les caractéristiques bas-niveau des images. Ces approches considèrent généralement que la couleur de l'illuminant de la scène est uniforme dans toute l'image. Par exemple, l'algorithme White Patch [50] suppose que la réponse chromatique maximale dans une scène est la couleur blanche. Considérant cela, il suffit de transformer l'image en divisant chacune de ses composantes par sa valeur maximale respective :

$$R' = \frac{R}{max(R)}, G' = \frac{G}{max(G)}, B' = \frac{B}{max(B)} \tag{1.6}$$

**L'algorithme Grey-world** [8] considère que la couleur moyenne d'une image est achromatique, c'est-à-dire grise. Cela implique que toute déviation entre la couleur moyenne et le gris est causée par la source de lumière. Cette considération se base sur l'hypothèse plus générale que la réflectance moyenne de toutes les surfaces de la scène est achromatique.

$$R' = \frac{R}{mean(R)}, G' = \frac{G}{mean(G)}, B' = \frac{B}{mean(B)} \tag{1.7}$$

**L'espace de chromaticité** appelé aussi *RGB normalisé* dérive de l'espace colorimétrique RGB. Il permet de s'affranchir des effets des variations colorimétriques d'une image :

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \tag{1.8}$$

**La Normalisation Compréhensive** (Comprehensive Normalization) [27] utilise successivement les deux invariants colorimétriques précédents de façon itérative pour former une image invariante. Un algorithme itératif est utilisé pour former cette image en deux étapes : la première étape consiste à normaliser chaque pixel avec l'espace RGB normalisé, tandis que la seconde normalise chaque canal de couleur séparément pour que la somme des composantes soit égale à un tiers du nombre de pixels. Ces itérations continuent jusqu'à convergence de l'image résultante.

L'algorithme 1 décrit cette méthode pour une image $I$ de taille $N \times M$ :

**Algorithme 1** Algorithme de la Comprehensive Normalization décrite par [27]

Initialisation : $I_0 \leftarrow I$
**while** $I_{t+2} \neq I_t$ **do**
  **for** chaque pixel $p$ **do**
    $C_p^R \leftarrow \frac{R(p)}{R(p)+G(p)+B(p)}$
    $C_p^G \leftarrow \frac{G(p)}{R(p)+G(p)+B(p)}$
    $C_p^B \leftarrow \frac{B(p)}{R(p)+G(p)+B(p)}$
  **end for**
  $r_{t+1} \leftarrow \frac{3}{N \times M} \sum_p C_p^R$
  $g_{t+1} \leftarrow \frac{3}{N \times M} \sum_p C_p^G$
  $b_{t+1} \leftarrow \frac{3}{N \times M} \sum_p C_p^B$
  Puis on normalise ces valeurs :
  $r_{t+2}^p \leftarrow \frac{C_p^R}{r_{t+1}}$
  $g_{t+2}^p \leftarrow \frac{C_p^G}{g_{t+1}}$
  $b_{t+2}^p \leftarrow \frac{C_p^B}{b_{t+1}}$
  $I_{t+2} \leftarrow (r_{t+2}^p; g_{t+2}^p; b_{t+2}^p)$
**end while**

**La Normalisation Affine** (Affine Normalization) donne une invariance selon le modèle diagonal avec une translation [58]. L'équation 1.9 est effectuée pour chaque canal R, G et B d'une image $I$ :

$$f_{(p)}^k = \frac{I_{(p)}^k - mean(I_{(p)}^k)}{std(I_{(p)}^k}, k \in [R, G, B] \tag{1.9}$$

où la fonction $std(I_{(p)}^k$ est l'écart-type de $f_{(p)}^k$ calculée dans une fenêtre d'intérêt $W$ centrée sur le pixel $p$.

**RGB-Rang** Finlayson et al. [26] proposent une méthode (dénommée RGB-Rang) basée sur le classement des mesures colorimétriques des pixels de l'image, classement qui est invariant aux changements d'illumination. Ainsi, les pixels sont caractérisés par leur rang RGB afin d'obtenir une nouvelle image. Pour ce faire, la mesure du rang $M_k(n)$ du niveau de couleur $n$ pour l'image $I_k$ de composante $k \in [R, G, B]$, peut être obtenue par l'égalisation de l'histogramme mono-dimensionnel $H_k[I]$ décrite par l'équation 1.10.

$$M_k[I](u) = \frac{\sum_{u=0}^{n} H_k[I](u)}{\sum_{u=0}^{N-1} H_k[I](u)}, n = 0, ..., (N-1) \tag{1.10}$$

où $u \in [0..255]$ et $N$ indique le nombre de niveaux de quantifications des composantes colorimétriques. Généralement, on considère que $N = 256$.

## 1.3 Espaces de représentation couleur basés sur un modèle physique

Shafer [69] a proposé un *modèle de réflexion dichromatique* basé sur la physique. Il se base sur l'hypothèse que la lumière réfléchie d'une surface diélectrique peut être divisée en deux composantes :

1. La réflexion *spéculaire*, dont la couleur est celle de la source de lumière. Cette réflexion

s'observe lorsqu'un rayon lumineux frappe une surface lisse avec un certain angle. La réflexion de ce rayon s'effectue avec le même angle d'incidence que le rayon lumineux.

2. La réflexion *diffuse* ou *body*, correspond à la réflexion d'un rayon lumineux qui s'effectue dans toutes les directions.

Supposons une surface de taille infinitésimale sur un objet, et trois capteurs pour le rouge, le vert et le bleu, sensibles à certaines longueurs d'onde ($f_R(\lambda)$, $f_G(\lambda)$, $f_B(\lambda)$) pour obtenir une image de cette surface. Nous avons l'équation 1.11 :

$$C = m_b(n,s)\int_\lambda f_c(\lambda)E(\lambda)c_b(\lambda)\mathrm{d}\lambda + m_s(n,s,v)\int_\lambda f_c(\lambda)E(\lambda)c_s(\lambda)\mathrm{d}\lambda \tag{1.11}$$

avec $C \in \{R,G,B\}$ et $E(\lambda)$ la lumière incidente. $c_b(\lambda)$ et $c_s(\lambda)$ sont respectivement l'Albédo de la surface et la réflectance de Fresnel. Les termes $m_b$ et $m_s$ sont les dépendances géométriques du matériau (diffuse) et de la surface (spéculaire), $\lambda$ est la longueur d'onde. $n$ est le vecteur normal à la surface, $s$ la direction de la source de lumière et $v$ la direction de l'observateur. Nous remarquons que la composante spéculaire est dépendante de la position de l'observateur, ce qui n'est pas le cas du modèle lambertien.

L'équation précédente peut être simplifiée en considérant que la réflectance de Fresnel $c_s(\lambda)$ a une valeur constante indépendante de la longueur d'onde et que l'illumination est blanche, impliquant une densité d'énergie égale pour toutes les longueurs d'ondes du spectre visible de la lumière. nous avons donc dans ce cas $E(\lambda) = E$ et $c_s(\lambda) = c_s$, ce qui modifie l'équation 1.11 :

$$C_w = E.m_b(n,s)\int_\lambda f_c(\lambda)c_b(\lambda)\mathrm{d}\lambda + E.m_s(n,s,v).c_s\int_\lambda f_c(\lambda)\mathrm{d}\lambda \tag{1.12}$$

pour $C_w \in \{R_w, G_w, B_w\}$ donnant les réponses sensorielles rouge, verte et bleue sous une lumière blanche. Le deuxième terme de l'équation 1.12 peut également être simplifié puisque sous une lumière blanche, nous avons :

$$\int_\lambda f_R(\lambda)\mathrm{d}\lambda = \int_\lambda f_G(\lambda)\mathrm{d}\lambda = \int_\lambda f_B(\lambda)\mathrm{d}\lambda = f \tag{1.13}$$

Ce qui simplifie encore l'équation 1.12 :

$$C_w = E.m_b(n,s)\int_\lambda f_c(\lambda)E(\lambda)c_b(\lambda)\mathrm{d}\lambda + E.m_s(n,s,v).c_s.f \tag{1.14}$$

L'équation 1.15 peut être retenue pour le calcul de la couleur $I$ du pixel $p$ :

$$I(\lambda,p) = m_b c_b(\lambda) + m_s c_s(\lambda) = D(p) + m(p)L \tag{1.15}$$

Dans l'équation simplifiée, $D = (D_r, D_g, D_b)$ est la composante *diffuse* reflétée et $L = (L_r, L_g, L_b)$ est le vecteur couleur global de l'illuminant, multiplié par la fonction $m(p)$ qui dépend de la position spatiale et de la géométrie locale du point visible de la scène pour le pixel $p$. Le terme spéculaire $m(p)L$ a la même distribution spectrale que la lumière incidente. En se basant sur ce modèle de réflexion, de nouveaux espaces couleur peuvent être proposés.

**L'espace c1c2c3** proposé par Gevers et Smeulders [34] est conçu pour être plus adapté aux surfaces mates et peu brillantes, voire sombres. L'objectif de cet espace est d'être efficace même en

présence de sur-illuminations avec la contrainte d'une source de lumière blanche.

$$\begin{aligned} c1 &= arctan\left(\frac{R}{max(G,B)}\right) \\ c2 &= arctan\left(\frac{G}{max(R,B)}\right) \\ c3 &= arctan\left(\frac{B}{max(R,G)}\right) \end{aligned} \tag{1.16}$$

**L'espace l1l2l3** est également proposé par Gevers et al. [34] pour les matériaux brillants. Il est plus approprié pour les observations où la lumière est contrôlée.

$$\begin{aligned} l1 &= \frac{(R-G)^2}{M} \\ l2 &= \frac{(R-B)^2}{M} \\ l3 &= \frac{(G-B)^2}{M} \end{aligned} \tag{1.17}$$

avec $M = (R-G)^2 + (R-B)^2 + (G-B)^2$.

**L'espace m1m2m3** est aussi proposé [34] pour les cas d'observation sans contrainte sur la distribution spectrale de l'énergie lumineuse $E(\lambda)$.

$$\begin{aligned} m1 &= log\left(\frac{R \times mean(G)}{mean(R) \times G}\right) \\ m2 &= log\left(\frac{R \times mean(B)}{mean(R) \times B}\right) \\ m3 &= log\left(\frac{G \times mean(B)}{mean(G) \times B}\right) \end{aligned} \tag{1.18}$$

Ces trois espaces c1c2c3, l1l2l3 et m1m2m3 sont invariants à la direction d'observation, aux changements de la géométrie des objets et d'illuminations. l1l2l3 est également invariant aux sur-illuminations. m1m2m3 est indépendant de la couleur de l'illuminant et des inter-réflexions (qui concernent les objets recevant de la lumière reflétée par d'autres objets) contrairement aux deux autres qui y sont très sensibles.

## 1.4 Espaces de représentation couleur

Certains espaces de représentation couleur présentent des caractéristiques invariantes. Nous en présentons une liste non exhaustive mais suffisamment représentative de ceux présentés dans la littérature et que nous avons testés au cours de nos travaux.

**L'espace couleur HSL** (*Hue, Saturation, Lightness* ou *TSL* en français) est défini par un triplet de composantes définissant respectivement la Teinte (*Hue*), la Saturation et la Luminosité (parfois appelée abusivement luminance). Les modèles informatiques de *HSL* recherchent moins la fidélité dans la représentation des couleurs que la simplicité. Cet espace est aussi décrit sous la forme de deux systèmes légèrement différents : *HSV* et *HSL* que nous détaillons ci-après.

La Teinte (*Hue*) caractérise la couleur représentée sur le disque chromatique associé à l'espace couleur *HSL* (figure 1.2). La Saturation représente la chromaticité, autrement dit l'intensité de la

coloration et en ceci sa di rence avec le gris, dit *achromatique*. Une saturation gale 0 repr - sente donc une sensation color e inexistante, alors que cette impression est maximale lorsque la Saturation est 1.

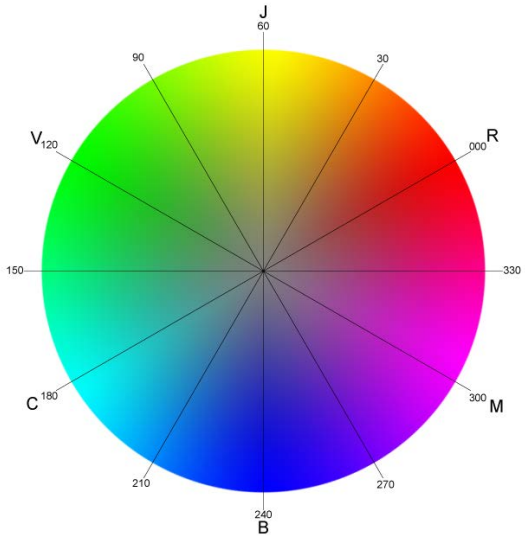



FIGURE 1.2 – Disque chromatique de l'espace HSL

La troisi me composante, selon le syst me employ (*HSV* ou *HSL*), repr sente soit la Valeur (*Value*) soit la *Luminosité*. Elle est d finie en pourcentage et ne repr sente pas tout fait la m me information dans les deux syst mes.

Dans le cas de *HSV*, la simplicit est choisie : la Valeur prend comme valeur maximale la composante trichromatique la plus forte. Ainsi, les couleurs primaires ont une Saturation et une Valeur identiques. En cons quence, le bleu primaire et le blanc ont une Valeur identique, ce qui ne correspond pas l'impression visuelle ressentie car le blanc est g n ralement ressenti comme plus "lumineux" que le bleu primaire.

Dans le cas de *HSL*, la troisi me composante adopte un calcul l g rement plus complexe. Elle prend pour valeur la luminosit moyenne entre les composantes trichromatiques maximale et minimale. Le blanc a alors une luminosit sup rieure la couleur la plus "forte" (except le blanc lui-m me), savoir le jaune le plus lumineux.

L' quation de passage entre les espaces *RGB* et *HSV* est :

$$H = \begin{cases} indefini & \text{si } max(R,G,B) - min(R,G,B) = 0 \\ 60 \times \frac{(G-B)}{max(R,G,B)-min(R,G,B)} mod 360 & \text{si } max(R,G,B) = R \\ 60 \times \frac{(B-R)}{max(R,G,B)-min(R,G,B)} + 120 & \text{si } max(R,G,B) = G \\ 60 \times \frac{(R-G)}{max(R,G,B)-min(R,G,B)} + 240 & \text{si } max(R,G,B) = B \end{cases} \quad (1.19)$$

$$V = max(R,G,B)$$

$$S = \frac{max(R,G,B)-min(R,G,B)}{V}$$

L'équation de passage entre les espaces *RGB* et *HSL* est :

$$H = \begin{cases} \text{indefini} & \text{si } max(R,G,B) - min(R,G,B) = 0 \\ 60 \times \frac{(G-B)}{max(R,G,B)-min(R,G,B)} mod 360 & \text{si } max(R,G,B) = R \\ 60 \times \frac{(B-R)}{max(R,G,B)-min(R,G,B)} + 120 & \text{si } max(R,G,B) = G \\ 60 \times \frac{(R-G)}{max(R,G,B)-min(R,G,B)} + 240 & \text{si } max(R,G,B) = B \end{cases} \tag{1.20}$$

$L = \frac{1}{2}(max(R,G,B) + min(R,G,B))$

$S = \frac{max(R,G,B)-min(R,G,B)}{1-|2L-1|}$

**L'espace couleur L*a*b*** est un espace colorimétrique également standardisé par la CIE [1]. $L^*$ représente la Clarté ; $a^*$ et $b^*$ sont les composantes de Chrominance. Elles expriment l'écart de couleur par rapport à celle d'une surface grise de même clarté. La définition de cette surface grise implique la caractérisation explicite de la lumière qui éclaire la surface colorée (*illuminant*). On considère généralement cet illuminant comme étant la lumière du jour, normalisée D65.

Cette représentation ressemble au fonctionnement du système visuel qui fonctionne par contraste entre rouge et vert, et entre bleu et jaune. Chacune de ces composantes est déterminée à l'aide d'un spectrophotomètre. Prises séparément, les composantes $a^*$ et $b^*$ représentent chacune une gamme de 600 niveaux de couleur sur un axe respectivement rouge-vert et jaune-bleu, les valeurs de cet axe étant compris entre +299 et -300. Néanmoins, ces valeurs sont le plus souvent notées entre -127 et +128 afin de pouvoir être codées sur 8 bits, à l'instar d'autres espaces comme RGB.

Le passage entre les espaces *RGB* et *CIE L*a*b** nécessite une transition par l'espace *CIE XYZ* dont dérive L*a*b*. On a donc l'équation suivante :

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.418456 & -0.158657 & -0.082833 \\ -0.091167 & 0.252426 & 0.015707 \\ 0.000921 & -0.002550 & 0.178595 \end{bmatrix} . \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{1.21}$$

puis :

$$\begin{aligned} L^\star &= 116 f(Y/Y_n) - 16, \\ a^\star &= 500 \left[ f(X/X_n) - f(Y/Y_n) \right], \\ b^\star &= 200 \left[ f(Y/Y_n) - f(Z/Z_n) \right], \end{aligned} \tag{1.22}$$

où

$$f(t) = \begin{cases} t^{1/3} & \text{si } t > (\frac{6}{29})^3, \\ \frac{1}{13} \left(\frac{29}{6}\right)^2 t + \frac{4}{29} & \text{sinon.} \end{cases} \tag{1.23}$$

et $X_n, Y_n, Z_n$ sont les composantes du blanc de référence décrit dans l'espace XYZ.

**L'espace couleur YCh1Ch2** fut introduit en 1995 par Carron et al. [10]. Il est similaire aux autres systèmes de Luminance-Chrominance dans le sens où Y la Luminance et les deux composantes *Ch1* et *Ch2* représentent une information chromatique opposant respectivement cyan-rouge et vert-bleu.

La transformation de l'espace *RGB* à l'espace *YCh1Ch2* est donnée par :

$$\begin{bmatrix} Y \\ Ch1 \\ Ch2 \end{bmatrix} = \begin{bmatrix} 0.299 \times R + 0.587 \times G + 0.114 \times B \\ R - \frac{G+B}{2} \\ \frac{\sqrt{3}}{2} \times (B - G) \end{bmatrix} \tag{1.24}$$

**L'espace couleur YUV** définit également un espace colorimétrique basé sur un système de Luminance-Chrominance. Il est notamment utilisé dans le standard de diffusion télévisuelle PAL. Y représente la Luminance, U et V représentent la Chrominance. La transformation de l'espace RGB à l'espace YUV est donnée par l'équation :

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.25}$$

**L'espace couleur YCbCr** est un espace basé sur la constatation que l'œil humain est plus sensible à la Luminance qu'à la Chrominance. Ceci implique que la qualité colorimétrique d'une image peut être dégradée tout en conservant un aspect visuel suffisant pour une bonne visualisation. C'est le système qui est utilisé dans le format d'images JPEG, qui est un format de compression avec pertes. Dans ce cas précis, les couleurs originelles de l'image sont dégradées pour en limiter le nombre autant que possible, tout en restant le plus fidèle à l'image originale. Cet espace transformé depuis l'espace RGB est donné par l'équation suivante :

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.26}$$

**L'espace couleur YIQ** définit également un espace colorimétrique basé sur un système de Luminance-Chrominance. A l'instar de YUV, il est utilisé pour la diffusion télévisuelle mais dans le standard NTSC. $Y$ représente la Luminance, $I$ est appelée la composante en phase et $Q$ la composante en quadrature de la Chrominance. La transformation de l'espace RGB à l'espace YIQ est donnée par l'équation :

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.321 \\ 0.211 & -0.523 & 0.311 \end{bmatrix} . \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.27}$$

## 1.5 Invariants basés sur un apprentissage

À l'instar des méthodes de segmentation produisant un modèle à partir de données d'apprentissage, certaines méthodes effectuent l'estimation de l'illuminant après un apprentissage effectué pendant une certaine durée. C'est le cas de Gehler *et al.* [30] et Rosenberg *et al.* [67] qui utilisent une approche Bayésienne au lieu de la méthode statique Greyworld précédemment décrite, ainsi que d'autres algorithmes comme Grey-edge [77] pouvant être eux-même décrits comme des instances spécifiques de l'approche Bayésienne. Grey-edge se base sur l'hypothèse que la moyenne des différences de réflectance d'une scène est achromatique.

Gehler *et al.* indiquent que l'estimation Bayésienne semble apporter des résultats significativement supérieurs à cette dernière en termes de valeurs d'erreurs quadratiques entre ces méthodes.

Les tests ont été effectués à la fois sur des images intérieures et extérieures pour lesquelles l'illuminant a été déterminé avec précision (à l'aide d'un Color Checker de MacBeth) afin de fournir une référence fiable. Gehler *et al.* confirment dans leurs travaux les résultats de Rosenberg *et al.* montrant une performance de l'approche Bayésienne légèrement meilleure que Greyworld.

Ces méthodes, bien qu'intéressantes pour les applications de détection des objets mobiles utilisant un apprentissage, n'ont pas été privilégiées pour les travaux de cette thèse et pourront faire l'objet de futurs tests.

## 1.6 Synthèse des différents invariants colorimétriques et espaces de représentation couleur

La littérature offre un bon nombre d'invariants colorimétriques et d'espaces de représentation couleur présentant des propriétés invariantes. Nous avons sélectionné un certain nombre d'entre eux en essayant d'être le plus représentatif possible parmi les différentes familles existantes. Le tableau 1.1 récapitule les invariants colorimétriques et espaces de représentation couleur retenus avec la dénomination (courte) qui sera utilisée dans les tableaux de résultats présentés dans le chapitre 2. La figure 1.3 illustre pour une image donnée le résultat de l'application d'un invariant colorimétrique ou espace de représentation couleur. Au vu de ces illustrations, nous sommes persuadés que la simplification d'une image par l'application d'invariants colorimétriques ou espaces de représentation couleur permettra d'améliorer grandement les résultats de détection d'objets mobiles dans les images que nous traitons dans cette thèse et qui révèlent de fortes contraintes en termes de changements d'éclairage, que ce soit en environnement intérieur ou extérieur.

| **Intitulé** | **Invariant colorimétrique** | **Intitulé** | **Invariant colorimétrique** |
|---|---|---|---|
| rgb | Espace couleur RGB | m1m2m3 | Espace couleur m1m2m3 |
| cs | Chromaticity Space | l1l2l3 | Espace couleur l1l2l3 |
| gw | Grey World | hsl | Espace couleur HSL |
| cn | Comprehensive Normalization | yiq | Espace couleur YIQ |
| an | Affine Normalization | ycbcr | Espace couleur YCbCr |
| rgb-r | RGB-Rang | lab | Espace couleur L*a*b* |
| c1c2c3 | Espace couleur c1c2c3 | | |

TABLE 1.1 – Correspondances entre les noms des invariants colorimétriques et espaces de représentation couleur utilisés, et leur dénomination courte affichée dans les résultats des chapitres suivants.

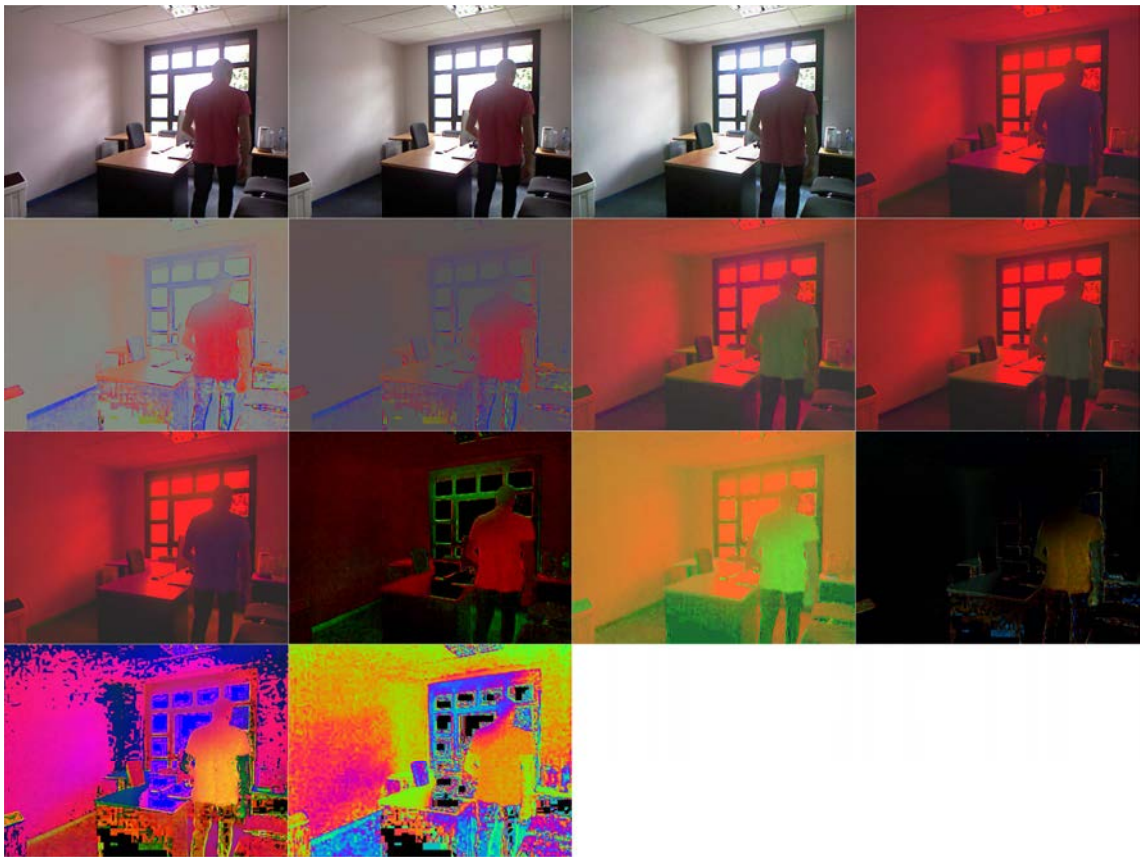


FIGURE 1.3 – Illustration de l'application d'invariants colorim triques (de gauche droite et de haut en bas : image originale (RGB), Grey-World, RGB-Rang, YUV, c1c2c3, Chromaticity Space, YCh1Ch2, YIQ, YCbCr, Espace des couleurs oppos es, L*a*b*, m1m2m3, HSL, l1l2l3)

# 2 Méthodes de segmentation fond/forme

La segmentation d'images est une discipline du traitement d'images visant à regrouper des informations contenues dans des images en utilisant un ou plusieurs critères pré-définis. Ces regroupements d'informations se caractérisent généralement par des ensembles de pixels formant des régions de l'image qui, ainsi séparées des autres, sont ensuite utilisés pour d'autres traitements ultérieurs, comme un suivi temporel ou une reconnaissance d'objets.

L'extraction de ces régions peut être effectuée de diverses manières. Depuis plusieurs années, la littérature scientifique abonde de méthodes proposées que l'on peut classer en fonction des critères utilisés pour caractériser les régions d'intérêt. Dans le cadre de la détection et du suivi des objets mobiles dans des séquences d'images, des méthodes de segmentation (visant à séparer les objets mobiles de l'arrière-plan de l'image) sans modélisation du fond, qui reposent généralement sur des techniques de seuillage, de gradient temporel ou spatio-temporel. Elles s'appliquent uniquement sur les frames courante et précédente de la séquence d'images, et permettent d'isoler les objets mobiles sur un fond statique, en assumant selon les cas que ces objets sont de couleur(s) différente(s) du fond ou qu'ils sont en mouvement constant. Ces limitations, si elles correspondent partiellement à notre problématique, sont cependant trop incertaines pour une application opérable en environnement extérieur soumis à des changements de couleurs et de luminosités importants.

C'est pourquoi dans cette section, nous présenterons différentes méthodes de segmentation connues pour leurs bons résultats utilisant notamment une modélisation du fond et qui permettent de segmenter une image en différentes régions d'intérêt, même dans les situations difficiles énoncées ci-après.

## 2.1 Problématique

Une méthode de segmentation fond/forme idéale devrait être capable de surmonter l'ensemble des différentes difficultés rencontrées dans les séquences réelles et énoncées ci-après, mais ceci est une tâche difficile qu'aucune méthode ne surmonte totalement actuellement.

— Objet déplacé : il s'agit d'un objet fixe, appartenant normalement au fond, qui a été déplacé. Cet objet ne devrait pas être considéré comme un objet mobile après un temps donné.

— Moment de la journée : les changements graduels de l'illumination au cours de la journée altèrent l'apparence de l'arrière-plan.

— Lumière allumée/éteinte : il s'agit d'un changement brusque de la luminosité qui altère l'apparence de la scène.

— Feuillage des arbres : certaines parties d'arrière-plan peuvent vaciller de façon plus ou moins cyclique (notamment à cause du vent). Ce type de difficulté nécessite donc de gérer différentes observations pour les mêmes régions de l'image.

— Occultation : un objet mobile peut passer derrière un élément de l'arrière-plan et devient de ce fait occulté durant un laps de temps donné.

— *Bootstrapping* : ce terme correspond à des cas d'utilisation où les images contiennent toujours un ou des objets mobiles, y-compris pendant l'apprentissage. Ce genre de situation nécessite que l'algorithme soit en mesure d'éliminer les potentiels objets mobiles présents pendant l'apprentissage afin que le modèle représentant l'arrière-plan n'en soit pas impacté.

— Objet de couleur homogène : Il s'agit d'un objet de couleur homogène qui se déplace. Il est difficile de détecter les pixels se déplaçant en son centre, ce qui peut conduire à la non-détection de l'objet complet.

— Voiture parquée : il s'agit d'un objet mobile devenant immobile et qui ne peut donc être distingué d'un élément mobile de l'arrière-plan qui s'immobilise également.

— Réveil d'objet mobile : il s'agit d'un un objet de l'arrière-plan qui se déplace. Deux régions peuvent alors être considérées comme mobiles peuvent apparaître : l'objet mobile lui-même, et l'espace vide qu'il a laissé derrière lui et qui n'est alors pas reconnu comme arrière-plan.

— Ombres : les objets mobiles peuvent projeter des ombres sur l'arrière-plan, ce qui crée des régions différentes du fond et peuvent ne pas être considérées comme tel par l'algorithme et faire partie de la région voisine.

Les méthodes de segmentation fond/forme peuvent agir à différents niveaux de l'image :

— Niveau pixel : la représentation de l'arrière-plan est effectuée sur la base d'un élément unitaire représentant un seul pixel à la fois et ignore les informations des pixels environnants. Cela permet d'éviter, dans une certaine limite, plusieurs des problèmes précédents (objet déplacé, moment de la journée, feuillage des arbres, occultation, bootstrapping).

— Niveau région : le modèle prend en considération le voisinage des pixels pour aider à caractériser chacun des pixels de l'image, ce qui peut aider à la résolution des problèmes d'objets de couleur homogène.

— Niveau image : le modèle prend en considération l'ensemble des images, et permet de tenir compte de l'aspect temporel et de détecter les changements importants impliqués dans sa globalité (lumière allumée/éteinte).

Depuis plusieurs décennies, de nombreuses méthodes de segmentation fond/forme ont été proposées dans la littérature. Outre la modélisation simple basée sur la soustraction frame à frame, la méthode la plus fréquemment employée consiste à modéliser le fond de la scène. Nous retrouvons ainsi différentes méthodes que nous avons catégorisées en différentes familles que sont la modélisation basique du fond, la modélisation statistique, la modélisation floue, la modélisation par clustering et la modélisation par estimation du fond. La figure 1.4 ainsi que les paragraphes suivants décrivent les principales méthodes de chacune de ces familles.

## 2.2 Segmentation par modélisation basique du fond

Basiquement, la détection des régions mouvantes dans une image consiste en la soustraction d'une première image de fond $BG_{t-1}$ (sans aucun objet mobile) au temps $t-1$, avec l'image courante $I_t$ prise au temps $t$. L'image résultante $BGS$ est donnée par l'équation suivante pour un pixel $p$ de coordonnées $(i,j)$ :

$$BGS(i,j) = \begin{cases} 1 & \text{si } |I_t(i,j) - BG_{t-1}(i,j)| > \epsilon_s \\ 0 & \text{sinon} \end{cases} \tag{1.28}$$

L'image $BGS$ qui résulte de cette soustraction est une image binaire dont les pixels de valeur 1 correspondent à des objets mobiles, et les pixels de valeur 0 à du fond. Cette méthode est représentative du concept même de la segmentation par soustraction d'arrière-plan, en anglais *background subtraction*. Une étape importante de cette méthode réside dans l'obtention de l'image d'arrière-plan BG.

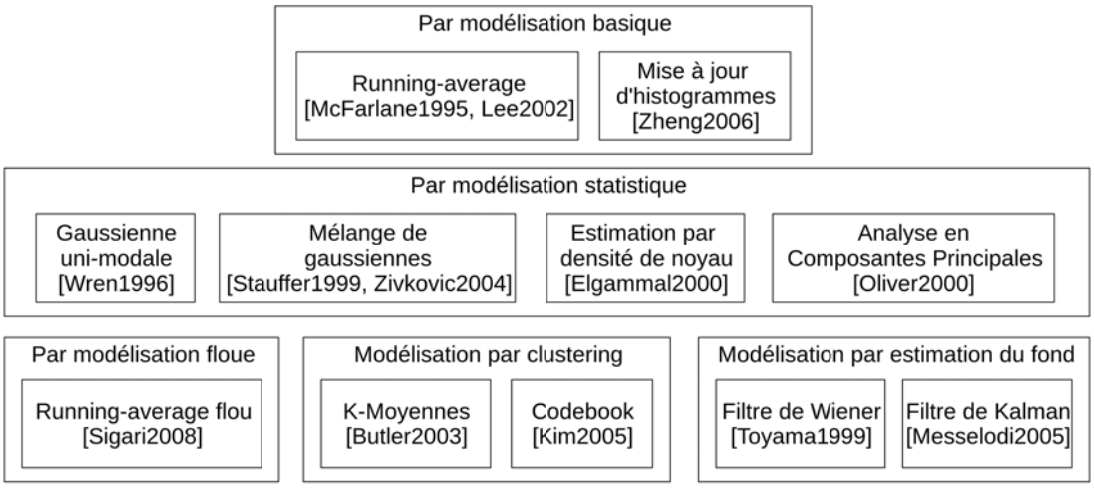



FIGURE 1.4 – Les di rentes familles de m thodes de segmentation.

Pour ce faire, une des plus communes mani res consiste calculer une image moyenn e sur plusieurs frames d'apprentissage (m thode commun ment appel e *running average*). Ce mod le doit n cessairement tre tenu jour cause des changements d'illumination de la sc ne et peut tre e ectu e par la relation suivante :

$$BG_t = \alpha.BG_{t-1} + (1-\alpha).I_t \tag{1.29}$$

avec $\alpha \in [0.5, 1]$, sachant qu'une valeur proche de 0.5 constitue une moyenne *court-terme*, alors qu'une valeur proche de 1 constitue une moyenne *long-terme*.

Parmi les autres m thodes propos es, nous pouvons citer McFarlane et al. [52] qui r alise une soustraction entre la frame observ e et une image m diane de fond (au lieu de l'image moyenne). Cette m thode a l'avantage d' tre tr s rapide et peu co teuse en ressources. Cependant, elle s'av re tr s vite limit e en situation non contr l e. En e et, la mise jour de cette image m diane n cessite d'avoir une sc ne vide ou des objets constamment en mouvement afin qu'ils n'y soient pas int gr s.

Une autre m thode propos e par Zheng et al. [82] consiste construire un histogramme de valeurs pour chaque canal rouge, vert et bleu de chaque pixel. Cet histogramme permet alors de d terminer la tendance g n rale de la couleur du pixel consid r . Ainsi, si la couleur observ e pour ce pixel dans une nouvelle image se situe hors des valeurs les plus observ es, alors ce pixel est consid r obstru par un objet mobile.

## 2.3 Segmentation par modélisation statistique

Une autre famille de m thodes de segmentation fond/forme se base sur la mod lisation statistique o l'on voit appara tre des m thodes bas es sur des Gaussiennes, une estimation par densit de noyau, une analyse en composantes principales, etc.

Wren et al. [80] choisissent de mod liser chaque pixel de l'image par une Gaussienne relative sa couleur moyenne. L'adaptation de cette Gaussienne au fil du temps s'e ectue en mettant jour la moyenne et la variance par une moyenne glissante. Cependant, force est de constater qu'une seule Gaussienne ne permet pas de mod liser e cacement un fond contenant des r gions en mouvement, qu'il soit r gulier ou al atoire (comme par exemple avec le mouvement du feuillage

d'un arbre). Stauffer et al. [73] proposent une technique basée sur un Mélange de Gaussiennes (MOG en anglais).

La technique proposée par Stauffer et Grimson [73] se base sur une méthode probabiliste dont le principe est le suivant : on modélise chaque pixel de l'image comme un mélange (mixture) de gaussiennes séparées. La détection des objets mobiles est faite, pour chaque pixel, par le matching ou non d'une distribution du modèle avec celle de la frame observée grâce à une mesure de distance. On met ensuite à jour ce modèle en adaptant lentement les valeurs des gaussiennes pour chaque pixel.

Cette méthode nécessite la définition de deux paramètres :

1. $\alpha$ : le taux d'apprentissage ;
2. $T$ : la mesure de la portion minimum de données qui sera prise en compte pour l'arrière-plan (nombre de modes).

Si $T$ est petit, alors le modèle est unimodal, c'est-à-dire que le modèle représente l'arrière-plan en utilisant une seule distribution (la plus probable). Si T est grand, alors la distribution est multi-modale, ce qui permet au modèle de stocker plusieurs couleurs pour le même pixel, ce qui permet de prendre en compte les ombres et les zones oscillantes (branches d'arbres, écrans d'ordinateurs, etc). T est généralement compris entre 3 et 5, mais dépend du hardware. Zivkovic [83] améliore cette méthode en proposant de déterminer automatiquement le nombre de composantes gaussiennes nécessaires pour chaque pixel du modèle. Ceci permet de réduire l'espace nécessaire au stockage du modèle en mémoire ainsi que d'améliorer sensiblement la qualité de la segmentation.

La probabilité d'observer un pixel $x_t$ est donnée par la formule suivante :

$$P(x_t) = \sum_{i=1}^{K} \omega_{i,t}.\eta(x_t, \mu_{i,t}, \Sigma_{i,t}) \tag{1.30}$$

où $K$ est le nombre de gaussiennes associées au pixel et $\omega_{i,t}$ est un poids associé à chaque gaussienne qui définit son importance dans la détection ; $\mu_{i,t}$ est la moyenne et $\Sigma_{i,t}$ la variance.

Pour chaque pixel de l'image observé, on détermine une gaussienne définie par la fonction de densité de probabilité $\eta$ :

$$\eta(x_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t-\mu_t)^T \Sigma^{-1}(x_t-\mu_t)} \tag{1.31}$$

avec $\Sigma$ la matrice de covariance de la gaussienne et $\mu_t$ la valeur moyenne de la gaussienne. Le cas général consiste à utiliser une matrice de covariance de taille 3x3, tandis que la méthode améliorée proposée par [83] effectue un traitement séparé pour chaque canal.

Soit $X_i = \{RGB\}^T$ un vecteur représentant la couleur d'un pixel $i$ appartenant à l'image $I$. La matrice de covariance est calculée à l'aide de 2 vecteurs. Ici on assume que les valeurs R, G et B de $X_i$ sont indépendantes et ont la même variance, ce qui n'est certainement pas le cas mais qui évite des calculs d'inversion de matrice coûteux au prix d'une légère imprécision.

La matrice de covariance est donc de la forme :

$$\Sigma_{k,t} = \sigma_k^2 I \tag{1.32}$$

Chaque pixel étant caractérisé par son intensité dans l'espace RGB, on représente le modèle avec un mélange de K gaussiennes. Puis, pour chaque pixel, on considère uniquement les gaussiennes du

modèle qui dépassent un certain seuil T. Ensuite, à chaque nouvelle frame (t+1) on cherche une correspondance pour chaque pixel avec une gaussienne du modèle selon la distance de Mahalanobis. On considère qu'un pixel $x_{t+1}$ correspond à une gaussienne lorsque :

$$\sqrt{(x_{t+1} - \mu_{i,t})^T . \Sigma_{i,t}^{-1} . (x_{t+1} - \mu_{i,t})} < k\sigma_{i,t} \tag{1.33}$$

avec $k = 2.5$.

Deux cas peuvent alors survenir :

1. Une correspondance est trouvée avec une des K gaussiennes. Dans ce cas si la distribution gaussienne est identifiée comme arrière-plan, le pixel est identifié comme arrière-plan ; sinon il est identifié comme appartenant à une forme.
2. Aucune correspondance n'est trouvée parmi les K gaussiennes. Dans ce cas, le pixel est identifié comme appartenant à une forme.

À l'issue de cette étape, il est nécessaire de mettre à jour le modèle d'arrière-plan pour permettre une détection sur la prochaine frame observée.

Comme pour la détection du premier-plan, deux cas peuvent survenir :

1. Une correspondance est trouvée avec une des K gaussiennes. Pour la gaussienne correspondante, on met à jour ses paramètres :

   — La valeur de poids $\omega$ :
   $$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha \tag{1.34}$$

   — La moyenne $\mu$ :
   $$\mu_{i,t+1} = (1 - \rho)\mu_{t-1} + \rho X_t \tag{1.35}$$

   — L'écart-type $\sigma$ :
   $$\sigma_{i,t+1}^2 = (1 - \rho)\sigma_t^2 + \rho(X_{t+1} - \mu_{i,t+1})^T . (X_{t+1} - \mu_{i,t+1})^T \tag{1.36}$$

   avec
   $$\rho = \alpha\eta(X_{t+1}, \mu_i, \sigma_i) \tag{1.37}$$

   Pour les $K - 1$ gaussiennes non matchées, on ne modifie que la valeur de poids $\omega$ :

   $$\omega_{j,t+1} = (1 - \alpha)\omega_{j,t} \tag{1.38}$$

2. Aucune correspondance n'est trouvée parmi les K gaussiennes. Dans ce cas, on remplace la distribution gaussienne la moins probable par une nouvelle distribution définie par les paramètres suivants :

   — La valeur du poids $\omega$

   — La moyenne $\mu$

   — L'écart-type $\sigma$

Au vu de la définition de cette méthode de détection, il apparaît qu'elle nécessite un paramétrage qui s'avère à la fois important et technique. En effet, ce paramétrage impacte directement la sensibilité de l'apprentissage de l'algorithme ainsi que sa mise à jour. Une mise à jour trop rapide peut conduire à l'absorption des objets mobiles qui se déplacent lentement ; au contraire, une mise

à jour trop lente rend l'algorithme plus sensible aux changements de luminosité pouvant survenir dans la scène.

Néanmoins, cette méthode s'avère plus efficace que la modélisation par Gaussienne unimodale, même si elle reste sensible à la présence d'ombres portées et aux variations rapides d'intensité lumineuse. Par ailleurs, cette méthode est semi-paramétrique et nécessite la définition d'une variable d'apprentissage définissant la sensibilité de la mise à jour du modèle. Une sensibilité trop faible aura pour effet de produire de nombreux faux-positifs ; au contraire une sensibilité trop forte causera l'absorption plus ou moins partielle des objets mobiles dans le modèle. Malgré ces limitations, cette méthode reste l'une des plus populaires encore aujourd'hui. Zivkovic [83] propose une optimisation de l'algorithme de Mixture de Gaussiennes avec la mise à jour constante des paramètres ainsi que la détermination automatique du nombre idéal de gaussiennes utilisées pour décrire chaque pixel du modèle.

Elgammal et al. [19] proposent une méthode différente basée sur une estimation par densité de noyau qui a l'avantage d'être plus efficace sur des régions contenant des mouvements légers et répétitifs. Le noyau est estimé par la déviation de la valeur absolue de la médiane sur plusieurs échantillons d'images dont l'intensité est proche pour chaque pixel. Cette estimation peut être imprécise si la distribution de ces échantillons est multimodale, la médiane n'étant pas représentative dans ce cas. Cette méthode présente aussi l'avantage de ne pas nécessiter de paramétrage mais son coût calculatoire reste important, et nécessite donc d'importantes quantités de mémoire, même si l'utilisation de tables de recherche pré-calculées permet de réduire la charge de traitement de l'algorithme.

Oliver et al. [59] proposent quant à eux une modélisation basée sur une Analyse en Composantes Principales. Chaque pixel de l'arrière-plan est modélisé par un modèle d'arrière-plan en fonction des valeurs propres (Eigenbackground). Ce modèle permet une modélisation robuste du fond à condition que les objets mobiles soient de petite taille et ne restent pas immobiles trop longtemps pendant la période d'apprentissage. Cette méthode est relativement lourde en temps de calcul en ce qui concerne la mise à jour du modèle. Enfin, il est important de noter que cette méthode fut à l'origine proposée avec des images en niveau de gris, et que l'utilisation de composantes supplémentaires implique une augmentation des dimensions de l'espace et donc une plus grande complexité.

## 2.4 Segmentation par modélisation floue

La Logique Floue est une forme de logique qui se différencie de l'algèbre de Boole par le fait qu'elle permet de décrire un concept d'une manière plus proche du mode de réflexion humain. Plus précisément, l'algèbre de Boole définit nécessairement des variables *vraies* et *fausses*, alors que la Logique floue introduit l'idée de variables décrivant une "vérité partielle", dont la valeur peut s'étendre de totalement vrai à totalement faux.

Un exemple simple d'une application de la logique floue consiste à classer des personnes dans trois classes, "petit", "moyen" et "grand". Supposons qu'en dessous de 1m60 on considère qu'une personne est "petite", qu'en dessous de 1m80 elle est "moyenne" et qu'au-dessus de 1m80, elle est "grande". Si l'on s'en tient à cette séparation nette des trois groupes, une personne mesurant 1m79 sera classée dans la classe "moyen", alors qu'elle est manifestement extrêmement proche de la classe "grand". Le principe de la logique floue est de considérer qu'une personne peut être classée dans plusieurs groupes à la fois. Ces appartenances sont régies par des règles pré-définies. Il en résulte

un indice de confiance pour chacun des groupes. Dans notre exemple, une personne mesurant 1m79 sera classée à la fois dans les groupes "moyen" et "grand", avec un indice de confiance plus important pour la classe "grand" que la classe "moyen". Ces indices de confiance sont ensuite utilisés pour prendre une décision finale. Cette explication très schématique résume rapidement l'idée de la logique floue, utilisée dans ce contexte de segmentation fond/forme.

Sigari et al. [71] proposent une méthode basée sur la logique floue (FBGS - *Fuzzy Background Subtraction*) en vue d'améliorer les résultats apportés par la méthode de l'image moyenne (*running average*) décrite dans la famille "segmentation par modélisation basique". L'équation 1.39 détaille cette méthode :

$$FBGS(i,j) = \begin{cases} 1 & \text{si } |I_t(i,j) - BG_{t-1}(i,j)| > \epsilon_s \\ \frac{|I_t(i,j) - BG_t(i,j)|}{\epsilon_s} & \text{sinon} \end{cases} \tag{1.39}$$

Il apparaît fort logiquement que le résultat de la segmentation n'est plus binaire, mais composé de valeurs réelles comprises dans l'intervalle $[0,1]$, où 1 représente les pixels appartenant à des objets mobiles et où les autres valeurs correspondent à du fond. Pour obtenir une segmentation binaire, l'image $FBGS$ est binarisée à l'aide d'un filtre passe-bas (LPF) comme le décrit l'équation 1.40 :

$$BGS(i,j) = \begin{cases} 1 & \text{si } |LPF(FBGS(i,j))| > \epsilon_{fs} \\ 0 & \text{sinon} \end{cases} \tag{1.40}$$

Le filtre passe-bas utilisé peut être un filtre moyenneur de taille $3 \times 3$. Le seuil $\epsilon_{fs}$ détermine la valeur en dessous de laquelle le pixel peut être considéré comme appartenant au fond. Cette méthode se différencie également par le mode de mise à jour du modèle. En effet, il considère une valeur $\alpha$ différente pour chaque pixel donnée par la relation suivante :

$$\alpha(i,j) = 1 - (1 - \alpha_{min}) exp(-5 \times FBGS(i,j)) \tag{1.41}$$

Dans cette dernière équation, la valeur $\alpha_{min}$ est la valeur minimale pour $\alpha$ (les auteurs proposent une valeur de 0.9). La mise à jour du modèle s'effectue alors selon la relation suivante :

$$BG_t(i,j) = \alpha(i,j).BG(i,j) + (1 - \alpha).I_t(i,j) \tag{1.42}$$

## 2.5 Segmentation par estimation du fond

Une autre famille de méthodes de segmentation concerne l'estimation du fond. Ces techniques se basent sur des méthodes de prédiction afin de déterminer le prochain état probable de chaque pixel. Ces méthodes englobent entre autres les méthodes de prédiction tels que les filtres de Wiener, de Kalman ou de Chebytchev.

La méthode Wallflower [76] est une des premières propositions efficientes de la mise à jour d'un modèle d'arrière-plan. Elle fonctionne sur les trois niveaux de l'image (pixel, région et image). Au niveau pixel, l'algorithme effectue une prédiction probabiliste du prochain état attendu pour chaque pixel à l'aide d'un filtre prédictif de Wiener à 1 étape. Cette prédiction semble fonctionner correctement pour des pixels dont les valeurs sont périodiques. Au niveau région, l'algorithme se base sur l'hypothèse que les objets mobiles (dont la couleur est homogène) peuvent être segmentés en plusieurs sous-régions (qu'il est possible de re-connecter) et ainsi affiner la détection préliminaire obtenue au niveau pixel. Enfin, au niveau image, l'algorithme stocke plusieurs modèles d'arrière-

plan différents obtenus à l'aide d'un clustering effectué par l'algorithme des k-moyennes. Le modèle finalement retenu est celui qui produit le moins de pixels considérés comme objets mobiles.

Messelodi *et al* [53] proposent également une méthode par prédiction de l'état des pixels à l'aide cette fois-ci du Filtre de Kalman. Cependant, toutes les variations de l'arrière-plan ne peuvent pas être considérées comme du bruit. Les changements d'illumination globale sont des événements extérieurs différents de la simple présence d'objets mobiles. Aussi, l'estimation de ces différents changements est traitée séparément, à savoir les changements d'illumination globale d'une part et la variance du bruit d'autre part.

Ces techniques, bien qu'offrant de bons résultats, deviennent plus difficilement applicables dans un contexte d'images couleur. En effet, les temps de traitement sont multipliés par le nombre de canaux présents dans l'image. Par ailleurs, l'efficacité de ces méthodes dans le cas du traitement d'images couleur est beaucoup moins connue de la littérature.

## 2.6 Segmentation par clustering

La dernière famille de méthodes de segmentation fond/forme proposée dans notre découpage regroupe des méthodes basées sur le clustering. La segmentation et le clustering sont deux concepts qui peuvent sembler proches.

La définition de la segmentation est relativement simple : il s'agit de "diviser en plusieurs entités en fonction de certains critères". Dans notre exemple, la segmentation fond/forme est constituée de deux entités, arrière-plan et forme, qui sont définies par des critères différents : le fond est statique alors que les formes sont la plupart du temps mobiles, ou encore le fond est défini par des couleurs précises qui ne sont pas celles des formes qui le traversent, etc.

La définition du clustering nécessite quant à elle la définition d'un "cluster". Il s'agit d'un terme anglais pour définir ce qu'en français nous pouvons appeler un groupe ou une partition. On retrouve ce terme dans "Cluster Analysis" (Analyse de partitionnement) qui consiste à réunir des objets dans différents groupes (ou clusters) de telle manière que les objets d'un même groupe soient plus similaires entre eux qu'avec ceux d'un autre groupe.

Cette tâche de clustering peut être réalisée à l'aide de différentes méthodes, parmi lesquelles la bien connue méthode des K-moyennes utilisée entre autres par Butler et al. [9]. Cette méthode est effectivement économe en coût de traitement, mais elle reste comme beaucoup d'autres sensible aux ombres portées par les objets mobiles ainsi qu'aux régions multi-modales de l'arrière-plan.

Kim et al. [47] proposent une segmentation fond/forme appelée Codebook. Elle consiste en un clustering des échantillons obtenus pour chaque pixel pendant une période d'apprentissage dans un ensemble de codewords selon deux critères : la couleur et la brillance. Cette technique permet d'obtenir de bons résultats de détection et ce dans de nombreux cas d'utilisation. Il est aussi important de noter que cette méthode est robuste à la fois aux faibles changements d'illumination et aux ombres portées, et aux pixels des régions multimodales du fond (pris en compte dans son modèle d'arrière-plan.

Au vu de la réputation de cette méthode et de ses nombreux avantages pour notre domaine d'application, nous avons fait le choix de nous y intéresser davantage dans cette thèse et la décrivons donc en détails ci-après.

**Segmentation par la méthode Codebook** L'algorithme Codebook (CB) proposé par Kim et al. [47] et découpé en deux phases consiste en la construction d'un modèle d'arrière-plan représenté

par une collection de *codewords* regroupés dans un *codebook*, à raison d'un codebook par pixel, suivi d'une détection des objets mobiles pour arriver à une segmentation fond/forme. Chaque codebook en revanche ne contient pas forcément le même nombre de codewords : ce nombre dépend de la variation de l'arrière-plan. Une représentation de ce modèle est illustrée sur la figure 1.5.

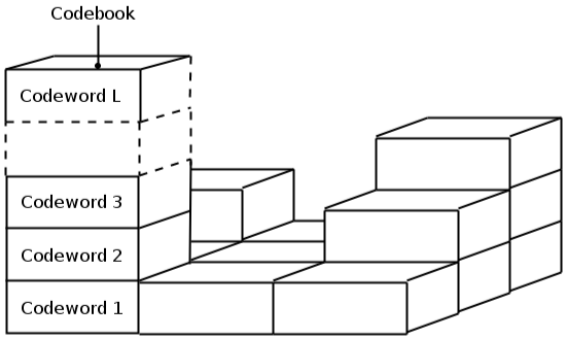



FIGURE 1.5 – Représentation des codebooks d'un groupe de pixels voisins.

Soit le codebook d'un pixel donné, contenant $L$ codewords. Chaque codeword $c_m \in c_1, c_2, ..., c_L$ est une structure définie par deux vecteurs :

1. un vecteur $v_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ contenant les composantes couleurs moyennes de ce codeword ;
2. un 6-tuple $aux_m = (\check{I}_m, \hat{I}_m, f_m, \lambda_m, p_m, l_m)$ avec $\check{I}_m$ et $\hat{I}_m$ respectivement les brillances minimum et maximum assignées au codeword ; $f_m$ la fréquence d'observation de ce codeword, autrement dit le nombre de fois que ce codeword a été mis en correspondance avec une observation ; $\lambda_m$ la MNRL (*Maximum Negative Run-Length*) correspondant à la plus longue durée pendant laquelle ce codeword n'a *pas* été mis en correspondance avec une observation ; $p_m$ et $l_m$ représentant respectivement le premier et le dernier temps d'accès à ce codeword.

Cet algorithme construit pendant une première période d'apprentissage un modèle d'arrière-plan basé sur l'architecture précédemment décrite, puis intervient une seconde étape dans laquelle une segmentation est obtenue. Cependant, il a été observé que pour la plupart des méthodes de détection fond/forme de la littérature se basant sur la couleur seule, les faux positifs avaient tendance à se grouper dans les zones sombres de l'image [38]. Aussi, afin de prendre en compte les changements de luminosité de la scène, la couleur et la brillance sont deux informations qui doivent être considérées séparément. En effet, la distribution de la couleur d'un pixel en fonction du temps pour une scène soumise à des changements de luminosité forme un ensemble de points 3D (R, G, B) dispersé sur un axe aligné avec l'origine (0,0,0). On peut donc supposer que la couleur d'un pixel évolue principalement sur sa brillance.

Kim et al. proposent un modèle obtenu par décomposition des composantes couleur de chaque pixel en deux conditions : 1) la distorsion de couleur et 2) la distorsion de brillance. La première condition a pour objectif de déterminer une valeur numérique représentant l'écart entre la couleur d'un pixel observé et celle d'un codeword. La seconde vérifie que la brillance du pixel observé est bien similaire à celle du codeword comparé, c'est-à-dire que la brillance doit être comprise dans l'intervalle de la brillance minimale à la brillance maximale du codeword comparé. Soit une observation $x_t = (R, G, B)$, et un codeword, dont nous souhaitons établir la correspondance ou non, composé des deux vecteurs $v_m$ et $aux_m$ comme décrit précédemment :

1. La distorsion de couleur, notée $\delta$, entre un pixel $x_t$ et un codeword caractérisé par son premier vecteur $v_m$ est calculée de la façon suivante :

$$\begin{aligned} colorDist(x_t, v_m) &= \sqrt{\|x_t\|^2 - p^2}, \\ p^2 &= \|x_t\|^2 cos^2\theta = \frac{\langle x_t, v_m\rangle^2}{\|v_m\|^2} \end{aligned} \tag{1.43}$$

avec

$$\begin{aligned} \|x_t\|^2 &= R^2 + G^2 + B^2, \\ \|v_m\|^2 &= \bar{R}_m^2 + \bar{G}_m^2 + \bar{B}_m^2, \\ \langle x_t, v_m\rangle^2 &= (\bar{R}_m R + \bar{G}_m G + \bar{B}_m B)^2 \end{aligned} \tag{1.44}$$

2. La distorsion de brillance est alors calculée selon l'équation suivante :

$$brightnessDist(I, \langle \check{I}, \hat{I}\rangle) = \begin{cases} vrai & \text{si } I_{low} \leq \|x_t\| \leq I_{high}, \\ faux & \text{sinon} \end{cases} \tag{1.45}$$

avec

$$\begin{aligned} I_{low} &= \alpha \hat{I}_m, \\ I_{high} &= min\{\beta \check{I}_m, \frac{\check{I}_m}{\alpha}\} \end{aligned} \tag{1.46}$$

où $\alpha < 1$ et $\beta > 1$ qui sont des seuils fixés et donnés en paramètre de l'algorithme, et où $\hat{I}_m$ et $\check{I}_m$ sont respectivement les brillances minimum et maximum du codeword (issues du second vecteur, $aux_m$, qui caractérise le codeword, défini ci-dessus).

L'algorithme Codebook construit un modèle d'arrière-plan pendant une période d'apprentissage effectuée sur $N$ images selon l'algorithme 2 rappelé ci-après.

**Algorithme 2** Construction du Codebook tel que décrit dans [47]

$L \leftarrow 0, C \leftarrow \emptyset$
**for** $t = 1$ à $N$ images **do**
(i) $x_t = (R, G, B), I \leftarrow \sqrt{R^2 + G^2 + B^2}$
(ii) Trouver le codeword $c_m$ dans $C = \{c_i | 1 \leq i \leq L\}$ correspondant à $x_t$ selon les deux conditions :
(a) $colorDist(x_t, v_m) \leq \epsilon_1$
(b) $brightnessDist(I, \langle \check{I}_m, \hat{I}_m\rangle) = vrai$
**if** (iii) $C = \emptyset$ ou bien aucune correspondance n'a été trouvée **then**
$L \leftarrow L + 1$. On crée un nouveau codeword $c_L$ en paramétrant :
$v_L \leftarrow (R, G, B)$
$aux_L \leftarrow \langle I, I, 1, t-1, t, t\rangle$.
**else**
(iv) Mettre à jour le codeword matché $c_m$ contenant $v_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ et $aux_m = [\check{I}_m, \hat{I}_m, f_m, \lambda_m, p, q]$ avec :
$v_m \leftarrow (\frac{f_m \bar{R}_m + R}{f_m + 1}, \frac{f_m \bar{G}_m + G}{f_m + 1}, \frac{f_m \bar{B}_m + B}{f_m + 1})$
$aux_m \leftarrow [min(I, \check{I}_m), max(I, \hat{I}_m), f_m + 1, max(\lambda_m, t - q_m), p_m, t]$
**end if**
**end for**
Pour chaque codeword $c_i, i \in \{1, ..., L\}$, régler $\lambda_i$ avec $\lambda_i \leftarrow max(\lambda_i, (N - q_i + p_i - 1))$

Dans cette phase d'apprentissage et pour la 1ère frame, le modèle $C$ est rempli avec un codeword par codebook (donc pour chaque pixel). Ensuite, pour les $N-1$ frames suivantes, une correspondance entre le pixel courant et un des codewords du codebook est recherchée. Si aucune correspondance n'est trouvée, un nouveau codeword est créé dans le codebook. A contrario, si une correspondance est trouvée, le codeword concerné est mis à jour.

Le matching entre le pixel courant et un des codewords du codebook est calculé en fonction de 3 paramètres :

1. $\epsilon$ est le seuil de distorsion de couleur acceptable. L'algorithme utilise deux valeurs différentes pendant l'apprentissage et pendant le traitement. Une distorsion de couleur faible signifie une proximité entre les couleurs du pixel et du codeword. Un seuil légèrement plus faible est généralement choisi pendant l'apprentissage afin de favoriser un apprentissage moins permissif sur la distorsion de couleur. Par exemple, les valeurs $\epsilon_1 = 10$ et $\epsilon_2 = 16$ peuvent être utilisées.

2. $\alpha$ est le facteur de brillance inférieur. Sa valeur est généralement comprise entre 0.4 et 0.7. Les auteurs indiquent avoir déterminé ces valeurs expérimentalement. Typiquement, une valeur proche de 0.7 permet de créer un intervalle plus large alors qu'une valeur proche de 0.4 permet un intervalle plus serré.

3. $\beta$ est le facteur de brillance supérieur. Sa valeur est généralement comprise entre 1.1 et 1.5. Ces valeurs sont également déterminées expérimentalement.

Une fois cet apprentissage effectué, le modèle d'arrière-plan $C$ est raffiné par la suppression des codewords correspondant potentiellement à des objets mobiles mouvant dans la scène pendant l'apprentissage ou du bruit. Pour cela, nous faisons référence au paramètre $\lambda$ : un codeword ayant une valeur de MNRL (*maximum negative run-length*) faible signifie qu'il a été matché plus ou moins régulièrement pendant l'apprentissage. Par exemple, un codeword situé dans la zone du ciel aura généralement une MNRL faible puisqu'il correspond à une couleur variant très peu, sera donc observé très régulièrement. A l'inverse, un codeword situé sur une fin de branche d'arbre aura une MNRL légèrement plus élevée car il sera observé avec une fréquence plus faible, mais cependant régulière. À l'extrême, un codeword correspondant à une silhouette qui s'est déplacée pendant l'apprentissage aura une MNRL très élevée puisqu'il sera observé tant que la silhouette est sur cette position, puis ne le sera plus par la suite.

Par conséquent, le raffinement du modèle $C$ est effectué en supprimant les codewords ayant une valeur de MNRL $\lambda$ trop élevée par expérience. On considère généralement qu'un codeword n'ayant pas été observé pendant au moins la moitié du nombre de frames d'apprentissage (donc la moitié du temps d'apprentissage) correspond à un objet mobile. Le modèle $M$ devient donc :

$$M = \{c_m | c_m \in C \text{ and } \lambda_m \leq T_M\}. \tag{1.47}$$

avec $T_M = \frac{N}{2}$.

Le modèle d'arrière-plan obtenu à l'issue de ce raffinement représente le fond de la séquence d'apprentissage. La deuxième phase consiste alors à détecter les objets mobiles et aboutir à une segmentation fond/forme pour chaque nouvelle frame. L'Algorithme 3 décrit, pour un pixel observé $x_t = (R, G, B)$, les étapes de la détection d'objets mobiles et la segmentation qui en découle.

**Algorithme 3** Détection des objets mobiles comme détaillée dans [47]

$x = (R, G, B), I \leftarrow \sqrt{R^2 + G^2 + B^2}$
Pour chaque codeword dans $M$, trouver le codeword $c_m$ correspondant à $x_t$ suivant les conditions :
(a) $colorDist(x, v_m) \leq \epsilon_2$
(b) $brightnessDist(I, \langle \check{I}_m, \hat{I}_m \rangle) = vrai$
Mettre à jour le codeword matché comme dans l'étape (iv) de l'Algorithme 2
**if** Il y a eu une correspondance **then**
    $x$ est marqué comme BG (background = fond)
**else**
    $x$ est marqué comme FG (foreground = forme)
**end if**

Nous pouvons noter que la procédure pour obtenir une segmentation avec l'algorithme Codebook consiste à rechercher un matching parmi les codewords inclus dans le codebook de chaque pixel. L'existence d'une correspondance entre un de ces codewords et le pixel observé dans la frame indique que ce pixel correspond bien à du fond. Au contraire, s'il n'en existe aucun alors ce pixel est marqué comme une forme, et donc un objet mobile.

Une analyse approfondie de cet algorithme nous permet de constater qu'en l'état, des changements à plus long terme s'opérant dans l'arrière-plan ne sont pas pris en compte. Par exemple, une voiture entrant sur un parking sera détectée comme foreground, puisqu'elle est en déplacement, et ce jusqu'à son stationnement. Cependant, même une fois stationnée, elle continuera d'être considérée comme foreground car l'algorithme n'intègre pas de mécanisme d'intégration progressive dans le modèle d'arrière-plan.

Devant cette constatation, Kim *et al.* [47] proposent néanmoins d'utiliser un second modèle d'arrière-plan en parallèle du premier, constituant un modèle *tampon*. Les codewords qu'il contient sont en tous points identiques à ceux du modèle d'arrière-plan effectif. Lorsqu'aucune correspondance n'a pu être déterminée, un codeword est créé dans le modèle tampon et le pixel est traité comme du foreground. Lors du traitement des frames suivantes, si aucune correspondance ne peut encore être trouvée dans le modèle d'arrière-plan effectif, une correspondance est alors recherchée dans le modèle tampon et le codeword est mis à jour si cette correspondance existe. Si ce codeword est resté dans le modèle tampon pendant une période minimale pré-définie, alors il est déplacé dans le modèle effectif.

Néanmoins, l'algorithme Codebook original (sans ce mécanisme de tampon) est robuste dans la plupart des cas, même s'il existe des situations dans lesquelles il échoue. Citons ici 2 exemples :

— Les changements d'illumination et de couleurs brusques dans la scène peuvent conduire à des segmentation très bruitées, les rendant inutilisables.

— Si une personne est habillée en couleurs sombres par temps froid, sa couleur peut être confondue avec celle du sol, auquel cas elle peut ne pas être détectée ;

# 3 Conclusion

Dans ce chapitre, nous avons introduit un certain nombre d'invariants colorimétriques et d'espaces de représentation couleur représentatifs qui pourraient permettre de simplifier les images réelles acquises en environnement intérieur/extérieur et présentant de fortes contraintes de luminosité, ombres, etc, en vie d'une segmentation fond/forme. A cela s'ajoute une description de différentes méthodes de segmentation fond/forme connues de la littérature avec une prise de position sur la méthode Codebook.

En conclusion, dans la suite de ce manuscrit nous avons fait le choix de considérer l'algorithme Codebook et de le coupler à l'utilisation d'invariants colorimétriques et espaces de représentation couleur de sorte à améliorer les résultats de segmentation dans des conditions d'éclairage difficiles (et ainsi répondre au premier élément de la liste précédente décrivant 2 exemples d'échecs de l'algorithme Codebook). Dans une second temps, nous étendrons les possibilités de cet algorithme en intégrant de nouvelles informations comme la profondeur afin de rendre la détection d'objets mobiles plus robuste (et ainsi répondre au second élément de la liste précédente). Ces premières propositions sont développées dans les deux chapitres suivants.

# Chapitre 2

# Segmentation fond/forme par la méthode Codebook avec invariance colorimétrique

Dans cette thèse, nous ambitionnons de développer une stratégie de segmentation d'objets mobiles opérable en environnement intérieur et extérieur et à tout moment de la journée, en s'affranchissant des changements d'éclairage, de luminosité et d'ombres qui viennent perturber la scène et de ce fait la détection des objets d'intérêt.

Pour mener à bien ces travaux, nous avons fait le choix d'utiliser une caméra Kinect (très utilisée dans la littérature ces dernières années, avec notamment le concept RGB-D que nous verrons dans le chapitre 3), mais aussi une caméra stéréoscopique Bumblebee 2 pour étendre nos travaux aux environnements extérieurs (puisque la caméra Kinect ne peut pas nous fournir une information de profondeur, utile dans nos travaux présentés dans le chapitre suivant, pour ce type d'environnement).

Avant de détailler la stratégie de segmentation d'objets mobiles mise en place nous allons présenter la méthode d'évaluation de nos résultats ainsi que les bases de données réelles que nous utiliserons également dans les deux chapitres suivants pour évaluer l'approche proposée.

# 1 Bases de données vidéos

La littérature offre un certain nombre de bases de données vidéos pour tester les algorithmes de détection et de suivi d'objets mobiles. Cependant, celles-ci ne présentent pas toutes les difficultés que nous souhaitons relever dans le cadre de cette thèse. En effet, dans le cadre de nos travaux nous cherchons à détecter des objets mobiles en tous lieux (extérieur, intérieur) et en tous temps (ensoleillé, nuageux, de jour, de nuit, etc).

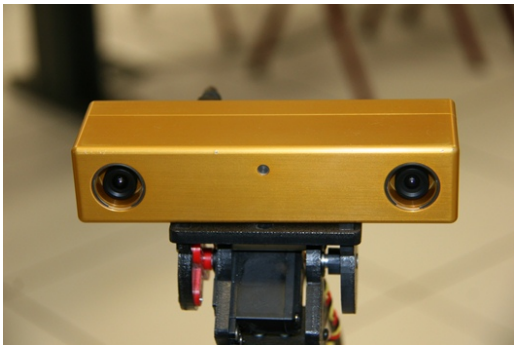
L'impact des changements d'illumination au cours d'une journée sur une scène (dûs à l'orientation du soleil selon l'horaire et aux conditions climatiques changeantes), qu'elle soit intérieure ou extérieure, est considérable à la fois en termes de changements de couleur et d'intensité de lumière globale. À cela s'ajoute la présence de lumières non naturelles dans la scène lorsque l'obscurité devient importante, ou encore d'ombres (qu'elles soient portées ou non), qui amènent à des conditions d'observation nettement altérées que les algorithmes de détection et de suivi d'objets mobiles doivent gérer.

Dans ce contexte, nous avons créé 3 bases de données vidéos intégrant ces différentes difficultés. Ces acquisitions, à la différence de celles qui sont mises à disposition dans la littérature, ont été effectuées dans des conditions de luminosité multiples à différents moments de la journée, pour certaines du matin très tôt au soir très tard. Celles-ci ont été acquises avec un capteur Microsoft Kinect (figure 2.1a) pour les scènes intérieures et avec un capteur stéréoscopique Bumblebee 2 (figure 2.1b) pour les scènes extérieures avec respectivement des images de tailles 640×480 et 1032×776. Les caractéristiques de ces deux caméras sont détaillées dans le tableau 2.1.

| | **Point Grey Bumblebee 2** | **Kinect** |
|---|---|---|
| Résolution | 1032×776 | 640×480 |
| Framerate | 20 FPS | 30 FPS |
| Megapixels | 0.8 MPX | 0.3 MPX |
| Capteur | Sony ICX204 (CCD) | CMOS |
| Focale | 3.8 mm, 65-deg HFOV | 57-deg HFOV, 43-deg FFOV |
| Ouverture | f/2.0 | inconnue |
| Interface | FireWire 1394a | USB |

TABLE 2.1 – Caractéristiques des caméras Point Grey Bumblebee 2 et Microsoft Kinect


(a) Caméra Microsoft Kinect



(b) Caméra stéréoscopique Point Grey Bumblebee 2

FIGURE 2.1 – Caméras utilisées

| Séquence | Caractéristiques |
|---|---|
| $L1$ | Matin, lumière éteinte, rideau levé |
| $L2$ | Matin, lumière allumée, rideau levé |
| $L3$ | Après-midi, lumière allumée, rideau levé |
| $L4$ | Après-midi, lumière éteinte, rideau levé |
| $L5$ | Soir, lumière éteinte, rideau baissé |
| $L6$ | Soir, lumière allumée, rideau baissé |

TABLE 2.2 – Caractéristiques de chaque séquence de la base "BUREAU"

Présentons maintenant les bases d'images qui ont été acquises au cours de ces travaux et utilisées pour tester la stratégie de détection d'objets mobiles mise en place. Elles sont au nombre de trois et caractérisent respectivement une scène intérieure et deux scènes extérieures. Chaque base d'images a été acquise à plusieurs moments d'une même journée (à intervalles réguliers ou non). De sorte à disposer de conditions de luminosité différentes, la base ainsi générée est découpée en séquences $L_x$ correspondant chacune à des conditions de luminosité spécifiques.

## 1.1 Scène intérieure : "BUREAU"

Cette base d'images acquise à l'aide d'une caméra Microsoft Kinect, a été générée à partir de six séquences acquises au cours d'une journée avec des conditions d'illumination différentes (cf. figure 2.2). Le tableau 2.2 représente les différentes conditions d'acquisition de cette base.

Le scénario qui a été joué pour pouvoir tester in fine la stratégie de détection d'objets mobiles est répété à l'identique pour chaque séquence de cette base "BUREAU" : une personne entre dans la pièce, se dirige vers le fond de la pièce, puis revient sur ses pas, contourne le bureau, s'assied un instant, et enfin se lève pour sortir de la pièce.

## 1.2 Scène extérieure : "JARDIN"

Cette deuxième base d'images a été acquise en extérieur au cours d'une même journée d'été (figure 2.3) à l'aide de la caméra stéréoscopique Bumblebee 2 puisque la caméra Microsoft Kinect ne peut pas fonctionner en extérieur lorsque l'on souhaite (comme cela sera le cas par la suite) utiliser l'information de profondeur. Quatre séquences permettent de générer cette deuxième base, chacune réalisée à un horaire différent de la journée. Les détails de ces séquences sont donnés dans le tableau 2.3. Cette base présente par conséquent plusieurs difficultés que nous pouvons illustrer :

| Séquence | Caractéristiques |
|---|---|
| $L1$ | Matin (10h), ciel clair |
| $L2$ | Début d'après-midi, ciel ensoleillé et nuageux et beaucoup de vent |
| $L3$ | Après-midi, ciel plus nuageux que L2, moins d'occultations de soleil |
| $L4$ | Après-midi, ciel nuageux, plus longue durée, une personne supplémentaire |

TABLE 2.3 – Caractéristiques de chaque séquence de la base "JARDIN"

| Séquence | Heure | Prédominance de la couleur de la scène |
|---|---|---|
| $L1$ | 7h00 | Gris foncé |
| $L2$ | 9h00 | Vert |
| $L3$ | 12h30 | Jaune/Blanc |
| $L4$ | 15h00 | Gris |
| $L5$ | 17h30 | Bleu/Violet |

TABLE 2.4 – Description des séquences de la base "RUE".

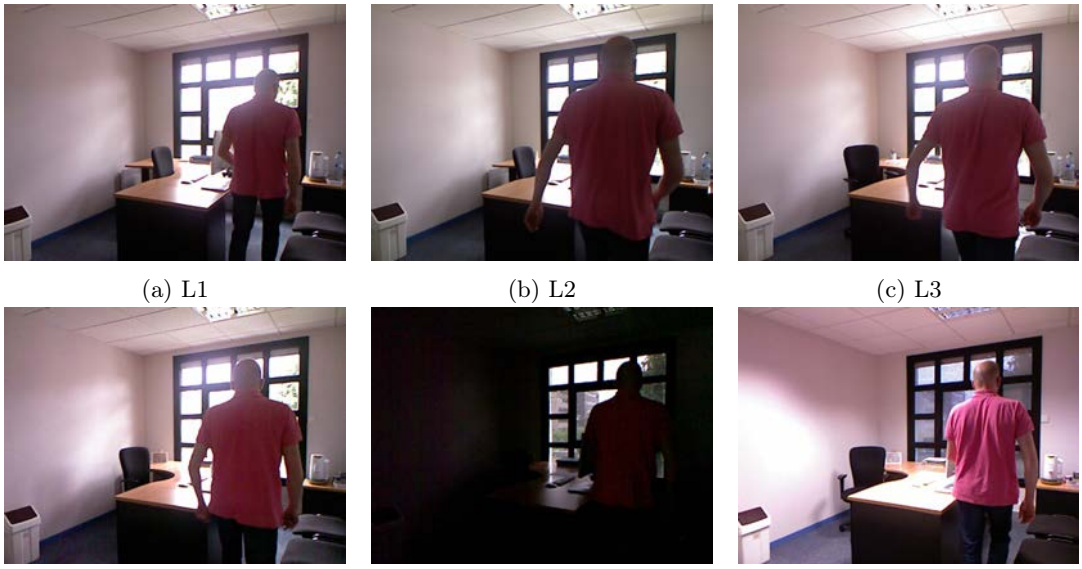
des occultations (des objets mobiles passent derrière des arbres), des objets mobiles avec une petite taille (car le champ de vision est important), un nombre important de régions d'arrière-plan dynamiques (feuillage des arbres très mouvant à cause du vent), une colorimétrie majoritairement verte (gazon, buissons et feuilles) sensible aux forts changements d'illumination survenant au cours de la journée voire même parfois au milieu d'une même séquence (nuages qui passent devant le soleil).

Le scénario joué lors de ces acquisitions est similaire pour les 3 premières séquences : deux personnes parties depuis des directions opposées se rencontrent au centre de l'image et se dirigent ensuite ensemble vers la caméra. La quatrième séquence est légèrement différente puisque même si le scénario joué est identique, la longueur de celui-ci est plus importante et nous avons ajouté une personne supplémentaire qui se déplace à travers la scène, occasionnant ainsi des occultations supplémentaires.

## 1.3 Scène extérieure : "RUE"

La dernière base d'images a également été acquise en milieu extérieur avec la caméra Bumblebee 2 mais à la différence de la précédente, pendant une journée complète d'hiver. Les horaires d'acquisition vont du matin jusqu'au soir : 7h, 9h, 12h30, 15h et 17h30 comme le décrit le tableau 2.4 de sorte à disposer d'importants changements dans l'illumination de la scène comme illustré dans la figure 2.4. En effet, nous constatons que les couleurs passent du gris foncé au vert le matin, puis au jaune/blanc à la mi-journée, pour finir au bleu/violet le soir.

Le scénario joué pour cette base d'images consiste en plusieurs déplacements de personnes à différentes distances de la caméra (en prenant soin d'avoir des croisements, immobilisations, accélérations, rapprochements et éloignements de la caméra). Nous notons également divers véhicules qui circulent sur la chaussée, pouvant générer des ombres portées.



(a) L1 (b) L2 (c) L3

(d) L4 (e) L5 (f) L6

FIGURE 2.2 – Images des différentes séquences de la base “BUREAU”.

(a) L1 (b) L2

(c) L3 (d) L4

FIGURE 2.3 – Images des différentes séquences de la base “JARDIN”.

(a) L1 (b) L2 (c) L3

(d) L4 (e) L5

FIGURE 2.4 – Images des différentes séquences de la base "RUE".

## 2 Méthodes d'évaluation

Il existe dans la littérature de nombreuses méthodes d'évaluation des résultats de segmentations issues de méthodes de détection fond/forme. Elles peuvent majoritairement être découpées en deux catégories. Certains auteurs choisissent des techniques d'évaluation basées sur la *fonction d'efficacité du récepteur*, beaucoup plus fréquemment désignée sous le terme de courbe ROC de l'anglais Receiver Operating Characteristic (ou *courbe sensibilité/spécificité* en français).

D'autres auteurs préfèrent utiliser la courbe precision/rappel dont les deux informations sont contenues dans un score appelé F-Mesure (FM). Cependant, la plupart des mesures utilisées font appel au nombre de vrais et faux-positifs et vrais et faux négatifs contenus dans les segmentations. Pour les calculer, des segmentations de référence sont réalisées manuellement en labélisant les régions appartenant au fond et aux objets mobiles et une comparaison de ces valeurs avec les valeurs contenues dans les segmentations automatiques est effectuée.

Enfin, la littérature offre également des méthodes d'évaluation lorsqu'il n'est pas possible de disposer d'une segmentation de référence. A titre d'exemple, Chalidabhongse *et al* [11] proposent une mesure appelée PDR (Perturbation Detection Rate) pour mesurer les performances des algorithmes de détection fond/forme ne nécessitant pas l'utilisation de segmentation de référence. Elle est présentée comme une alternative à la courbe ROC et permet de mesurer la sensibilité de détection d'un algorithme. L'idée de base consiste à mesurer à quelle distance deux distributions doivent se trouver l'une de l'autre pour permettre un certain taux de détection. Autrement dit, on détermine à partir de quel moment un pixel que l'on fait progressivement varier d'une couleur du fond vers une autre couleur, est considéré comme objet mobile. Concrètement, cette méthode consiste à utiliser un nombre défini $N$ de frames d'apprentissage ne contenant pas d'objet mobile. Ces frames sont utilisées dans une ou plusieurs méthodes de détection à tester. Puis, plusieurs itérations vont être effectuées sur ces $N$ frames, en modifiant à chaque itération un certain nombre $M$ de pixels en faisant varier leur couleur (chaque composante, uniformément dans une direction aléatoire) selon une grandeur $\delta$ qui augmente à chaque itération. On obtient alors un taux de dé-

tection pour chaque valeur de $\delta$ que l'on peut afficher graphiquement. La première de ces itérations (lorsque $\delta = 0$) constitue le résultat obtenu en traitant les frames d'apprentissage sans aucune altération : le taux de faux-négatifs obtenu doit logiquement être très faible (entre 0.001 et 0.01). D'après [11], la plupart des méthodes de détection ont un taux de détection correct lorsque $\delta$ est grand. Aussi, il est préférable de s'intéresser aux résultats obtenus avec un contraste de couleur plus faible ($\delta < 40$) où les différences de détection sont plus grandes.

Revenons sur une des méthodes les plus connues introduite précédemment et faisant appel à la notion de précision et de rappel, à savoir la F-Mesure. Cette F-Mesure (ou F1-score en anglais) décrite dans l'équation suivante, permet de déterminer un pourcentage estimant la performance générale d'une segmentation.

$$F\_Mesure = \frac{2.(Recall.Precision)}{Recall + Precision} \tag{2.1}$$

avec

$$Precision = \frac{TP}{TP+FP}, \tag{2.2}$$

$$Recall = \frac{TP}{TP+FN}, \tag{2.3}$$

et où $TP$ représente le nombre de vrais-positifs, $TN$ le nombre de vrais-négatifs, $FP$ le nombre de faux-positifs et $FN$ le nombre de faux-négatifs.

L'utilisation de la F-Mesure nécessite une connaissance de la scène et donc la création de segmentations de référence. Ceci n'est pas un problème en soi, car de nombreuses séquences mises à disposition dans la littérature fournissent les segmentations de référence correspondantes.

Dans les sous-sections suivantes, nous présenterons les résultats de segmentation exprimés avec une F-Mesure calculée pour chaque classe de segmentation recherchée (background et foreground). En effet, notre objectif final étant d'améliorer la qualité des segmentations directement issues de l'algorithme de détection utilisé, nous souhaitons connaître l'impact de notre proposition à la fois sur les objets mobiles mais aussi sur le fond. Or, d'après l'équation 2.1, la F-Mesure ne tient pas compte de la bonne classification des pixels appartenant au fond (vrais-négatifs). C'est une mesure permettant de déterminer l'efficacité d'un classifieur binaire (comme l'est un algorithme de détection des objets mobiles), mais qui ne considère que l'existence de "bons résultats" et de "mauvais résultats" au sein d'un même ensemble. Dans notre cas, selon la classe considérée (fond ou forme), les "bons résultats" peuvent être soit les vrais positifs, soit les vrais négatifs. Ceci fait naître le besoin de considérer pour la Précision et le Rappel, deux formules de calcul supplémentaires, une pour chacune de ces deux classes, à savoir le fond (bg) et les objets mobiles (fg) que nous décrivons dans les équations données ci-après.

$$Precision_{fg} = \frac{TP}{TP+FP} \quad Precision_{bg} = \frac{TN}{TN+FN} \tag{2.4}$$

$$Recall_{fg} = \frac{TP}{TP+FN} \quad Recall_{bg} = \frac{TN}{TN+FP} \tag{2.5}$$

Afin de faciliter la lecture des résultats de détection d'objets mobiles, nous allons ci-après expliquer la notation employée et le protocole de test utilisé. Considérons un traitement effectué sur une des trois base d'images comportant $S$ séquences différentes, chacune acquise à un horaire différent. Ces séquences sont nommées $L1$ à $L_S$, selon l'ordre chronologique d'acquisition. Le protocole de

test est alors le suivant : un apprentissage est effectué sur un nombre $N$ de frames d'apprentissage d'une séquence donnée. Dans un second temps, les autres séquences de la base sont testées dans l'ordre chronologique. Nous réitérons cette procédure autant de fois qu'il existe de séquences dans la base considérée de sorte à observer l'influence de l'apprentissage sur une séquence plutôt que sur une autre.

A titre d'exemple, pour une base comportant 4 séquences différentes, 4 cas peuvent être testés. L'apprentissage peut être effectué sur $L1$ : dans ce cas, les bases testées sont $P1$, $P2$, $P3$ et enfin $P4$. De même, l'apprentissage peut être effectué sur $L2$ : dans ce cas, les bases testées sont dans l'ordre $P2$, $P3$, $P4$ puis enfin $P1$. La procédure est réitérée jusqu'à ce qu'il n'y ait plus de nouvelle séquence à apprendre.

Pour généraliser, si l'apprentissage est effectué sur une séquence $Lx$, les séquences testées sont nommées $Pabcd$, ce qui dans notre exemple donne les quatre tests possibles suivants, que nous notons $MOY\_L_x$ car les résultats qui seront présentés par la suite indiquent des valeurs moyennes calculées sur l'ensemble du test ainsi défini :

— $MOY\_L1$ correspondant à $L1\_P1234$

— $MOY\_L2$ correspondant à $L2\_P2341$

— $MOY\_L3$ correspondant à $L3\_P3412$

— $MOY\_L4$ correspondant à $L4\_P4123$

Dans les figures et annexes suivants, les résultats sont déterminés pour chaque séquence et une moyenne est ensuite calculée. Chaque colonne des tableaux de résultats présentés dans la suite de ce mémoire, ainsi que ceux donnés en annexe, correspond à un invariant colorimétrique. La colonne *rgb* correspond à un traitement effectué *sans* utilisation d'invariant colorimétrique : il s'agit donc du traitement avec les images brutes. Les correspondances entre les intitulés des tableaux sont affichées dans le tableau 1.1 du chapitre précédent. Précisons toutefois que les valeurs des F-Mesures moyennes indiquées dans tous les tableaux sont pondérées avec le nombre de frames composant chacune des séquences de la base considérée, c'est-à-dire selon l'équation 2.6 :

$$\mu_{total} = \sum_{i=1}^{S} \frac{nbframes(i)}{\sum_{k=1}^{S} nbframes(k)} \mu(i) \tag{2.6}$$

où $\mu_{total}$ est la F-Mesure moyenne sur l'ensemble des $S$ séquences testées, $nbframes(i)$ étant le nombre d'images composant la séquence $i$ et enfin $\mu(i)$ la F-Mesure moyenne de la séquence $i$.

# 3 Segmentation d'images par Codebook et invariance colorimétrique

Nous avons choisi d'utiliser l'algorithme Codebook pour tester la stratégie de détection d'objets mobiles. Cet algorithme se déroule en deux étapes : apprentissage et traitement. Pendant l'apprentissage, l'algorithme construit un "codebook" pour chaque pixel, contenant plusieurs "codewords" représentatifs des différentes couleurs et intensités observées pour ce pixel au cours des $N$ frames d'apprentissage. Puis, le traitement des nouvelles frames recherche dans le codebook de chaque pixel l'existence d'une correspondance avec le pixel observé. Si une correspondance existe, alors le pixel est considéré comme appartenant au fond, sinon il appartient aux formes.

Cet algorithme utilise trois paramètres $\alpha$, $\beta$ et $\epsilon$ qui permettent de rendre l'algorithme plus ou moins permissif dans son calcul de similarité entre un codeword présent dans le codebook d'un pixel, et le codeword un pixel observé. Cette similarité est déterminée par une valeur de distorsion de couleur qui ne doit pas dépasser un certain seuil ($\epsilon$) et une valeur de distorsion d'intensité qui doit être comprise dans un intervalle déterminé par les intensités minimale et maximale du codeword testé, elles-mêmes calculées à l'aide respectivement des facteurs $\alpha$ et $\beta$.

## 3.1 Détection d'objets mobiles sur la base "BUREAU"

Nous avons décrit précédemment les différentes étapes du fonctionnement de l'algorithme Codebook. Nous avons pu constater qu'il consiste, pour un pixel donné de l'image testée, en la recherche d'un codeword intégré au codebook. Cette correspondance se base sur deux types de distorsions (couleur et intensité). On considère qu'il y a correspondance si la distorsion de couleur est inférieure ou égale à un paramètre $\epsilon$ et que la distorsion d'intensité est comprise dans un intervalle déterminé par les intensités minimum et maximum du codeword (calculées respectivement à l'aide des paramètres $\alpha$ et $\beta$).

Kim *et al.* [47] indiquent dans leurs travaux que les valeurs de ces trois paramètres sont déterminées empiriquement et qu'ils retiennent les valeurs suivantes (l'algorithme utilise deux valeurs différentes pour $\epsilon$, une pour la phase d'apprentissage et l'autre pour la phase de test, respectivement $\epsilon_1$ et $\epsilon_2$) :

— $\alpha = 0.75$

— $\beta = 1.3$

— $\epsilon_1 = 10$

— $\epsilon_2 = 16$

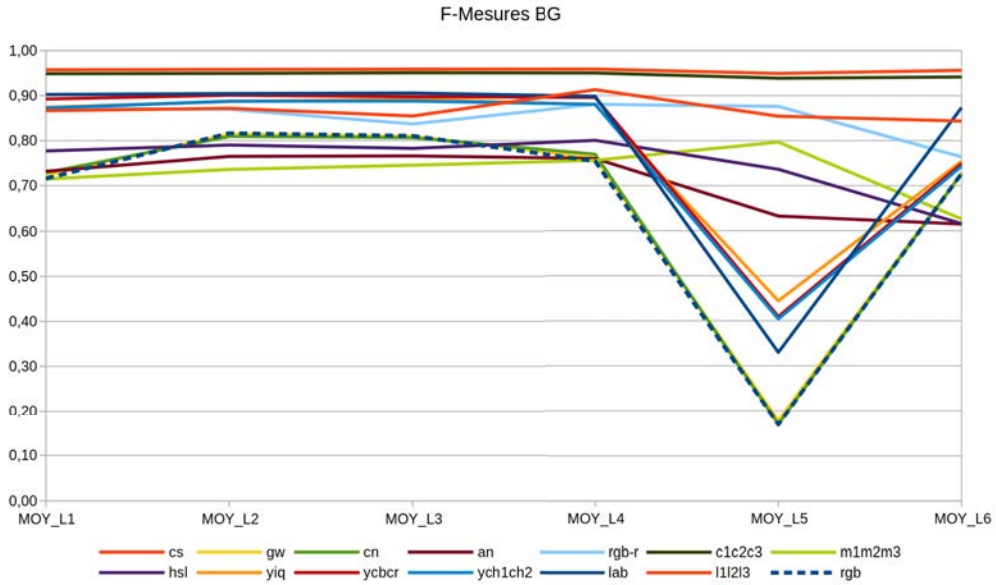
A ce jour, il n'existe pas de méthode permettant de les calculer de façon optimale, ceci à cause de trop nombreux facteurs externes : la dynamique de la scène, la caméra utilisée, l'angle de vue, la météo, etc.

Le paramètre $\epsilon$ est un seuil de détection compris entre 0 et 255, 0 correspondant à la distorsion de couleur obtenue entre deux couleurs identiques, et 255 la distorsion de couleur obtenue entre deux couleurs diamétralement opposées (en l'occurrence, entre le noir $(0, 0, 0)$ et le blanc $(255, 255, 255)$). Une valeur $\epsilon$ faible rendra l'algorithme plus sensible aux différences de couleur pour un même pixel, et aura donc pour effet d'intégrer plus de codewords dans le modèle afin de représenter toutes ces différences ; au contraire, une valeur importante de $\epsilon$ rendra le modèle moins sensible aux différences de couleur.

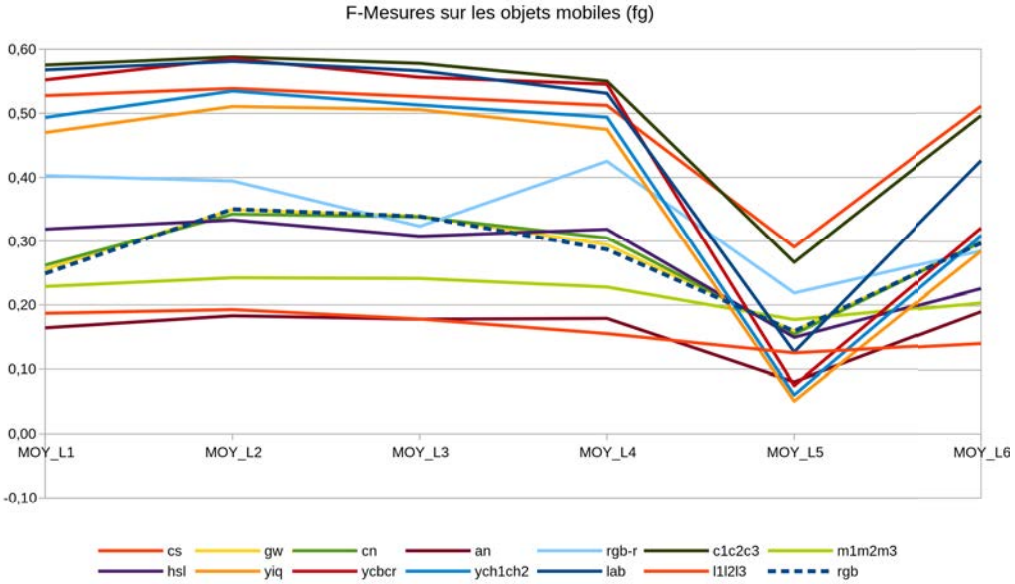
Les paramètres $\alpha$ et $\beta$ permettent de limiter la distorsion d'intensité entre deux bornes $I_{low}$ (intensité minimale) et $I_{high}$ (intensité maximale). Kim *et al.* [47] indiquent que $\alpha$ prend typiquement une valeur comprise entre 0.4 et 0.7. Plus $\alpha$ est faible, plus $I_{low}$ est petit. Une valeur $I_{low}$ faible augmente la sensibilité de l'algorithme aux intensités faibles, c'est-à-dire aux ombres pouvant apparaître pendant le traitement. Inversement, une valeur plus haute permet d'augmenter la sensibilité de l'algorithme pour les pixels plus sombres. Toujours selon Kim *et al.* [47], le paramètre $\beta$ est censé prendre typiquement une valeur comprise entre 1.1 et 1.5. Plus $\beta$ est fort, plus $I_{high}$ est grand. Une valeur $I_{high}$ haute diminue la sensibilité de l'algorithme pour les pixels dont l'intensité serait plus importante, comme dans le cas de sur-illuminations. Inversement, une valeur $I_{high}$ plus faible permet d'éviter la correspondance d'un pixel d'intensité plus importante que celle du codeword testé.

Au vu de ces explications, nous avons choisi d'utiliser les m mes valeurs que celles utilis es par Fernandez-Sanchez et al. [22], puisque les conditions de test sont similaires, savoir la d tection d'objets mobiles en environnement int rieur avec l'utilisation d'une cam ra Kinect.

Les tests ont alors t r alis s sur la base "BUREAU" contenant 6 s quences di rentes acquises plusieurs moments de la journ e. Comme indiqu pr c demment, six tests ont t e ectu s. Les tableaux 2.5a et 2.5b listent les moyennes des r sultats (F-Mesures) obtenus dans chacun de ces tests, et pour chacune des s quences test es dans l'ordre. Le d tail de ces r sultats pour chaque test, est pr sent dans les di rents tableaux 3.10, 3.11, 3.12, 3.13, 3.14 et 3.15 pr sent s en annexe.




(a) BG




(b) FG

FIGURE 2.5 – F-Mesures moyennes du fond (bg) et des objets mobiles (fg) calcul es avec des apprentissages sur les s quences L1, L2, L3, L4, L5 et L6, et di rents invariants pour la base "BUREAU".

| BG | rgb | cs | gw | cn | an | rgb-r | c1c2c3 |
|---|---|---|---|---|---|---|---|
| MOY_L1 | 0,72 | 0,96 | 0,72 | 0,73 | 0,73 | 0,87 | 0,95 |
| MOY_L2 | 0,82 | 0,96 | 0,81 | 0,81 | 0,76 | 0,87 | 0,95 |
| MOY_L3 | 0,81 | 0,96 | 0,81 | 0,81 | 0,77 | 0,84 | 0,95 |
| MOY_L4 | 0,75 | 0,96 | 0,76 | 0,77 | 0,76 | 0,88 | 0,95 |
| MOY_L5 | 0,17 | 0,95 | 0,18 | 0,17 | 0,63 | 0,88 | 0,94 |
| MOY_L6 | 0,73 | 0,96 | 0,73 | 0,73 | 0,61 | 0,76 | 0,94 |
| +/- | 0,00 | 0,29 | 0,00 | 0,00 | 0,05 | 0,18 | 0,28 |
| | | | | | | | |
| BG | m1m2m3 | hsl | yiq | ycbcr | ych1ch2 | lab | l1l2l3 |
| MOY_L1 | 0,71 | 0,78 | 0,87 | 0,89 | 0,87 | 0,90 | 0,87 |
| MOY_L2 | 0,74 | 0,79 | 0,89 | 0,90 | 0,89 | 0,90 | 0,87 |
| MOY_L3 | 0,75 | 0,78 | 0,89 | 0,90 | 0,89 | 0,91 | 0,85 |
| MOY_L4 | 0,76 | 0,80 | 0,88 | 0,90 | 0,88 | 0,90 | 0,91 |
| MOY_L5 | 0,80 | 0,74 | 0,44 | 0,41 | 0,40 | 0,33 | 0,85 |
| MOY_L6 | 0,63 | 0,62 | 0,75 | 0,75 | 0,74 | 0,87 | 0,84 |
| +/- | 0,06 | 0,08 | 0,12 | 0,13 | 0,11 | 0,14 | 0,20 |

(a) Fond (bg)

| FG | rgb | cs | gw | cn | an | rgb-r | c1c2c3 |
|---|---|---|---|---|---|---|---|
| MOY_L1 | 0,25 | 0,53 | 0,26 | 0,26 | 0,16 | 0,40 | 0,58 |
| MOY_L2 | 0,35 | 0,54 | 0,35 | 0,34 | 0,18 | 0,39 | 0,59 |
| MOY_L3 | 0,34 | 0,53 | 0,34 | 0,34 | 0,18 | 0,32 | 0,58 |
| MOY_L4 | 0,29 | 0,51 | 0,29 | 0,31 | 0,18 | 0,43 | 0,55 |
| MOY_L5 | 0,16 | 0,29 | 0,16 | 0,15 | 0,08 | 0,22 | 0,27 |
| MOY_L6 | 0,30 | 0,51 | 0,30 | 0,30 | 0,19 | 0,29 | 0,50 |
| +/- | 0,00 | 0,20 | 0,00 | 0,00 | -0,12 | 0,06 | 0,23 |
| | | | | | | | |
| FG | m1m2m3 | hsl | yiq | ycbcr | ych1ch2 | lab | l1l2l3 |
| MOY_L1 | 0,23 | 0,32 | 0,47 | 0,55 | 0,49 | 0,57 | 0,19 |
| MOY_L2 | 0,24 | 0,33 | 0,51 | 0,59 | 0,53 | 0,58 | 0,19 |
| MOY_L3 | 0,24 | 0,31 | 0,51 | 0,56 | 0,51 | 0,57 | 0,18 |
| MOY_L4 | 0,23 | 0,32 | 0,47 | 0,55 | 0,49 | 0,53 | 0,16 |
| MOY_L5 | 0,18 | 0,15 | 0,05 | 0,07 | 0,06 | 0,13 | 0,13 |
| MOY_L6 | 0,20 | 0,23 | 0,28 | 0,32 | 0,31 | 0,43 | 0,14 |
| +/- | -0,06 | 0,00 | 0,10 | 0,16 | 0,12 | 0,19 | -0,12 |

(b) Objets mobiles (fg)

TABLE 2.5 – F-Mesures moyennes pour le fond (bg) et les objets mobiles (fg) obtenues pour chaque invariant colorimétrique sur un test avec un apprentissage sur la séquence L1, L2, L3, L4, L5 ou L6 pour la base "BUREAU".

Cependant, pour une meilleure visualisation, nous concaténons les résultats sur la figure 2.5. Nous pouvons constater que la méthode Codebook est relativement efficace puisque sans avoir à utiliser d'invariant colorimétrique, les F-Mesures moyennes du fond (bg) sont en moyenne supérieures à 0.66, ce qui dénote des segmentations correctes des régions appartenant au fond, même si les résultats issus du test $L5$ diminuent cette moyenne de façon importante. La moyenne des F-Mesures moyennes des objets mobiles (fg) est proche de 0.28 sur cette base, ce qui reste une valeur correcte pour les cas d'utilisation difficiles considérés ici. L'exception se situe là encore sur la séquence $L5$ qui présente un nombre de fausses détections trop important du fait de l'inadaptation du modèle par rapport aux séquences suivantes.

Cependant, on observe que certains invariants colorimétriques permettent d'augmenter ces F-

Mesures, la fois pour le fond (bg) et pour les objets mobiles (fg). Les trois invariants colorim - triques qui permettent les meilleures am liorations sont, dans l'ordre croissant, la Comprehensive Normalization (cs) qui permet une augmentation de la F-Mesure moyenne du fond (bg) de 0.29 par rapport la F-Mesure moyenne sans invariant (colonne *rgb*), ainsi qu'une augmentation de 0.20 pour la F-Mesure moyenne des objets mobiles (fg). Vient ensuite l'invariant colorim trique c1c2c3 avec des augmentations de F-Mesures de 0.28 pour le fond (bg) et 0.23 (fg) pour les objets mobiles. Enfin, l'espace L*a*b* apporte galement une am lioration notable pour les F-Mesures du fond et des objets mobiles avec respectivement un gain de 0.14 et de 0.19.

La figure 2.6 illustre quelques r sultats de segmentation obtenus pour chaque s quence apr s un apprentissage e ectu sur des images de la s quence $L2$. Nous pouvons constater que les invariants colorim triques pr sent s permettent e ectivement une am lioration l g re mais sensible de la qualit de segmentation, notamment dans le fond. N anmoins, le cas de la s quence $L5$, beaucoup plus sombre, reste tr s complexe et peu satisfaisant m me si l'invariant Chromaticity Space semble avoir pour e et de "trouer" les r gions du fond, ce qui pourrait permettre en post-traitement de filtrer les r gions restantes, tout en pr servant les r gions appartenant aux objets mobiles qui, dans cet exemple, restent relativement concentr es.


FIGURE 2.6 – R sultats de segmentation sur une image de chaque s quence ($L1$ $L6$), apr s un apprentissage sur la s quence $L2$ (base int rieure "BUREAU") et selon l'invariant colorim trique utilis .

Ces r sultats nous permettent de tirer deux observations :

— L'emploi de certains invariants colorim triques ou espaces couleur comme par exemple Comprehensive Normalization, c1c2c3 ou encore L*a*b*, a un v ritable impact b n fique sur

la qualité des segmentations des objets mobiles lorsque l'apprentissage est effectué sur une séquence à l'aspect colorimétrique très différent des séquences testées après elle ;

— Pour ces tests, la séquence d'apprentissage $L2$ obtient les meilleures segmentations. Nous supposons donc qu'il existe dans chaque base d'images, une séquence dont l'usage en tant que séquence d'apprentissage permet d'obtenir de meilleurs résultats.

## 3.2 Détermination des paramètres optimaux de l'algorithme

Comme introduit dans le chapitre précédent, nous allons utiliser l'algorithme du Codebook que nous allons coupler à l'utilisation d'invariants colorimétriques pour réaliser la détection des objets mobiles. Pour cela, il est important d'optimiser les paramètres $\alpha$, $\beta$ et $\epsilon$.

Devant la difficulté à déterminer ces valeurs optimales, plusieurs ensembles de paramètres listés dans le tableau 2.6 ont été testés sur nos bases acquises à l'extérieur, puisque les intervalles de valeurs typiques proposés par Kim *et al.* [47] pour des bases acquises en intérieur se sont révélés trop limitants pour nos bases d'images, qui contiennent d'importants changements de couleur et d'intensité.

TABLE 2.6 – Liste des paramètres testés pour les bases extérieures.

| | | | | | | |
|---|---|---|---|---|---|---|
| $\alpha$ | 0 | 0.1 | 0.2 | 0.5 | 0.7 | |
| $\beta$ | 1.0 | 1.5 | 3.0 | | | |
| $\epsilon$ | 0 | 50 | 100 | 150 | 200 | 250 |

Dans tous les résultats présentés ci-après, les F-Mesures relatives aux objets mobiles ($fg$) calculées avec le paramètre $\epsilon = 0$ sont nulles et celles du fond ($bg$) sont très proches de 1.0 (100%). Ceci dénote des segmentations dont la quasi-totalité des pixels du fond sont correctement segmentés (vrais-négatifs), alors que tous les pixels des objets mobiles ne sont pas détectés comme tels (faux-négatifs).

Le fonctionnement attendu avec un paramètre $\epsilon$ faible est l'augmentation du nombre de codewords dans le codebook d'un pixel donné. En conséquence, devant la difficulté (voire l'impossibilité) de l'algorithme à trouver une correspondance entre les codewords présents dans le codebook et chaque nouveau pixel observé, un nouveau codeword est créé, incrémentant leur nombre à chaque frame d'apprentissage. A la fin de l'apprentissage, la taille du codebook d'un grand nombre de pixels se retrouve supérieure à $N/2$ ($N$ étant le nombre de frames d'apprentissage). Chacun des codewords présents dans ces codebooks a donc une valeur $\lambda$ importante car ils n'ont pas été matchés assez fréquemment. L'étape de raffinement se déroulant à la fin de l'apprentissage cause la suppression de tous les codewords dont la valeur $\lambda$ excède $N/2$ : les codebooks d'un grand nombre de pixels ne contiennent alors plus aucun codeword et sont donc vides. C'est ce qui se produit avec le paramètre $\epsilon = 0$ pour un très grand nombre de pixels.

Ce type de situation peut survenir pour un petit nombre de pixels dans un fonctionnement normal. Plusieurs comportements peuvent alors être envisagés si le codebook d'un pixel est vide après l'apprentissage :

— Dans la segmentation, considérer ces pixels comme des objets mobiles (faux-positifs) ;

— Dans la segmentation, considérer ces pixels comme du fond (faux-négatifs) ;

— Lors de l'opération de raffinement, garder le codeword ayant la plus petite valeur $\lambda$ afin d'avoir toujours au moins 1 codeword dans chaque codebook.

Le fonctionnement choisi ici est de considérer ces pixels comme du fond afin d'éviter de biaiser les résultats. Cependant, comme la détection des objets mobiles est fortement dégradée avec ce paramètre, nous ne le retiendrons pas comme paramètre optimal.

## 3.3 Détection d'objets mobiles sur la base "JARDIN"

Comme indiqué précédemment, les paramètres $\alpha$, $\beta$ et $\epsilon$ décrits dans [47] ne sont optimisés que pour la base intérieure. De ce fait, nous avons choisi de les ré-optimiser sur la base extérieure en déterminant les valeurs (parmi la liste présentée dans le tableau 1.1) qui permettent d'obtenir les meilleurs résultats, à la fois sur la séquence d'apprentissage et sur les séquences de test suivantes qui ont été acquises dans des conditions d'illumination différentes. Plus précisément, l'effet recherché est une minimisation du nombre de faux-positifs dans les régions du fond, tout en préservant au maximum la segmentation des objets mobiles.

D'un point de vue pratique, les tests effectués pour la recherche des paramètres optimaux ont été réalisés sur deux séquences d'apprentissage : $L1$ (tests sur $P1234$) et $L2$ (tests sur $P2341$). Les résultats de ces tests sont présentés dans les tableaux 2.7a à 2.8b. Nous informons le lecteur que ces mêmes tests ont également été réalisés avec le Codebook RGB-D détaillé dans le chapitre 3, de sorte que nous puissions vérifier qu'il s'agisse de paramètres optimaux pour les deux méthodes.

| $\epsilon$ | | | 0 | **50** | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | bg | | | | | |
| MOY_L1 | **0,1** | **3** | 0,98 | **0,96** | 0,97 | 0,97 | 0,97 | 0,97 |
| | **0,2** | 1 | 0,98 | 0,82 | 0,82 | 0,82 | 0,82 | 0,82 |
| | | 1,5 | 0,98 | 0,91 | 0,91 | 0,90 | 0,90 | 0,90 |
| | | 2 | 0,98 | 0,90 | 0,89 | 0,89 | 0,89 | 0,89 |
| | | 3 | 0,98 | 0,90 | 0,89 | 0,89 | 0,89 | 0,89 |
| | 0,5 | 3 | 0,98 | 0,71 | 0,72 | 0,72 | 0,72 | 0,72 |
| | 0,7 | 3 | 0,98 | 0,55 | 0,56 | 0,55 | 0,55 | 0,55 |

(a) Fond (bg)

| $\epsilon$ | | | 0 | **50** | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | fg | | | | | |
| MOY_L1 | 0,1 | 3 | 0,00 | 0,11 | 0,05 | 0,03 | 0,03 | 0,03 |
| | **0,2** | 1 | 0,00 | 0,04 | 0,04 | 0,04 | 0,04 | 0,04 |
| | | **1,5** | 0,00 | **0,10** | 0,08 | 0,07 | 0,07 | 0,07 |
| | | 2 | 0,00 | 0,10 | 0,07 | 0,06 | 0,06 | 0,06 |
| | | 3 | 0,00 | 0,10 | 0,07 | 0,06 | 0,06 | 0,06 |
| | 0,5 | 3 | 0,00 | 0,10 | 0,10 | 0,10 | 0,10 | 0,10 |
| | 0,7 | 3 | 0,00 | 0,08 | 0,08 | 0,08 | 0,08 | 0,08 |

(b) Objets mobiles (fg)

TABLE 2.7 – Moyenne des F-Mesures moyennes calculées sur le fond (bg) et les objets mobiles (fg) avec un apprentissage sur la séquence $L1$ de la base "JARDIN", pour différentes valeurs des paramètres $\alpha, \beta$ et $\epsilon$.

Au vu des résultats, nous constatons que les paramètres $\alpha = 0.1, \beta = 3.0$ permettent de réduire de façon très importante le bruit dans le fond. Les faux-positifs généralement présents dans les feuillages des arbres sont très peu nombreux, et la F-Mesure calculée sur le fond est supérieure à 0.95, quelle que soit la valeur du paramètre $\epsilon$. En revanche, la F-Mesure des objets mobiles reste faible : elle ne dépasse pas 0.11. En effet dans ce cas, les objets mobiles ne sont détectés que lorsqu'ils sont suffisamment proches de la caméra.

| $\epsilon$ | | | 0 | **50** | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | bg | | | | | |
| MOY_L2 | 0,1 | 3 | 0,98 | 0,97 | 0,97 | 0,97 | 0,97 | 0,97 |
| | **0,2** | 1 | 0,98 | 0,74 | 0,74 | 0,74 | 0,74 | 0,74 |
| | | **1,5** | 0,98 | **0,92** | 0,93 | 0,93 | 0,93 | 0,93 |
| | | 2 | 0,98 | 0,92 | 0,93 | 0,93 | 0,93 | 0,93 |
| | | 3 | 0,98 | 0,92 | 0,93 | 0,93 | 0,93 | 0,93 |
| | 0,5 | 3 | 0,98 | 0,68 | 0,68 | 0,68 | 0,68 | 0,68 |
| | 0,7 | 3 | 0,98 | 0,54 | 0,54 | 0,54 | 0,54 | 0,54 |

(a) Fond (bg)

| $\epsilon$ | | | 0 | **50** | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | fg | | | | | |
| MOY_L2 | 0,1 | 3 | 0,00 | 0,08 | 0,04 | 0,04 | 0,04 | 0,04 |
| | **0,2** | 1 | 0,00 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 |
| | | **1,5** | 0,00 | **0,08** | 0,08 | 0,07 | 0,07 | 0,07 |
| | | 2 | 0,00 | 0,08 | 0,07 | 0,07 | 0,07 | 0,07 |
| | | 3 | 0,00 | 0,08 | 0,07 | 0,07 | 0,07 | 0,07 |
| | 0,5 | 3 | 0,00 | 0,07 | 0,07 | 0,07 | 0,07 | 0,07 |
| | 0,7 | 3 | 0,00 | 0,07 | 0,07 | 0,07 | 0,07 | 0,07 |

(b) Objets mobiles (fg)

TABLE 2.8 – Moyenne des F-Mesures moyennes calculées sur le fond (bg) et les objets mobiles (fg) avec un apprentissage sur la séquence $L2$ de la base "JARDIN", pour différentes valeurs de paramètres $\alpha, \beta$ et $\epsilon$.

Avec les paramètres $\alpha = 0.2, \beta = 1.0$, la F-Mesure du fond (bg) est moins élevée : environ 0.82 pour un apprentissage sur $L1$ et 0.74 pour un apprentissage sur L2, quelle que soit la valeur du paramètre $\epsilon$, tout comme la F-Mesure des objets mobiles (fg) qui est également constante quelle que soit la séquence testée. Ces résultats reflètent une augmentation du nombre de faux-positifs (dans le fond), et une difficulté encore plus grande à segmenter les objets mobiles.

Considérons maintenant le couple de paramètres $\alpha = 0.2, \beta = 1.5$. L'augmentation du paramètre $\beta$ doit avoir pour effet de limiter les faux-positifs dans certaines régions du fond, soumises à d'importantes augmentations d'intensité lumineuse. Concrètement, les résultats impactés par cette modification voient leur F-Mesure, à la fois du fond (bg) et des objets mobiles (fg) augmentées. En effet, nous constatons que les F-Mesures moyennes observées pour les objets mobiles sont pratiquement multipliées par 2, passant de valeurs situées autour de 0.04 à des valeurs comprises entre 0.07 et 0.10. Pour les régions du fond, les valeurs des F-Mesures relevées sont également plus importantes, et se situent autour de 0.9, ce qui dénote un faible nombre de faux-négatifs et de faux-positifs.

En ce qui concerne le couple de paramètres $\alpha = 0.2, \beta = 2.0$, les différences sont difficilement notables comparé aux paramétrages décrits dans le paragraphe précédent. Néanmoins, les F-Mesures du fond et des objets mobiles apparaissent très faiblement inférieures (de moins de 0.01) à celles issues du paramétrage précédent. Cette observation est d'ailleurs vérifiée dans le second test où l'apprentissage est effectué sur la base L2. Même si cette différence est très faible, cela nous suffit pour exclure ce couple de paramètres au profit du précédent. Nous observons les mêmes remarques pour le couple de paramètres $\alpha = 0.2, \beta = 1.5$.

Comme avec le couple de paramètres précédent, nous n'observons pas de différence sur les valeurs des F-Mesure pour les régions du fond et des objets mobiles lorsque nous utilisons le couple

de paramètres $\alpha = 0.2, \beta = 3.0$.

Concernant le couple de paramètres $\alpha = 0.5, \beta = 3.0$, les F-Mesures obtenues pour le fond (bg) révèlent une dégradation plus importante qu'avec le couple de paramètres précédent puisqu'elles sont inférieures de 0.17 à 0.19. Cette dégradation est révélatrice d'une augmentation du nombre de faux-positifs. En revanche, dans le cas où l'apprentissage est effectué sur L1, nous remarquons une augmentation des F-Mesures moyennes des objets mobiles pouvant aller jusqu'à un gain de 0.03. Cette amélioration s'explique par l'augmentation du nombre de faux-positifs observée précédemment, qui induit également une diminution du nombre de faux-négatifs (certains pixels sur les silhouettes des objets mobiles se retrouvent partiellement ou totalement remplis). Néanmoins, cette augmentation ne se reproduit pas dans le cas où l'apprentissage est effectué sur L2, alors que les F-Mesures moyennes du fond sont encore plus dégradées, passant de 0.92 (en moyenne avec le paramètre $\epsilon = 50$) à 0.68, soit une dégradation de 0.24. En conclusion, ces résultats ne nous permettent pas de considérer ce couple de paramètres comme optimal.

Enfin, le couple de paramètres $\alpha = 0.7, \beta = 3.0$ obtient les plus mauvaises F-Mesures sur le fond parmi toutes les solutions testées, ce qui fait que nous ne le retiendrons pas quelle que soit la F-Mesure obtenue sur les objets mobiles.

En conclusion des différents paramétrages testés, nous retenons que des valeurs $\alpha$ et $\beta$ trop importantes provoquent une augmentation des mauvaises détections (faux-positifs), dégradant de façon importante la segmentation du fond (bg), alors que des valeurs moyennes de $\alpha$ et $\beta$ permettent d'obtenir des résultats corrects pour la segmentation des objets mobiles.

De même, sur l'ensemble des tests réalisés nous avons observé que le paramètre $\epsilon$ n'est pas le paramètre limitant. Néanmoins, nous attirons l'attention du lecteur sur le fait que la valeur de ce paramètre est différente pendant l'apprentissage ($\epsilon_1$) ou le test ($\epsilon_2$). Nous fixerons $\epsilon_1$ comme dans les travaux de Fernandez-Sanchez *et al.* [22] à $\frac{\epsilon_2}{1.6}$.

Nous rappelons que ces observations ont été effectuées également avec un apprentissage sur la base L1, mais aussi sur la base $L2$ dont l'aspect colorimétrique apparaît différent de la première et que, sauf contre-indication, les mêmes observations ont été effectuées.

En conclusion, les paramètres jugés optimaux et que nous retiendrons pour l'algorithme Codebook lorsque nous considérerons la base "JARDIN" sont :

— $\alpha = 0.1$

— $\beta = 3.0$

— $\epsilon_1 = 31.25$

— $\epsilon_2 = 50$

Maintenant que nous avons défini les paramètres optimaux de l'algorithme, intéressons-nous aux résultats de la stratégie globale. Les résultats de segmentation sont illustrés sur la figure 2.8.

Nous pouvons alors constater que les résultats de segmentation obtenus sans l'utilisation d'invariants colorimétriques sont globalement satisfaisants pour le fond, mais insuffisants pour les objets mobiles. Même si l'utilisation d'invariant colorimétrique permet dans certains cas d'augmenter ces résultats, la F-Mesure ne dépasse pas 0.28 (pour la séquence $L4$). Ceci peut sûrement s'expliquer par le fait que les silhouettes présentes dans les séquences de cette base sont éloignées de la caméra (figure 2.8), et donc que leur taille en termes de nombre de pixels est très petite. Aussi, un "trou" dans la détection d'une de ces silhouettes, aussi petit soit-il, peut causer une chute de la F-Mesure des objets mobiles.

Cependant, nous constatons clairement dans le tableau 2.9 que certains invariants colorimétriques permettent d'apporter une amélioration dans la qualité des segmentations. Nous pouvons citer Chromaticity Space (*cs*) qui permet une augmentation des F-Mesures du fond comprise entre 0.0 et 0.01 selon la séquence utilisée pour l'apprentissage, et une augmentation comprise entre 0.14 et 0.17 pour les objets mobiles (fg). Nous remarquons que plusieurs autres invariants colorimétriques ont un effet positif sur la qualité des segmentations, toutes séquences confondues. Il s'agit de c1c2c3 et de l'espace couleur L*a*b* que nous avons tous deux déjà identifiés comme très intéressants sur la base intérieure précédente. À titre d'exemple c1c2c3 permet d'améliorer la segmentation des objets mobiles (fg) entre 0.06 et 0.09 et L*a*b* entre 0.04 et 0.07.

| BG | rgb | cs | gw | cn | an | rgb-r | c1c2c3 |
|---|---|---|---|---|---|---|---|
| MOY_L1 | 0,96 | 0,97 | 0,92 | 0,92 | 0,72 | 0,93 | 0,92 |
| MOY_L2 | 0,97 | 0,98 | 0,92 | 0,92 | 0,87 | 0,94 | 0,95 |
| MOY_L3 | 0,97 | 0,97 | 0,96 | 0,96 | 0,85 | 0,96 | 0,95 |
| MOY_L4 | 0,97 | 0,98 | 0,95 | 0,95 | 0,84 | 0,96 | 0,95 |
| +/- | 0,00 | 0,01 | -0,03 | -0,03 | -0,15 | -0,02 | -0,03 |
| | | | | | | | |
| BG | m1m2m3 | hsl | yiq | ycbcr | ych1ch2 | lab | l1l2l3 |
| MOY_L1 | 0,80 | 0,54 | 0,91 | 0,91 | 0,90 | 0,92 | 0,64 |
| MOY_L2 | 0,77 | 0,83 | 0,93 | 0,92 | 0,93 | 0,93 | 0,82 |
| MOY_L3 | 0,80 | 0,66 | 0,93 | 0,94 | 0,93 | 0,95 | 0,65 |
| MOY_L4 | 0,83 | 0,76 | 0,92 | 0,92 | 0,92 | 0,95 | 0,69 |
| +/- | -0,17 | -0,27 | -0,05 | -0,05 | -0,05 | -0,03 | -0,27 |

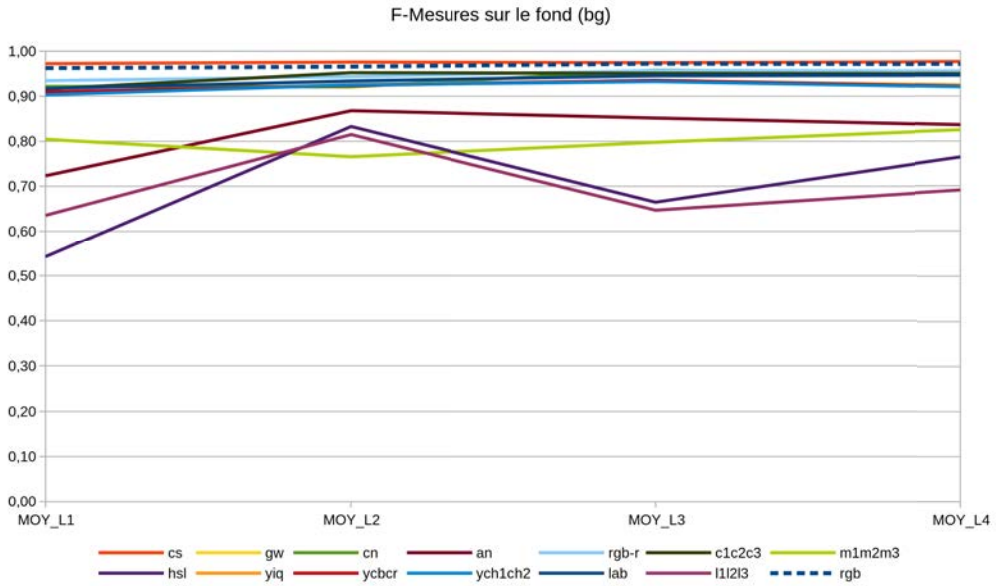
(a) Fond (bg)

| FG | rgb | cs | gw | cn | an | rgb-r | c1c2c3 |
|---|---|---|---|---|---|---|---|
| MOY_L1 | 0,11 | 0,26 | 0,11 | 0,11 | 0,08 | 0,11 | 0,19 |
| MOY_L2 | 0,08 | 0,25 | 0,08 | 0,08 | 0,08 | 0,10 | 0,17 |
| MOY_L3 | 0,10 | 0,24 | 0,12 | 0,12 | 0,07 | 0,13 | 0,17 |
| MOY_L4 | 0,13 | 0,28 | 0,13 | 0,13 | 0,07 | 0,14 | 0,20 |
| +/- | 0,00 | 0,15 | 0,00 | 0,00 | -0,03 | 0,01 | 0,07 |
| | | | | | | | |
| FG | m1m2m3 | hsl | yiq | ycbcr | ych1ch2 | lab | l1l2l3 |
| MOY_L1 | 0,08 | 0,10 | 0,16 | 0,16 | 0,15 | 0,18 | 0,07 |
| MOY_L2 | 0,07 | 0,10 | 0,10 | 0,10 | 0,11 | 0,14 | 0,05 |
| MOY_L3 | 0,07 | 0,07 | 0,11 | 0,12 | 0,11 | 0,16 | 0,06 |
| MOY_L4 | 0,07 | 0,08 | 0,10 | 0,11 | 0,10 | 0,18 | 0,06 |
| +/- | -0,04 | -0,02 | 0,01 | 0,01 | 0,01 | 0,06 | -0,05 |

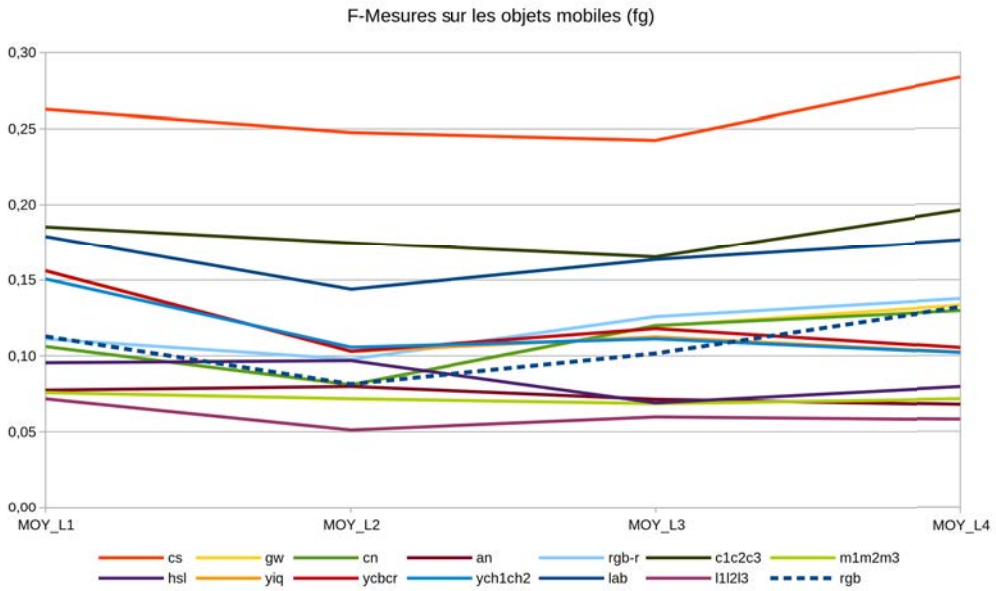
(b) Objets mobiles (fg)

TABLE 2.9 – F-Mesures moyennes pour le fond (bg) et les objets mobiles (fg) obtenues pour chaque invariant colorimétrique sur un test avec un apprentissage sur la séquence L1, L2, L3 ou L4 (base "JARDIN").

Pour ce qui est de la détection du fond, nous pouvons constater à l'aide de la figure 2.8 que les invariants colorimétriques ne permettent pas toujours d'obtenir une image contenant peu de bruit dans les régions de l'arrière-plan de la scène. Ceci est notamment le cas dans les régions difficiles comme le feuillage des arbres. Il est d'ailleurs complexe d'obtenir une parfaite détection dans cet environnement avec une méthode basée uniquement sur la couleur. L'image étant composée majoritairement de vert, cette couleur est fortement sensible aux changements de luminosité mais nous verrons dans le chapitre suivant que l'ajout d'une information de profondeur peut dans certains cas permettre d'améliorer encore ces résultats. L'élimination du bruit résiduel (non éliminé par




(a) BG

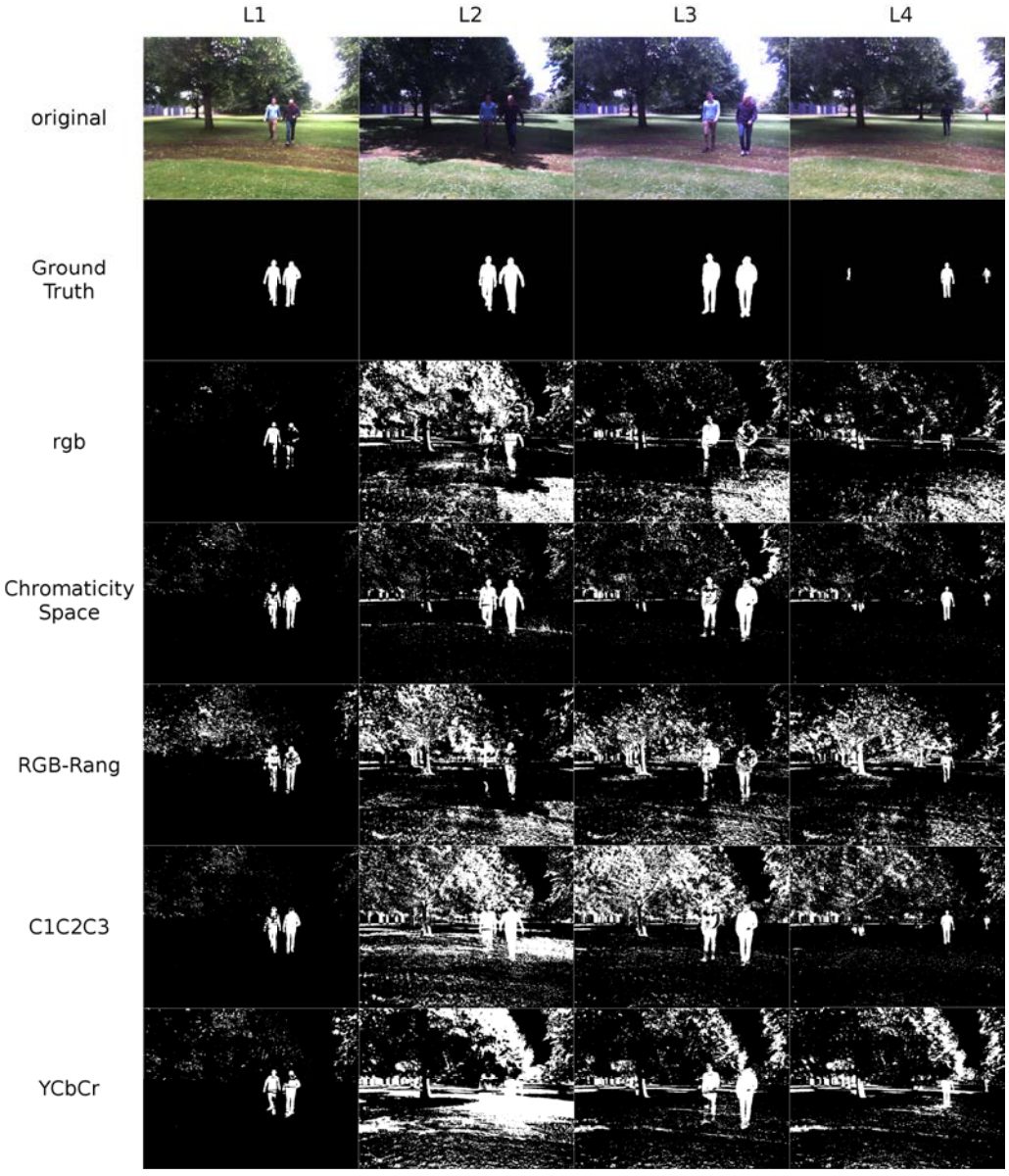



(b) FG

FIGURE 2.7 – F-Mesures moyennes du fond (bg) et des objets mobiles (fg) calcul es avec des apprentissages sur les s quences L1, L2, L3 et L4 et di rents invariants pour la base "JARDIN".

l'utilisation d'invariants colorim triques) pourra ventuellement tre trait par un filtrage ad quat, l'aide par exemple d'op rateurs morphologiques.

FIGURE 2.8 – R sultats de segmentation sur une image de chaque s quence (L1 L4), apr s un apprentissage sur la s quence $L1$ de la base ext rieure ("JARDIN") et selon l'invariant colorim trique utilis .



## 3.4 Détection d'objets mobiles sur la base "RUE"

Comme cela a été réalisé pour les deux bases précédentes, nous allons ci-après présenter les résultats de détection d'objets mobiles sur une dernière base que nous avons appelée "RUE" et qui a la particularité d'avoir été acquise tout au long d'une journée d'hiver et donc avec des changements de luminosité très importants.

Comme précédemment, nous avons cherché à optimiser les paramètres de l'algorithme Codebook en fonction de cette base. Les résultats de cette recherche sont présentés sur les tableaux 2.10a et 2.10b pour un apprentissage sur la séquence $L1$ de cette base d'images, et sur les tableaux 2.11a et 2.11b pour un apprentissage sur la séquence L3.

| $\epsilon$ | | | 0 | 50 | **100** | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | bg | | | | | |
| MOY_L1 | **0,1** | **3** | 0,99 | 0,93 | **0,94** | 0,94 | 0,94 | 0,94 |
| | 0,2 | 1 | 0,99 | 0,73 | 0,73 | 0,73 | 0,73 | 0,73 |
| | | 1,5 | 0,99 | 0,79 | 0,81 | 0,81 | 0,81 | 0,81 |
| | | 2 | 0,99 | 0,80 | 0,82 | 0,82 | 0,82 | 0,82 |
| | | 3 | 0,99 | 0,82 | 0,83 | 0,83 | 0,83 | 0,83 |
| | 0,5 | 3 | 0,99 | 0,42 | 0,44 | 0,44 | 0,44 | 0,44 |
| | 0,7 | 3 | 0,99 | 0,42 | 0,44 | 0,44 | 0,44 | 0,44 |

(a) Fond (bg)

| $\epsilon$ | | | 0 | 50 | **100** | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | fg | | | | | |
| MOY_L1 | **0,1** | **3** | 0,00 | 0,11 | **0,13** | 0,14 | 0,14 | 0,14 |
| | 0,2 | 1 | 0,00 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 |
| | | 1,5 | 0,00 | 0,12 | 0,14 | 0,15 | 0,15 | 0,15 |
| | | 2 | 0,00 | 0,11 | 0,13 | 0,14 | 0,14 | 0,14 |
| | | 3 | 0,00 | 0,11 | 0,13 | 0,14 | 0,14 | 0,14 |
| | 0,5 | 3 | 0,00 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 |
| | 0,7 | 3 | 0,00 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 |

(b) Objets mobiles (fg)

TABLE 2.10 – Moyenne des F-Mesures moyennes calculées sur le fond (bg) et les objets mobiles (fg) avec un apprentissage sur la séquence L1 de la base "RUE", pour différentes valeurs des paramètres $\alpha, \beta$ et $\epsilon$.

Considérons le couple de paramètres $\alpha = 0.1, \beta = 3.0$. Nous observons des F-Mesures pour le fond (bg) relativement élevées puisqu'elles sont supérieures à 0.90, quelle que soit la valeur de $\epsilon$ et la séquence d'apprentissage utilisée, ce qui implique une bonne segmentation des régions appartenant au fond. A partir d'une valeur $\epsilon = 100$, nous observons des valeurs des F-Mesures pour les objets mobiles (fg) comprises entre 0.13 et 0.14 pour la séquence d'apprentissage $L1$, et entre 0.26 et 0.27 pour un apprentissage sur la séquence $L3$. Nous garderons donc le $\epsilon$ médian qui correspond à une valeur de 100, évitant ainsi un paramètre potentiellement trop élevé ou trop faible.

En ce qui concerne le couple de paramètres $\alpha = 0.2, \beta = 1.0$, les tests semblent ici offrir en moyenne de moins bons résultats pour la segmentation du fond comme des objets mobiles. En effet, nous notons en moyenne une chute de F-Mesures de près de 0.20 pour le fond, et de près de 0.08 pour les objets mobiles. Ce résultat est explicable par la forte différence colorimétrique et d'intensité lumineuse qui caractérise les cinq séquences de cette base d'images. Le fait d'utiliser des valeurs de paramètres $\alpha$ et $\beta$ telles que l'intervalle entre ces deux valeurs est faible, a pour effet d'empêcher l'algorithme à accepter de grandes différences colorimétriques pour un même pixel

| $\epsilon$ | | | 0 | 50 | **100** | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | bg | | | | | |
| MOY_L3 | **0,1** | **3** | 0,99 | 0,97 | **0,97** | 0,97 | 0,97 | 0,97 |
| | 0,2 | 1 | 0,99 | 0,88 | 0,89 | 0,89 | 0,89 | 0,89 |
| | | 1,5 | 0,99 | 0,83 | 0,83 | 0,83 | 0,83 | 0,83 |
| | | 2 | 0,99 | 0,83 | 0,83 | 0,83 | 0,83 | 0,83 |
| | | 3 | 0,99 | 0,83 | 0,83 | 0,82 | 0,82 | 0,82 |
| | 0,5 | 3 | 0,99 | 0,24 | 0,24 | 0,24 | 0,24 | 0,24 |
| | 0,7 | 3 | 0,99 | 0,24 | 0,24 | 0,24 | 0,24 | 0,24 |

(a) Fond (bg)

| $\epsilon$ | | | 0 | 50 | **100** | 150 | 200 | 250 |
|---|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | fg | | | | | |
| MOY_L3 | **0,1** | **3** | 0,00 | 0,29 | **0,27** | 0,26 | 0,26 | 0,26 |
| | 0,2 | 1 | 0,00 | 0,18 | 0,19 | 0,19 | 0,19 | 0,19 |
| | | 1,5 | 0,00 | 0,24 | 0,23 | 0,22 | 0,22 | 0,22 |
| | | 2 | 0,00 | 0,23 | 0,21 | 0,21 | 0,20 | 0,20 |
| | | 3 | 0,00 | 0,22 | 0,21 | 0,20 | 0,20 | 0,20 |
| | 0,5 | 3 | 0,00 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 |
| | 0,7 | 3 | 0,00 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 |

(b) Objets mobiles (fg)

TABLE 2.11 – Moyenne des F-Mesures moyennes calculées sur le fond (bg) et les objets mobiles (fg) avec un apprentissage sur la séquence L3 de la base "RUE", pour différentes valeurs des paramètres $\alpha, \beta$ et $\epsilon$.

entre les différentes séquences, ce qui cause une importante augmentation des faux positifs dans le fond. L'effet sur les objets mobiles est moindre puisque ces régions sont composées d'un nombre inférieur de pixels. Nous pouvons légitimement nous attendre, pour le couple de paramètres ($\alpha = 0.2, \beta = 1.5$) testé ensuite, à une baisse similaire des F-Mesures, dont les valeurs correspondront à une augmentation de la taille de l'intervalle $[\alpha; \beta]$.

Comparativement au couple de paramètres précédent, le test avec le couple $\alpha = 0.2, \beta = 1.5$ dénote un plus grand intervalle $[\alpha; \beta]$ (la borne supérieure de l'intervalle est ici plus élevée). Cela implique une augmentation des F-Mesures dans trois cas sur quatre par rapport au couple de paramètres précédent, mais cette augmentation permet néanmoins d'obtenir des F-Mesures inférieures à celles obtenues avec le premier couple ($\alpha = 0.1, \beta = 3.0$).

Si l'on considère les couples de paramètres $\alpha = 0.2, \beta = 2.0$ et $\alpha = 0.2, \beta = 3.0$, les F-Mesures obtenues sont quasiment identiques à celles obtenues avec l'ensemble de paramètres précédent. Il semble que nous tendions vers une limite causée par le paramètre $\alpha$, puisque c'est ce paramètre qui n'a pas changé. Les diminutions sur les F-Mesures sont certes faibles (environ 0.01 en moyenne), mais là encore, c'est le premier couple de paramètres testé ($\alpha = 0.1, \beta = 3.0$) qui globalement offre de meilleurs résultats.

Pour les deux derniers couples testés, à savoir $\alpha = 0.5, \beta = 3.0$ et $\alpha = 0.7, \beta = 3.0$, l'hypothèse ci-dessus semble se confirmer. Les valeurs des F-Mesures du fond chutent ici radicalement de plus de 0.50, ce qui implique une augmentation spectaculaire du nombre de faux-positifs dans les régions du fond. Les valeurs des F-Mesures pour les objets mobiles chutent quant à elles entre 0.06 et 0.12 par rapport au couple $\alpha = 0.1, \beta = 3.0$. Les segmentations obtenues avec ces paramètres présentent un très fort taux de faux positifs (ce qui se traduit généralement par un aspect majoritairement blanc dans les segmentations). Il y a fort à penser que l'augmentation du paramètre $\alpha$ à 0.5 a

réduit davantage la marge minimale d'intensité lumineuse acceptable des pixels, ce qui empêche totalement l'algorithme de trouver une correspondance avec ces pixels dans le modèle d'arrière-plan.

**Paramètres choisis** En conclusion et au vu des résultats présentés dans les tableaux 2.10a, 2.10b, 2.11a et 2.11b, le couple $\alpha = 0.1, \beta = 3.0$ permet d'obtenir les meilleures F-Mesures moyennes. Comme précédemment, le paramètre $\epsilon$ n'est pas limitant. En effet, sa valeur impacte très peu les F-Mesures (pour des valeurs $\alpha, \beta$ identiques). Notons par exemple que pour des valeurs de $\epsilon = 50$ ou $\epsilon = 100$ (les deux valeurs de paramètres ayant montré les meilleurs résultats), les différences de F-Mesures sont très faibles (entre 0 et 0.02) et ce, que l'on considère le fond ou les objets mobiles et quelle que soit la base d'apprentissage utilisée. Mais, devant arrêter un choix entre ces deux valeurs de paramètre, nous avons choisi de garder la valeur $\epsilon$ ayant offert le plus de F-Mesures maximales (fond et objets mobiles confondus).

Nous avons donc retenu les paramètres suivants comme paramètres optimaux de l'algorithme Codebook pour le traitement de la base d'images "RUE" :

— $\alpha = 0.1$

— $\beta = 3.0$

— $\epsilon_1 = 62.5$

— $\epsilon_2 = 100$

Intéressons-nous maintenant aux résultats de la détection d'objets mobiles sur cette base complexe en montrant l'apport de l'utilisation d'invariants colorimétriques. Les résultats de segmentation sont, comme pour toutes les autres bases, concaténés sur la figure 2.10, mais le lecteur pourra s'il le souhaite se référer aux résultats détaillés présentés en annexe (tableaux 3.20, 3.21, 3.22, 3.23, 3.24).

Au vu de ces résultats, nous pouvons aisément indiquer que l'utilisation d'un invariant colorimétrique permet au minimum dans 12 cas sur 14 (avec un maximum à 13) d'améliorer la segmentation du fond (bg) et dans 6 cas sur 14 (avec un maximum à 11) celle des objets mobiles (fg).

Sur cette base d'images, les F-Mesures affichées dans les figures 2.9a et 2.9b ne montrent pas d'importantes améliorations sur les objets mobiles. En revanche, celles-ci existent principalement dans les régions de l'arrière-plan. Selon la séquence employée pour l'apprentissage, ces améliorations ne sont pas les mêmes. Par exemple, l'invariant colorimétrique RGB-Rang (rgb-r) permet une amélioration de la F-Mesure moyenne du fond (bg) et des objets mobiles (fg) de respectivement 0.04 et 0.06 lorsque la séquence d'apprentissage est $L1$, et respectivement de 0.05 et 0.14 lorsque la séquence d'apprentissage est $L2$. Également, plusieurs invariants colorimétriques permettent une amélioration de la F-Mesure du fond, mais dégradent celle des objets mobiles dans tous les cas, comme Chromaticity Space (cs) et c1c2c3, ainsi que HSL, YIQ, YCbCr, YCh1Ch2 et L*a*b*. Nous notons donc les bonnes performances de l'invariant colorimétrique RGB-Rang (rgb-r), Greyworld (gw) et Comprehensive Normalization (cn) sur cette base vidéo.

Concernant la séquence d'apprentissage la plus intéressante, il semble que la séquence $L3$ soit celle qui permet les F-Mesures les plus hautes à la fois pour le fond (bg) et pour les objets mobiles (fg). C'est déjà le cas lorsqu'aucun invariant colorimétrique n'est utilisé, mais l'utilisation d'un des trois invariants colorimétriques cités ci-dessus améliore par ailleurs ce résultat. L'apparition de ces pics de F-Mesures pour la séquence $L3$ est notamment causée par la présence d'un plus grand nombre d'objets mobiles à détecter (dû à l'horaire de grande affluence pour cette séquence) et bien

sûr à la luminosité plus importante occasionnant beaucoup moins de bruit que les autres séquences d'apprentissage.

| BG | rgb | cs | gw | cn | an | rgb-r | c1c2c3 |
|---|---|---|---|---|---|---|---|
| MOY_L1 | 0,94 | 0,99 | 0,95 | 0,95 | 0,85 | 0,98 | 0,98 |
| MOY_L2 | 0,93 | 0,99 | 0,94 | 0,94 | 0,86 | 0,98 | 0,99 |
| MOY_L3 | 0,97 | 0,99 | 0,98 | 0,98 | 0,89 | 0,98 | 0,99 |
| MOY_L4 | 0,83 | 0,99 | 0,84 | 0,83 | 0,87 | 0,97 | 0,99 |
| MOY_L5 | 0,83 | 0,99 | 0,91 | 0,91 | 0,86 | 0,98 | 0,99 |
| +/- | 0,00 | 0,09 | 0,02 | 0,02 | -0,04 | 0,07 | 0,09 |
| | | | | | | | |
| BG | m1m2m3 | hsl | yiq | ycbcr | ych1ch2 | lab | l1l2l3 |
| MOY_L1 | 0,76 | 0,96 | 0,97 | 0,98 | 0,97 | 0,99 | 0,76 |
| MOY_L2 | 0,82 | 0,94 | 0,98 | 0,98 | 0,97 | 0,99 | 0,41 |
| MOY_L3 | 0,78 | 0,91 | 0,99 | 0,99 | 0,99 | 0,99 | 0,75 |
| MOY_L4 | 0,83 | 0,93 | 0,97 | 0,97 | 0,97 | 0,99 | 0,83 |
| MOY_L5 | 0,81 | 0,95 | 0,98 | 0,98 | 0,97 | 0,99 | 0,73 |
| +/- | -0,10 | 0,04 | 0,07 | 0,08 | 0,07 | 0,09 | -0,21 |

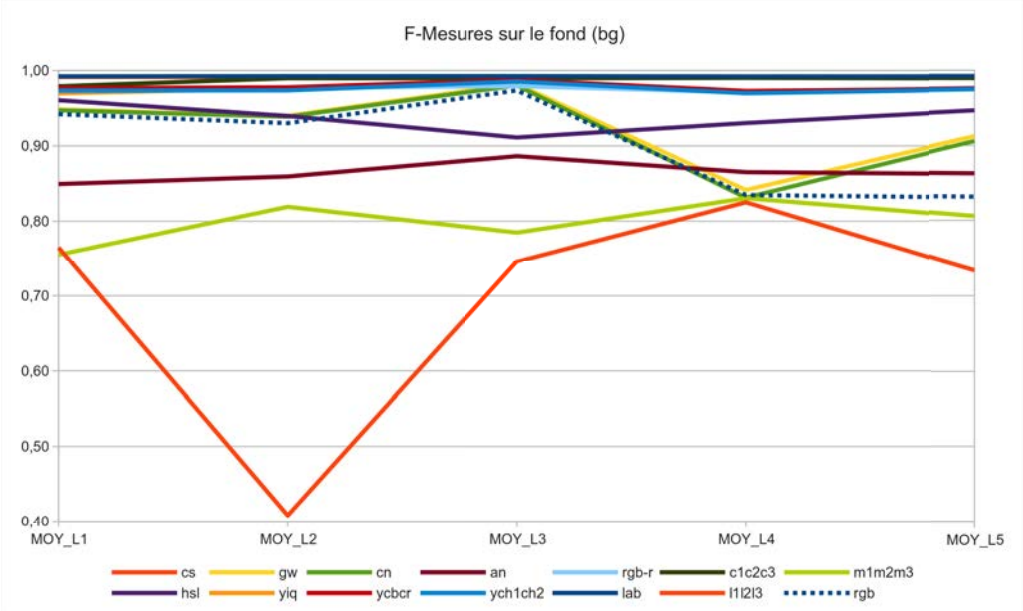
(a) Fond (bg)

| FG | rgb | cs | gw | cn | an | rgb-r | c1c2c3 |
|---|---|---|---|---|---|---|---|
| MOY_L1 | 0,13 | 0,04 | 0,15 | 0,15 | 0,10 | 0,19 | 0,06 |
| MOY_L2 | 0,13 | 0,05 | 0,15 | 0,15 | 0,13 | 0,27 | 0,10 |
| MOY_L3 | 0,27 | 0,05 | 0,30 | 0,29 | 0,16 | 0,27 | 0,09 |
| MOY_L4 | 0,21 | 0,04 | 0,17 | 0,16 | 0,15 | 0,24 | 0,08 |
| MOY_L5 | 0,13 | 0,04 | 0,19 | 0,20 | 0,16 | 0,26 | 0,07 |
| +/- | 0,00 | -0,13 | 0,02 | 0,01 | -0,03 | 0,07 | -0,09 |
| | | | | | | | |
| FG | m1m2m3 | hsl | yiq | ycbcr | ych1ch2 | lab | l1l2l3 |
| MOY_L1 | 0,04 | 0,09 | 0,03 | 0,04 | 0,04 | 0,09 | 0,04 |
| MOY_L2 | 0,08 | 0,11 | 0,05 | 0,05 | 0,04 | 0,12 | 0,11 |
| MOY_L3 | 0,07 | 0,14 | 0,18 | 0,19 | 0,21 | 0,12 | 0,08 |
| MOY_L4 | 0,07 | 0,09 | 0,03 | 0,03 | 0,03 | 0,07 | 0,07 |
| MOY_L5 | 0,07 | 0,08 | 0,04 | 0,04 | 0,04 | 0,12 | 0,07 |
| +/- | -0,11 | -0,07 | -0,11 | -0,10 | -0,10 | -0,07 | -0,10 |

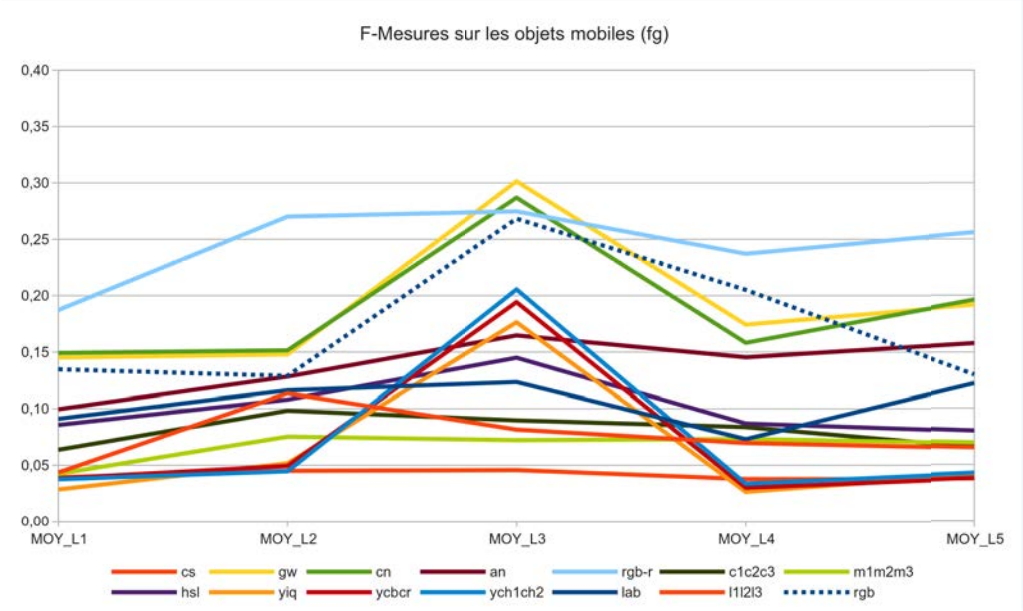
(b) Objets mobiles (fg)

TABLE 2.12 – F-Mesures moyennes obtenues pour chaque invariant colorimétrique sur un test avec un apprentissage sur la séquence L1, L2, L3, L4 et L5 pour la base "RUE".

La figure 2.10 présente les résultats de segmentation sur une image de chaque séquence ($L1$ à $L5$) après un apprentissage effectué sur des frames issues de la séquence $L1$. Nous pouvons constater sur cette figure que les changements colorimétriques induits au cours de la journée ont un impact très fort sur la qualité globale de la détection. On constate une quantité relativement importante de bruit dans les régions de l'arrière-plan, causée notamment par ces changements de luminosité et donc de couleurs. Les invariants colorimétriques ayant donné les meilleurs résultats précédemment ne permettent pas de pallier totalement à ce problème, à l'exception de l'invariant RGB-Rank, qui supprime une forte quantité de bruit quelle que soit la séquence testée. Néanmoins, nous pouvons noter que les ombres ne sont que difficilement supprimées par la stratégie proposée. Une perspective de ces travaux pourrait donc consister à intégrer à cette stratégie globale une méthode efficace de détection des ombres.




(a) BG




(b) FG

FIGURE 2.9 – F-Mesures moyennes obtenues pour chaque invariant colorim trique sur un test avec un apprentissage sur la s quence L1, L2, L3, L4 et L5 pour la base "RUE".

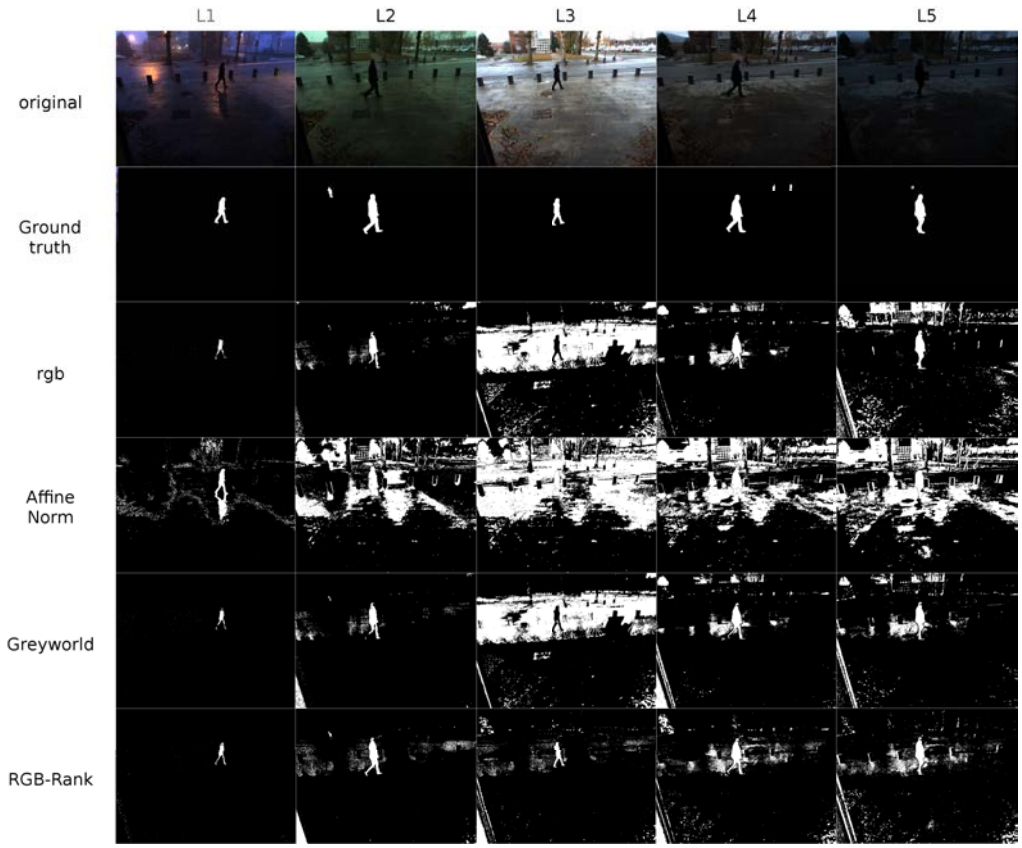



FIGURE 2.10 – R sultats de segmentations sur une image de chaque s quence (L1 L5), apr s un apprentissage sur la s quence L1 de la base ext rieure ("RUE") et selon l'invariant colorim trique utilis .

# 4 Conclusion

Dans ce chapitre, nous avons présenté une stratégie globale de détection d'objets mobiles basées sur l'algorithme du Codebook couplé à l'utilisation d'invariants colorimétriques. Notre objectif à long terme étant de disposer d'une méthode opérable en tous lieux (intérieur et extérieur) et en tous temps (à différents moments de la journée), nous sommes allés au délà de l'utilisation classique de la Microsoft Kinect en utilisant une caméra stéréoscopique pour les scènes extérieures. Nous avons également enregistré et préparé trois bases de données vidéo (une base "BUREAU", une base "JARDIN" et une base "RUE") de sorte à se confronter à un maximum de difficultés en termes de changements de luminosité, arbres, etc. Les tests ont pu être évalués à l'aide du critère F-Mesure. Les résultats présentés dans ce chapitre montrent clairement l'intérêt de l'utilisation d'invariants colorimétriques. En effet, les résultats obtenus sans l'utilisation d'invariant colorimétrique, mettent en évidence la présence importante de nombreux faux positifs dans les régions appartenant au fond (background) qui ne sont pas acceptables dans le cadre par exemple d'applications à caractère sécuritaire.

Le pré-traitement des images avant leur utilisation dans la méthode de détection Codebook permet, dans la majorité des cas, une séparation plus nette du fond et des formes. En effet, la modification des couleurs des images prises à plusieurs moments de la journée permet de les lisser et ainsi d'être plus constantes entre chaque horaire d'acquisition. Dans ce chapitre, nous montrons que quelle que soit la base de données vidéo utilisée, les résultats de segmentation sont toujours améliorés (même de peu), que ce soit pour le fond ou pour les objets mobiles, lorsque l'invariant colorimétrique ou l'espace de représentation couleur est idéalement choisi.

Cependant, au vu des résultats présentés, nous émettons l'hypothèse qu'il n'existe *a priori* pas un seul invariant colorimétrique pouvant offrir des bons résultats dans tous les cas et pour toutes les bases d'images. C'est pourquoi nous tentons de dresser une liste d'invariants colorimétriques à privilégier bien qu'il soit nécessaire de la valider en fonction du capteur, de la base d'images concernée, et de la séquence d'apprentissage utilisée.

Il en ressort une liste d'invariants qui permettent généralement d'améliorer les résultats, listés dans le tableau 2.13 :

| **Intitulé** | **Invariant colorimétrique** |
|---|---|
| cs | Chromaticity Space |
| rgb-r | RGB-Rank |
| lab | Espace couleur L*a*b* |
| c1c2c3 | Espace couleur c1c2c3 |

TABLE 2.13 – Liste des invariants améliorant généralement les résultats.

Néanmoins, nous sommes aussi convaincus que ces mêmes résultats pourraient être encore améliorés en intégrant une étape de post-traitement visant à lisser la segmentation obtenue, voire supprimer les ombres encore présentes à l'aide d'une technique adaptée. Enfin, nous pensons également que le fait de ne prendre en compte que la couleur comme information dans l'algorithme Codebook est trop restrictif et que la profondeur pourrait être intéressante. C'est vers l'utilisation de cette caractéristique que nous souhaitons faire tendre le Codebook exploité ici et donc notre stratégie décrite dans le chapitre suivant.

# Chapitre 3

# Segmentation FG/BG par fusion de la couleur et la profondeur (RGB-D)

La fusion de données multi-capteurs constitue un champ de recherche qui s'est élargi durant ces dernières années. Elle couvre de plus en plus de capteurs différents, dont les principaux restent notamment les caméras (de différents types : couleur, stéréoscopique, thermique, TOF, Infra-Rouge, etc) et d'autres capteurs comme les LRF (Laser Range Finder), les centrales inertielles ou encore les systèmes de navigation par satellite, utilisés par exemple dans le domaine automobile pour la détection d'obstacles [21], la navigation autonome [15] ou la localisation [79]. La fusion d'informations issues de capteurs optiques, et plus précisément la fusion RGB-D qui prend en compte les informations de couleur et de profondeur, est également en plein essor depuis plusieurs années avec l'apparition de capteurs stéréoscopiques de plus en plus performants et des prix en baisse, comme la caméra Kinect [46, 22]. Ce type de fusion se répand sur un large spectre de cas d'utilisation, parmi lesquels nous retrouvons la détection et le tracking d'abeilles à l'entrée des ruches [13], la reconstruction d'environnement 3D à partir d'une image stéréoscopique [57] ou encore la segmentation de personnes [7].

Dans le cadre de cette thèse, nous nous intéressons à la segmentation d'objets mobiles en tous lieux et en tous temps, c'est-à-dire à tout moment de la journée. Nous souhaitons obtenir des segmentations fond/forme utilisables à tout moment de la journée. Dans le chapitre précédent, nous avons constaté que les méthodes uniquement basées sur la couleur n'offraient pas des résultats exceptionnels si des changements de luminosité et/ou de couleur importants survenaient à un moment du traitement : l'apprentissage effectué n'est alors plus à jour, même si l'algorithme utilisé est conçu pour inclure plusieurs changements dans la scène, s'ils sont fréquents et cycliques (arrière-plans dynamiques). Pour les autres types de changements affectant la scène elle-même comme les changements lents d'illumination ou de couleur, un système de "tampon" peut être envisagé. Cependant, celui-ci souffre d'une certaine latence, puisqu'il s'agit basiquement de maintenir un modèle temporaire dans lequel sont placées les occurrences de pixels non rencontrés pendant l'apprentissage mais dont la fréquence a augmenté après celui-ci, puis de les déplacer dans le modèle d'arrière-plan principal une fois que ces occurrences ont été observées pendant un minimum de temps.

Si cette méthode dite "tampon" n'est pas disponible ou non souhaitée pour des causes de limitations matérielles, l'utilisation d'une caractéristique de la scène qui reste constante à tout moment de la journée, y-compris en milieu extérieur peut être envisagée. La recherche d'une telle caractéristique conduit souvent à considérer la profondeur du point de l'espace correspondant à chaque pixel de l'image. Elle peut être obtenue de plusieurs manières : à l'aide d'un capteur actif de type Microsoft Kinect ou bien par le calcul d'une carte de disparités construite à partir des images gauche et droite d'une caméra stéréoscopique. C'est cette stratégie que nous avons considéré dans ce troisième chapitre. Les bases de données que nous avons utilisé sont les mêmes que pour le chapitre précédent.

Pour les séquences d'acquisition en milieu intérieur, nous avons employé comme cela est généralement le cas, une caméra Kinect. Nous en expliquons le principe de fonctionnement dans la section 1.1.

Pour les séquences d'acquisition en milieu extérieur, l'utilisation de capteurs actifs basés sur l'utilisation de lumière infra-rouge telle que le capteur Kinect n'étant pas possible, nous avons choisi d'utiliser une caméra stéréoscopique Bumblebee 2 de chez Point Grey, disposant de deux capteurs couleur CCD (gauche et droit, avec la certitude que les deux images sont bien prises au même instant). De ces deux images sera alors calculée une carte de disparités. Une fois cette carte de disparités obtenue, nous l'intégrerons dans la méthode de détection Codebook afin d'améliorer

la classification finale des pixels. Nous constaterons son impact ainsi que celui de l'utilisation des invariants colorimétriques comme nous l'avons fait au chapitre précédent.

# 1 Principe de Fusion de données couleur et profondeur

## 1.1 Capteur Kinect

L'utilisation de capteur actif a explosé ces dernières années pour plusieurs raisons : leur mode de fonctionnement permet d'obtenir des cartes de profondeur de très bonne qualité, facilement utilisables car assez peu bruitées, mais surtout parce que leur prix a fortement chuté grâce à la mise à disposition de ces dispositifs au grand public à des fins vidéoludiques. L'usage de ces capteurs pour la recherche est apparu à ce moment-là, remplaçant parfois des appareils de technologie proche comme les caméras Temps-de-Vol (Time-of-Flight) dont les prix sont plus importants et dont la résolution est souvent trop basse pour pouvoir être utilisées avec un angle de vue suffisant. Le capteur Microsoft Kinect appartenant à cette catégorie a déjà été utilisé dans le cadre de la segmentation fond/forme, y compris en utilisant l'algorithme Codebook [22], pour segmenter des objets mobiles en milieu contrôlé.

Dans sa première version (la seconde n'étant disponible que depuis fin 2014), la caméra Kinect dispose d'un capteur RGB standard permettant une acquisition à 30 images par seconde, avec une résolution de 320 × 240 (en 16 bits) ou bien de 640 × 480 (en 32 bits) mais aussi d'un couple émetteur-récepteur infra-rouge (figure 3.1). Cet émetteur projette une grille de points lumineux dans le domaine de l'infra-rouge, puis mesure l'intensité de la lumière reflétée pour chaque point : une intensité forte signifie une importante proximité avec l'objet ayant reflété le rayon infra-rouge ; au contraire, une lumière reflétée d'intensité faible dénote un certain éloignement. En utilisant ce principe, le capteur est ainsi capable de déterminer une carte de disparités pour des distances comprises entre approximativement 0.7m et 6m. Néanmoins, les angles de vision horizontaux et verticaux du couple émetteur/récepteur infra-rouge (environ 62.0°et 48.6°respectivement) sont légèrement supérieurs à ceux de la caméra couleur (environ 58.5°et 45.6°respectivement).

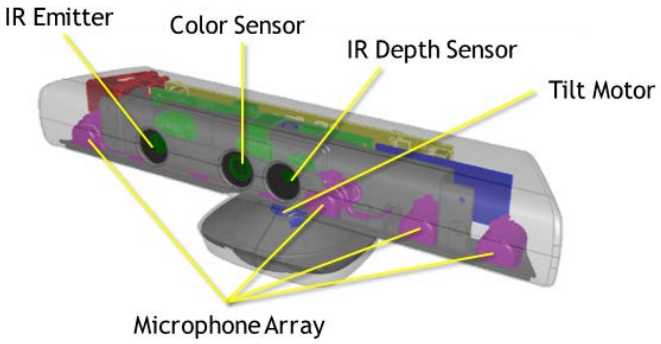



FIGURE 3.1 – Caméra Microsoft Kinect (version 1)

De par le fonctionnement de ce capteur, aucune mise en correspondance n'est nécessaire pour l'obtention de cartes de disparités. Chaque Kinect intégrant ses propres données de calibrage, le capteur est capable de produire des cartes de disparités rectifiées en rapport avec la scène acquise par la caméra couleur.

## 1.2 Mise en correspondance stéréoscopique d'images couleur

L'utilisation d'images stéréoscopiques dans le cadre de la segmentation fond/forme découle de la nécessité d'une information spatiale qu'une image couleur seule ne permet pas d'obtenir. Ces traitements consistent habituellement en une reconstruction (plus ou moins partielle) d'un environnement 3D dans un support informatisé, ou encore dans des applications de suivi de cibles nécessitant la prise en compte de la position proche ou éloignée de ces cibles.

Dans ce contexte, l'utilisation d'images stéréoscopiques, c'est-à-dire d'images obtenues par un dispositif spécifique, est envisagée. Ces images, dénommées habituellement image gauche et image droite, sont acquises au même moment par ce type de caméras nécessairement calibré, de sorte à pouvoir par la suite rectifier ces images gauche et droite (c'est-à-dire les transformer afin que la projection d'un point de l'espace 3D sur les images gauche et droite se trouve sur la même ligne horizontale, dite ligne épipolaire) et de procéder à la reconstruction 3D.

Une fois les deux images gauche et droite rectifiées, il est nécessaire d'apparier chaque pixel de l'image gauche à son correspondant dans l'image droite. Cette étape est dénommée "mise en correspondance" et consiste souvent à déterminer une mesure de corrélation minimale entre deux voisinages de pixels gauche et droite, parmi plusieurs voisinages possibles disposés sur les droites épipolaires conjuguées par ces pixels. Ces voisinages à comparer peuvent être soit composés de niveaux de gris ou d'informations de couleur ; dans ce second cas, les trois composantes couleur sont considérées séparément.

La littérature abonde de mesures de correspondance. On les classe habituellement dans deux catégories : locale et globale. La mise en correspondance locale consiste à établir une correspondance entre deux pixels à partir d'une zone d'agrégation autour d'eux. La recherche est effectuée dans l'image droite à partir d'une fenêtre autour de la même position que celle du pixel à apparier dans l'image de gauche. Pour chacun des pixels de cette fenêtre, on prend en considération une zone d'agrégation (ou voisinage) que l'on compare via une mesure de corrélation au voisinage du pixel à apparier de l'image gauche. Le pixel de l'image droite, dont la zone d'agrégation sera mesurée comme la plus semblable au pixel étudié de l'image gauche, sera alors considéré comme son homologue.

Les mesures *locales* les plus classiques sont SAD (*Sum of Absolute Differences*), SSD (*Sum of Square Differences*), NCC (*Normalized Cross-Correlation*) ainsi que leurs versions à distances centrées (*Zero-mean distances*) dans lesquelles la moyenne des intensités de chaque fenêtre est soustraite. Les équations 3.1 à 3.6 décrivent chacune de ces méthodes entre deux voisinages $I_L$ et $I_R$ (ou fenêtres, de taille $b \times m$) de pixels. Une synthèse beaucoup plus exhaustive de ces mesures de corrélation (mesure de similarité ou de dissimilarité selon les cas) est présentée dans [12].

$$SAD = \sum_{x=0}^{b-1} \sum_{y=0}^{m-1} |I_L(x,y) - I_R(x,y)| \tag{3.1}$$

$$ZSAD = \sum_{x=0}^{b-1} \sum_{y=0}^{m-1} |I_L(x,y) - M_L - I_R(x,y) - M_R| \tag{3.2}$$

avec $M_x$ la moyenne des éléments de la fenêtre $I_x$ où $x \in \{L, R\}$.

$$SSD = \sum_{x=0}^{b-1} \sum_{y=0}^{m-1} (I_L(x,y) - I_R(x,y))^2 \tag{3.3}$$

$$ZSSD = \sum_{x=0}^{b-1} \sum_{y=0}^{m-1} (I_L(x,y) - M_L - I_R(x,y) - M_R)^2 \tag{3.4}$$

avec $M_x$ la moyenne des éléments de la fenêtre $I_x$ où $x \in \{L, R\}$.

$$NCC = \frac{\sum_{x=0}^{b-1} \sum_{y=0}^{m-1} I_L(x,y).I_R(x,y)}{\sqrt{\sum_{x=0}^{b-1} \sum_{y=0}^{m-1} (I_L(x,y) - M_L)^2}.\sqrt{\sum_{x=0}^{b-1} \sum_{y=0}^{m-1} (I_R(x,y) - M_R)^2}} \tag{3.5}$$

$$ZNCC = \frac{\sum_{x=0}^{b-1} \sum_{y=0}^{m-1} (I_L(x,y) - M_L) \times (I_R(x,y) - M_R)}{\sqrt{\sum_{x=0}^{b-1} \sum_{y=0}^{m-1} (I_L(x,y) - M_L)^2 . \sum_{x=0}^{b-1} \sum_{y=0}^{m-1} (I_R(x,y) - M_R)^2}} \tag{3.6}$$

La mise en correspondance *globale* réduit la zone d'agrégation au pixel de l'image gauche considéré et nous cherchons une minimisation (ou maximisation, selon le cas) globale du coût d'une fonction objective. De nombreuses méthodes globales comme GraphCut [48], LBP (Loopy Belief Propagation) [61] ou l'algorithme ICM (Iterated conditional modes) [75] font appel à une modélisation sur les champs de Markov aléatoires. La fonction objective est d'abord définie à partir des différentes contraintes à respecter, comme une somme de termes représentant chacun une de ces contraintes de telle manière que la minimisation (ou maximisation) de cette fonction corresponde à la meilleure solution. On optimise alors cette fonction objective selon une méthode d'optimisation choisie. Plusieurs méthodes d'optimisation ont été proposées : sans être exhaustif, nous pouvons citer la programmation dynamique [28], la régularisation [2], le recuit simulé [3], la coupure de graphe (ou *graphcut*) [45], les réseaux de neurones [68], les algorithmes génétiques [68, 12] et la propagation de croyance [20].

Devant le nombre important de méthodes existantes, plusieurs benchmarks ont été mis en place afin de classer ces méthodes par ordre d'efficacité. Ces benchmarks sont constitués d'un certain nombre de séquences d'images stéréoscopiques accompagnées de leurs vérités terrain : nous pouvons notamment citer le classement Middlebury [55], le classement KITTI [32] ou encore HCI Robust Vision Challenge [39]. Dans ce chapitre, nous présentons le classement Middlebury qui fait référence dans le domaine de la stéréovision.

**Classement Middlebury**

Dans ce paragraphe, nous reportons le haut du classement des algorithmes actuels de calcul de cartes de disparités à partir d'images stéréoscopiques. Le tableau 3.1 illustre pour chaque algorithme considéré la moyenne pondérée (pour compenser la variation de difficulté entre les bases d'images) de la métrique utilisée (ici le pourcentage de "mauvais" pixels dont l'erreur est supérieure à 2.0) sur les différents tests effectués sur 15 paires d'images différentes. Le choix de considérer ce classement est motivé par la difficulté de mesurer les différences dans les résultats entre ces méthodes.

Un grand nombre de méthodes supplémentaires existent dans la littérature mais nous nous limitons ici à la liste fournie dans la version 3 de l'évaluation de Middlebury.

D'après ce classement, nous remarquons un assez bon positionnement ($4^{\text{ème}}$ position) de la méthode Semi-Global Matching (SGM) [41] dont la version itérative de ce même algorithme [40] a remporté la dernière édition du HCI Robust Vision Challenge [39], nous pouvons supposer que cette méthode présente des avantages certains en termes d'efficacité et en temps de traitement. Par ailleurs, cet algorithme se classait encore très récemment dans le haut du classement, mais les avancées dans le domaine l'ont rétrogradé. Néanmoins, du fait de l'écart suffisamment faible entre

TABLE 3.1 – Classement des différentes métriques fourni par Middlebury [55].

| Rang | Algorithme | Moyenne pondérée |
|---|---|---|
| 1. | MeshStereo [81] | 13.4 |
| 2. | TMAP [64] | 17.1 |
| 3. | IDR [49] | 18.4 |
| 4. | SGBM [41] | 18.7 |
| 5. | LPS [72] | 19.4 |
| 6. | SNCC [17] | 22.2 |
| 7. | SGBM (OpenCV) [60] | 24.0 |
| 8. | LAMC_DSM [74] | 26.2 |
| 9. | Cens5 [42] | 26.9 |
| 10. | ELAS [33] | 29.3 |
| 11. | REAF [14] | 31.4 |

les nouvelles méthodes et le Semi-Global Block-Matching (SGBM), mais aussi de l'utilisation de plus en plus large de ce dernier dans l'industrie, notamment pour des problématiques de vision pour l'automobile, comme la détection de véhicules au passage de carrefours [66, 65, 29, 31], nous pouvons considérer que cet algorithme (dont nous rappelons le principe ci-après) reste une référence dans ce domaine. C'est pourquoi nous avons fait le choix de retenir pour nos travaux et d'utiliser la version implémentée par OpenCV (SGBM1 dans le tableau 3.1). Nous présentons cet algorithme de manière plus détaillée dans le paragraphe 1.2.

**Semi-Global Block Matching**

La méthode Semi-Global Block Matching (SGBM) proposée par Hirschmuller *et al* [41] permet de générer une carte de disparités et se base sur l'enchaînement de quatre étapes, décrites ci-après :

1. *Le calcul des coûts d'appariement* est effectué à partir de l'intensité des pixels ou de leur couleur. Un gradient peut être calculé pour prendre en compte les différences radiométriques.
2. *L'agrégation des coûts* connecte les coûts des voisinages de pixels en déterminant le plus court chemin avec un voisinage de pixels de l'image comparée. Ceci est réalisé à l'aide d'une fonction d'énergie globale effectuant cette recherche dans toutes les directions dans l'image.
3. *Le calcul des disparités* est réalisé à partir de l'énergie calculée précédemment selon une implémentation "le gagnant emporte tout" (*winner-takes-all*).
4. *L'affinage des disparités* est ensuite effectué pour stabiliser la carte de disparités. Il consiste en un filtrage des pics de disparité, la sélection des disparités d'intensités constantes et une interpolation des discontinuités de disparités. Cette étape est cependant séparée de la méthode de mise en correspondance à proprement parler.

**Calcul des coûts d'appariement** Pour un pixel $p$ de l'image de gauche $b$, on calcule tout d'abord l'intensité $I_{b,p}$ de son voisinage et celle de sa correspondance potentielle $I_{m,q}$ dans l'image droite $m$. Le pixel correspondant $q$ dans l'image droite est recherché sur la ligne épipolaire conjuguée du pixel $p$ de l'image $I_b$. Pour des images rectifiées, $q = e_{b,m}(p, D) = [p_x - D, p_y]^T$, où $e_{b,m}$ symbolise la ligne épipolaire conjuguée dans l'image droite $m$ pour le pixel $p$ de l'image gauche $b$, $D$ étant la valeur de disparité. La taille du voisinage a une influence sur la robustesse de la carte de disparités obtenue. Le coût de disparité $C(p, D)$ est calculé pour le pixel $p$ et la disparité $D$. Il est donné par

exemple par la méthode proposée par Birchfeld et Tomasi [6] sous la forme de la valeur absolue de la différence des minimums des intensités du voisinage considéré. Mais d'autres fonctions de calcul de ce coût peuvent être utilisées. A la fin de cette étape, nous disposons d'une première estimation des disparités pour la quasi-totalité des pixels de l'image.

**Agrégation des coûts** Pour affiner les mauvaises disparités calculées précédemment et trouver la disparité correcte pour les pixels concernés, l'énergie globale $E(D)$ est calculée pour chaque disparité $D$ de la carte de disparités. Des pénalités $P_1$ et $P_2$ sont appliquées lorsque des différences de disparités faibles ou importantes sont observées. Un matching correct est alors déterminé par la minimisation de l'énergie $E(D)$.

$$E(D) = \sum_{p}(C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]) \tag{3.7}$$

où $E(D)$ est l'énergie pour l'image de disparités et $D_p$ et $D_q$ sont respectivement les valeurs de disparités pour les pixels $p$ et $q$.

Le premier terme de l'équation 3.7 est la somme de tous les coûts d'appariement pour la disparité $D$. Les deuxièmes et troisième termes ajoutent respectivement la pénalité $P_1$ et la pénalité $P_2$ si la condition $T$ associée est vérifiée, et ce pour tous les pixels $q$ dans le voisinage $N_p$ du pixel $p$ (dans lequel les disparités varient normalement peu). La condition $T$ est une fonction qui retourne 1 si son argument est vrai, 0 sinon. $P_1$ est la pénalité ajoutée si la différence de disparité entre les pixels voisins est égale 1 ; $P_2$ est la pénalité ajoutée pour une différence supérieure à 1 entre les pixels voisins.

La fonction 3.7 minimisée produit une carte de disparités affinée en fonction des paramètres $P_1$ et $P_2$. Cependant, la minimisation de la fonction d'énergie dans l'espace image 2D étant un problème NP-complet, on la réduit à un problème à 1 dimension. L'idée est d'agréger le coût minimum d'appariement des disparités voisines dans toutes les directions. Soit $L_r$ un chemin traversé dans la direction $r$. Le coût $L_r(p, d)$ du pixel $p$ pour la disparité $D$ est défini récursivement par :

$$\begin{aligned} L_r(p, D) = & C(p, D) \\ & + min(L_r(p - r, D), \\ & L_r(p - r, D - 1) + P_1, \\ & L_r(p - r, D + 1) + P_1, \\ & min_i L_r(p - r, i) + P_2) \\ & - min_k L_r(p - r, k) \end{aligned} \tag{3.8}$$

où le coût actuel $C(p, D)$ est ajouté au pixel précédent dans la direction $r$ afin de trouver la disparité, à l'aide des coûts suivants :

- Le coût minimal au pixel précédent avec la disparité $D$ ;
- Le coût au pixel précédent avec les disparités $D-1$ et $D+1$, à laquelle est ajoutée la pénalité $P_1$ ;
- Le coût au pixel précédent avec les disparités inférieures à $D - 1$ et supérieures à $D + 1$, à laquelle est ajoutée la pénalité $P_2$.

A noter que dans l'équation 3.8, la valeur du coût $L$ augmente tout le long de sa résolution, ce qui peut donner une valeur finale très grande. Aussi, afin de limiter la valeur finale du coût et

ainsi optimiser le calcul, le coût minimal du chemin du pixel précédent $k$ est soustrait dans cette équation. Ainsi, les valeurs de coût calculées restent petites et le coût du chemin minimum ne change pas, puisque le coût du chemin minimum du pixel précédent est constant. On calcule alors le coût agrégé $S(p, D)$ pour le pixel $p$ et la disparité $D$ comme suit :

$$S(p, D) = \sum_r L_r(p, D) \tag{3.9}$$

où $r$ est la direction utilisée pour converger vers le pixel $p$, et $L_r(p, D)$ le coût minimal du chemin pris dans la direction $r$ depuis le pixel $p$ pour la disparité $D$.

Hirschmuller [41] propose de traiter 16 directions. L'implémentation d'OpenCV n'en considère que 5 par défaut pour des raisons de rapidité d'exécution (une seule itération), mais il est possible d'activer l'algorithme original utilisant la programmation dynamique (2 itérations). Dans le cadre des travaux de cette thèse, c'est l'algorithme original (2 itérations) qui a été utilisé.

**Calcul des disparités** La carte de disparités finale est calculée en choisissant, pour chaque pixel $p$ et pour chaque valeur de disparité $D$, la disparité qui produit le coût minimum pour ce pixel. Ce choix est déterminé par le minimum d'une courbe quadratique pour tous les coûts des pixels voisins.

Il est possible d'améliorer la qualité de la carte de disparités (notamment déterminer les occlusions) en calculant les disparités une fois de plus en inversant le rôle des deux images à mettre en correspondance. On obtient alors la carte de disparités $D_b$, il suffit alors d'effectuer une comparaison entre chaque pixel des deux cartes. Si cette comparaison relève des disparités différentes, le pixel est marqué comme invalide (équation 3.10) :

$$D_p = \begin{cases} D_{b,p} & si |D_{b,p} - D_{m,q}| \leq 1, q = e_{b,m}(p, D_{b,p}), \\ D_{inv} & sinon. \end{cases} \tag{3.10}$$

## 1.3 Codebook RGB-D

Comme expliqué dans le chapitre 1, la méthode de segmentation Codebook proposée par Kim *et al* [47] fonctionne au niveau pixel. L'algorithme extrait deux types d'informations à partir de la couleur de chaque pixel afin de les comparer avec les données du modèle d'arrière-plan : la distorsion de couleur et la distorsion d'intensité. Nous avons pu constater précédemment l'efficacité de cette méthode, mais également ses difficultés dans le cas d'un traitement continu tout au long de la journée où la scène observée est soumise à des changements de couleur et d'intensité lumineuse rapides.

Nous avons également pu remarquer que lorsque la scène est impactée par une augmentation de luminosité importante (quand la source lumineuse est partiellement occultée par une silhouette par exemple), les faux-positifs observés dans les segmentations sont causés par des distorsions d'intensité trop importantes. Les distorsions de couleur sont faiblement impactées par ces sur-illuminations. L'utilisation des invariants colorimétriques seuls ne supprime pas totalement ces faux-positifs, bien que nous ayons confirmé leur impact positif dans le premier chapitre. Il est donc intéressant d'étendre cette méthode, en vue d'améliorer davantage les résultats.

À ce titre, nous pensons que l'information de profondeur obtenue avec le calcul de cartes de disparités peut être utilisée au sein même de l'algorithme de segmentation fond/forme. Celle-ci étant une donnée à 1 dimension, elle peut être traitée de la même manière que l'information d'intensité lumineuse. Il convient ensuite de modifier la condition de fusion de ces trois informations afin de classer un pixel donné comme fond ou comme objet mobile. Nous nous basons ici sur la proposition formulée par Fernandez-Sanchez *et al* [22] qui ont testé cette méthode sur des images couleur et des cartes de disparités toutes deux obtenues à l'aide d'une caméra Kinect en milieu intérieur uniquement. Nous employons cette méthode en vue d'observer son comportement sur des scènes extérieures (et donc avec un autre capteur : le BumbleBee 2), tout en profitant des avantages tirés de l'utilisation des invariants colorimétriques décrits précédemment.

Concrètement, l'ajout de l'information de profondeur dans l'algorithme du Codebook est effectué comme pour l'intensité lumineuse : deux bornes inférieure $D_{low}$ et supérieure $D_{high}$ sont déterminées à partir des disparités minimale $\check{D}$ et maximale $\hat{D}$ observées pour un codeword donné, et à l'aide de deux coefficients $\alpha_D$ et $\beta_D$ (équation 3.11) :

$$\begin{aligned} D_{low} &= \alpha_D \hat{D} \\ D_{high} &= min\{\beta_D \hat{D}, \tfrac{\check{D}}{\alpha_D}\} \end{aligned} \tag{3.11}$$

avec $\alpha_D \in [0.4; 0.7]$ et $\beta_D \in [1.1; 1.5]$. Ces coefficients, comme ceux définis précédemment pour les distorsions de couleur et d'intensité lumineuse, sont déterminés empiriquement afin d'offrir les meilleurs résultats sur un maximum d'images de chaque base, sachant que comme le propose Fernandez *et al* [22], $\alpha_D$ doit être compris entre 0.4 et 0.7 et $\beta_D$ entre 1.1 et 1.5. Comme pour les paramètres $\alpha_B$ et $\beta_B$ (pour le calcul de la distorsion d'intensité lumineuse), un seuil plus faible permet de considérer un intervalle de validité $[I_{low}, I_{high}]$ plus large ; une valeur plus importante génère un intervalle plus étroit.

En ce qui concerne la distorsion de disparité, elle est vérifiée selon la condition formulée dans l'équation 3.12.

$$disparityDist(D, \langle \check{D}, \hat{D} \rangle) = \begin{cases} vrai & \text{si } estInvalide(D) \vee (D_{low} \leq D \leq D_{high}) \\ faux & \text{sinon.} \end{cases} \tag{3.12}$$

Le terme $estInvalide(D)$ renvoie vrai si la valeur de disparité $D$ est invalide (généralement une valeur égale à 0).

Enfin, la condition de fusion des informations (équation 3.13) basée sur les trois types de distorsions (couleur, intensité, disparité) est utilisée pour déterminer à quelle classe, fond (bg) ou objet mobile (fg) appartient le pixel considéré.

$$decision(x) = \begin{cases} BG & \text{si } (colorDist(x) \leq \epsilon_1 \vee \\ & (\epsilon_1 < colorDist(x, c_m) \leq \epsilon_2 \\ & \wedge disparityDist(D, \langle \check{D}, \hat{D} \rangle))) \wedge \\ & brightnessDist(I, \langle \check{I}, \hat{I} \rangle) \wedge \\ & disparityDist(D, \langle \check{D}, \hat{D} \rangle) \\ FG & \text{sinon} \end{cases} \tag{3.13}$$

Nous constatons que le terme $disparityDist(D, \langle \check{D}, \hat{D} \rangle)$ est vrai si la disparité observée $D$ pour le pixel est indéfinie. C'est le cas lorsque ce pixel se trouve dans une zone d'occultation stéréoscopique, ce qui empêche l'algorithme de mise en correspondance de déterminer un appariement correct entre deux pixels (gauche et droit dans les images stéréoscopiques). Pour ce type de pixels, la disparité associée est généralement nulle. Ainsi, la condition de distorsion de disparité est vérifiée lorsque la disparité observée est invalide ($estInvalide(D) = vrai$) afin que seules les distorsions de couleur et d'intensité soient prises en compte, ce qui revient au comportement de l'algorithme original proposé par Kim *et al* [47].

Ainsi, cette version de l'algorithme peut être résumée selon les 3 cas suivants :

1. Si la disparité observée pour le pixel considéré est invalide, l'algorithme n'utilise que les distorsions de couleur et d'intensité, ce qui revient à l'algorithme original.
2. Si la distorsion de couleur est inférieure au seuil $\epsilon_1$, les distorsions d'intensité et de disparité sont testées séparément.
3. Si la distorsion de couleur est comprise entre $\epsilon_1$ et $\epsilon_2$, alors la distorsion de disparité est testée ensemble pour valider un choix.

Si une correspondance est établie entre le pixel testé et un codeword contenu dans le codebook associé à ce pixel, alors il est classé comme du fond. Dans le cas contraire, il est classé comme objet mobile.

L'effet attendu de cette modification est notamment la suppression de faux-négatifs ("trous") dans les segmentations d'objets mobiles, ainsi qu'une meilleure segmentation générale de ces derniers lorsqu'ils se trouvent à une distance plus importante de la caméra. En effet, plus un objet mobile se situe loin de la caméra, plus sa taille se réduit et plus sa détection est difficile : si sa segmentation est partielle, les régions de petite taille résultant de cette détection sont susceptibles d'être considérées comme du bruit (faux-positifs). L'utilisation de la disparité apporte une condition supplémentaire permettant d'obtenir des silhouettes moins "trouées", et donc globalement mieux segmentées.

Concernant la suppression des faux-positifs présents dans les régions du fond, il est probable qu'elle ne soit pas aussi importante. L'utilisation d'un modèle d'arrière-plan basé sur la couleur seule alors même que ce modèle n'est plus à jour (c'est-à-dire qu'il ne contient pas de codeword pouvant être mis en correspondance avec les nouvelles images ayant un aspect colorimétrique

différent) laisse apparaître de nombreuses régions considérées à tort comme des objets mobiles. Un modèle d'arrière-plan basé sur les deux informations de couleur et de disparité nécessite *a fortiori* de vérifier les conditions de distorsion validant à la fois la couleur du pixel considéré et sa disparité. Ainsi, comme le montre l'équation 3.13, même si la disparité observée correspond à celle du modèle, la correspondance n'est pas forcément établie si les distorsions de couleur ou d'intensité ne sont pas également validées. Par conséquent, l'information de couleur pour les pixels des régions de l'arrière-plan reste insuffisante pour leur bonne détection. C'est pourquoi nous avons proposé dans ce début de chapitre de considérer l'algorithme Codebook RGB-D couplé à l'utilisation des invariants colorimétriques considérés précédemment.

Il est également à noter que l'utilisation des invariants colorimétriques appliqués en pré-traitement sur les frames gauche et droite acquises par la caméra stéréoscopique n'a pas eu pour effet d'améliorer la qualité des cartes de disparités avec la méthode Semi-Global Block Matching (illustration sur la figure 3.2). Néanmoins, ce constat fait état d'une observation qualitative : une étude plus approfondie est à envisager en perspectives de ces travaux.

FIGURE 3.2 – Exemples de cartes de disparités obtenues avec la BumbleBee 2 (de gauche à droite : images originales, avec l'invariant colorimétrique RGB-Rank et avec l'invariant colorimétrique Chromaticity Space).

# 2 Segmentation d'images par Codebook RGB-D

Maintenant que nous avons rappelé le principe du Codebook RGB-D, nous allons présenter les résultats de segmentation sur les trois bases de données citées dans le chapitre 1. Le protocole d'évaluation reste identique. Néanmoins, pour considérer l'information de profondeur, deux capteurs ont été utilisés en fonction de la base étudiée.

De par le capteur utilisé, la base d'images "BUREAU", les images couleur et les cartes de disparités ont été acquises à l'aide d'un capteur Kinect. Celui-ci produit des images couleur et cartes de profondeur d'une taille de 640×480 à une fréquence d'environ 15 images par seconde. Les cartes de disparités sont limitées à une disparité maximum de 20. Pour les deux autres bases d'images "JARDIN" et "RUE", les images ont été acquises à l'aide d'une caméra stéréoscopique BumbleBee 2. Les cartes de disparités ont été calculées à partir de ces images stéréoscopiques à l'aide de la méthode du Semi-Global Block Matching décrite précédemment. Les segmentations sont alors effectuées à partir de l'image de gauche redimensionnée à une résolution de 640×480.

Dans les sous-sections suivantes, nous présenterons les résultats obtenus avec la méthode Codebook RGB-D, sans puis avec invariants colorimétriques. Comparer les résultats de segmentation entre les deux méthodes RGB et RGB-D avec l'utilisation des invariants colorimétriques n'a pas de sens car il serait difficile de déterminer si la différence observée est dûe à l'utilisation de l'information de profondeur ou à celle des invariants colorimétriques. D'autre part, les méthodes de segmentation issues de la littérature ne sont généralement pas testées avec des invariants colorimétriques. Nous préférons donc dans un premier temps, comparer les résultats RGB vs RGB-D sans l'utilisation d'invariants colorimétriques.

Ceci nous permettra de mesurer l'impact de l'utilisation de la disparité au sein de l'algorithme de détection Codebook. Puis, dans un second temps, nous mesurerons l'impact des invariants colorimétriques sur une segmentation réalisée par Codebook RGB-D.

## 2.1 Détection d'objets mobiles sur la base "BUREAU"

La base "BUREAU" profite de cartes de disparités relativement précises. L'effet sur la qualité des segmentations est réel, mais limité. En effet, d'après le tableau 3.2, nous observons que les valeurs de F-Mesures sont comprises entre 0.72 et 0.82 pour le fond (à l'exception du test avec apprentissage sur la base L5), et que les résultats calculés par la méthode RGB-D montrent en moyenne un gain compris entre 0.01 et 0.04. Les F-Mesures relatives aux objets mobiles affichent des valeurs comprises entre 0.25 et 0.42, alors que les résultats calculés par la méthode RGB-D montrent un gain compris entre +0.02 et +0.07.

| BG | RGB | RGB-D |
|---|---|---|
| MOY_L1 | 0,72 | **0,74** |
| MOY_L2 | 0,82 | **0,82** |
| MOY_L3 | **0,81** | 0,80 |
| MOY_L4 | 0,75 | **0,79** |
| MOY_L5 | 0,17 | **0,18** |
| MOY_L6 | 0,73 | **0,74** |

(a) Fond (bg)

| FG | RGB | RGB-D |
|---|---|---|
| MOY_L1 | 0,25 | **0,30** |
| MOY_L2 | 0,35 | **0,42** |
| MOY_L3 | 0,34 | **0,39** |
| MOY_L4 | 0,29 | **0,36** |
| MOY_L5 | 0,16 | **0,18** |
| MOY_L6 | 0,30 | **0,33** |

(b) Objets mobiles (fg)

TABLE 3.2 – F-Mesures moyennes calculées sur le fond (bg) et sur les objets mobiles (fg) avec des avec des apprentissages sur les séquences L1, L2, L3, L4, L5 et L6 de la base "BUREAU".

Pour une meilleure visualisation des di rences entre ces r sultats, nous illustrons sur la figure 3.3 les r sultats de segmentation avec la m thode Codebook RGB et Codebook RGB-D.

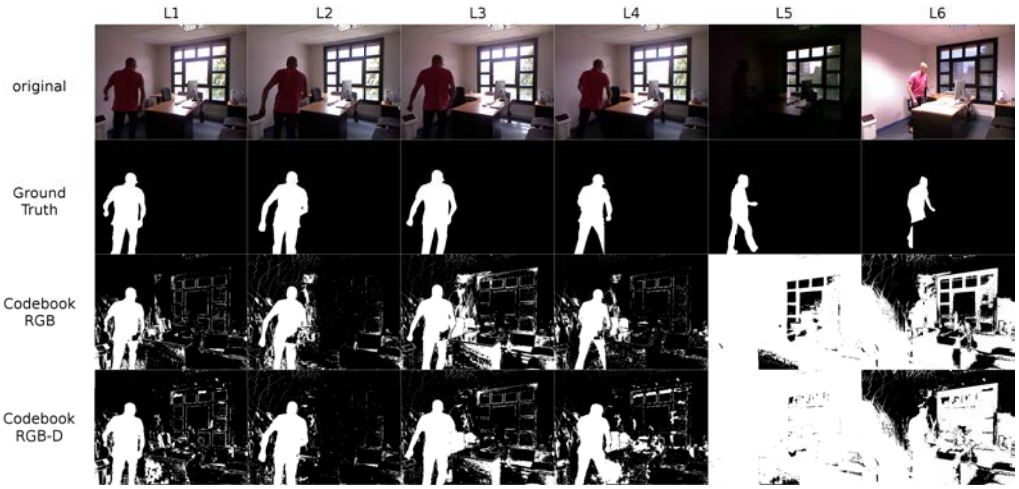



FIGURE 3.3 – R sultats de segmentations obtenues avec la m thode Codebook RGB et la m thode Codebook RGB-D avec un apprentissage sur la s quence L2 de la base "BUREAU".

## 2.2 Détection d'objets mobiles sur la base "JARDIN"

La base "JARDIN" tant acquise en milieu ext rieur, l'algorithme de segmentation est potentiellement soumis des images pr sentant un bruit plus important. Les variations de lumi re sont galement beaucoup plus visibles qu'en milieu int rieur, savoir avec la base "BUREAU". Ce bruit implique des erreurs de classification localis es majoritairement dans les r gions appartenant au fond. En e et, d'apr s le tableau 3.3, nous observons une l g re d gradation des F-Mesures pour les r gions du fond lorsque l'on consid re la profondeur (entre -0.02 et -0.07), mais aussi une l g re augmentation des F-Mesures pour les objets mobiles (entre +0.02 et +0.06). Ainsi, au prix d'une l g re d gradation de la qualit de segmentation du fond, les objets mobiles sont l g rement mieux segment s. La figure 3.4 illustre visuellement ces r sultats. Nous pouvons alors constater que les "trous" visibles dans les silhouettes lorsque l'on consid re le Codebook RGB (2ème ligne) sont bouch s si l'on consid re la profondeur travers le Codebook RGB-D (3ème ligne).

Cependant, les am liorations restent limit es. Ces r sultats mitig s peuvent s'expliquer par la qualit moindre des cartes de disparit s, pour lesquelles le nombre maximal de disparit s a t augment en comparaison celles de la cam ra Kinect. En e et, la disparit maximale des cartes de disparit s Kinect a t limit e 20 pour des raisons techniques. Une valeur plus lev e a t utilis e avec l'algorithme SGBM pour les images issues du BumbleBee 2. Cette augmentation induit plus de bruit, et donc une moins bonne classification des pixels appartenant th oriquement au fond et qui sont en quantit plus importante.

## 2.3 Détection d'objets mobiles sur la base "RUE"

Comme expliqu dans le premier chapitre, la base "RUE" est la plus compliqu e des trois bases d'images test es. En e et, il s'agit de celle qui a che les plus grandes variations entre les s quences, en termes de couleur et d'intensit . Ces variations impactent forc ment la pr cision des cartes de disparit s puisqu'elles comportent davantage de bruit. Or, les s quences L1 et L5 tant acquises

| BG | RGB | RGB-D |
|---|---|---|
| MOY_L1 | **0,91** | 0,84 |
| MOY_L2 | **0,92** | 0,90 |
| MOY_L3 | **0,96** | 0,92 |
| MOY_L4 | **0,95** | 0,92 |

(a) Fond (bg)

| FG | RGB | RGB-D |
|---|---|---|
| MOY_L1 | 0,10 | **0,12** |
| MOY_L2 | 0,08 | **0,14** |
| MOY_L3 | 0,12 | **0,14** |
| MOY_L4 | 0,13 | **0,15** |

(b) Objets mobiles (fg)

TABLE 3.3 – F-Mesures moyennes calcul es sur le fond (bg) et sur les objets mobiles (fg) avec des avec des apprentissages sur les s quences L1, L2, L3 et L4 de la base “JARDIN”.


FIGURE 3.4 – R sultats de segmentation obtenus avec la m thode Codebook RGB et la m thode Codebook RGB-D avec un apprentissage sur la s quence L2 de la base “JARDIN”.

des moments de la journ e o l’intensit lumineuse fait d faut (respectivement 7h du matin et 17h30 le soir en hiver), a chent une quantit non-n gligeable de bruit.

De fait, les r sultats pr sent s dans le tableau 3.4 a chent une diminution des F-Mesures du fond, m me si celle-ci ne d passe pas 0.06 et ce quelle que soit la s quence d’apprentissage. En ce qui concerne les r gions des objets mobiles, les r sultats montrent cette fois des F-Mesures diminu es ou augment es en fonction de la s quence d’apprentissage utilis e (entre -0.08 et +0.03). Nous constatons que cette base d’images sou re des m mes contraintes que pr c demment. Notons galement que les r sultats pr sent s ici sont des moyennes calcul es sur une base compl te qui compte 5 s quences. Nous constaterons par la suite que c’est e ectivement la s quence L1 qui fait grandement chuter ces moyennes.

La figure 3.5 illustre ces constatations. Du bruit est visible dans plusieurs cas (Codebook RGB-D) alors qu’il ne l’est pas dans les r sultats obtenus avec Codebook RGB. La sensation visuelle donne l’impression d’une l g re am lioration, notamment dans la segmentation de l’objet mobile visible dans ces images, mais la d gradation se situe en fait dans l’arri re-plan de ces images.

| BG | RGB | RGB-D |
|---|---|---|
| MOY_L1 | **0,94** | 0,92 |
| MOY_L2 | **0,93** | 0,89 |
| MOY_L3 | **0,97** | 0,91 |
| MOY_L4 | **0,83** | **0,83** |
| MOY_L5 | **0,83** | **0,83** |

(a) Fond (bg)

| FG | RGB | RGB-D |
|---|---|---|
| MOY_L1 | **0,13** | 0,10 |
| MOY_L2 | 0,13 | **0,16** |
| MOY_L3 | **0,27** | 0,19 |
| MOY_L4 | **0,21** | 0,15 |
| MOY_L5 | 0,13 | **0,14** |

(b) Objets mobiles (fg)

TABLE 3.4 – F-Mesures moyennes calcul es sur le fond (bg) et sur les objets mobiles (fg) avec des avec des apprentissages sur les s quences L1, L2, L3, L4 et L5 de la base "RUE".

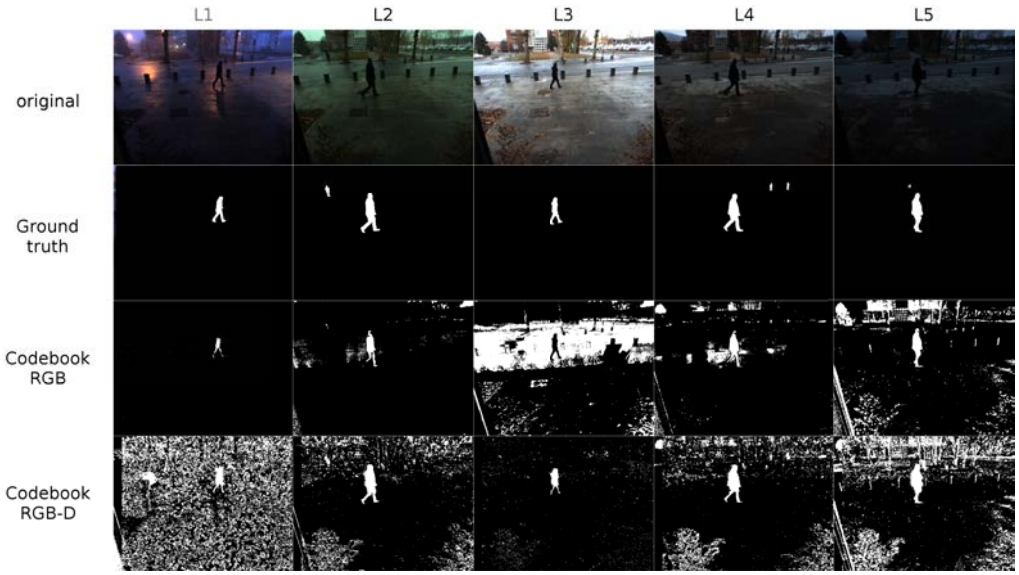



FIGURE 3.5 – R sultats de segmentation obtenus avec la m thode Codebook RGB et la m thode Codebook RGB-D avec un apprentissage sur la s quence L3 de la base "RUE".

## 2.4 Discussion

Malgr des r sultats mitig s sur les bases ext rieures (mais complexes), nous pouvons conclure que la fusion RGB-D au sein du Codebook semble dans certains cas pouvoir am liorer les r sultats de segmentation d'objets mobiles. Aussi, nous avons volontairement souhait valuer des segmentations non filtr es afin de constater de l'e cacit de la m thode et ventuellement proposer en perspective de ces travaux l'int r t d'un filtrage morphologique comme tape de post-traitement.

Au vu de ces r sultats, nous pensons que l'utilisation des invariants colorim triques pourraient, comme cela a t montr dans le premier chapitre, am liorer les r sultats de segmentation d'objets mobiles. Nous pensons aussi qu'il pourrait tre int ressant de r fl chir une autre m thode de fusion des informations de couleur et de profondeur, ce que nous allons r aliser dans la suite de ce chapitre.

# 3 Segmentation d'images par Codebook RGB-D et invariance colorimétrique

Nous avons montr  pr c demment que la m thode de fusion RGB-D employ e dans l'algorithme de d tection des objets mobiles Codebook pr sente un int r t dans certains cas, notamment en environnement int rieur, mais que les tests en environnement ext rieur n'ont pas permis d'am liorer de fa on notable la qualit  des segmentations sur plusieurs s quences successives. Nous proposons alors ici de consid rer l'apport d'invariants colorim triques dans le cadre de cette fusion RGB-D.

Le diagramme 3.6 d crit la strat gie adopt e pour combiner l'utilisation d'invariants colorim - triques avec l'algorithme de segmentation Codebook RGB-D d crit pr c demment dans le cas o la cam ra st r oscopique Bumblebee 2 est utilis e. Nous ne pr sentons pas le m me diagramme pour le capteur Microsoft Kinect puisqu'il s'av re tr s simple du fait qu'il contient une seule image couleur et que l' tape de rectification est directement r alis e par le capteur.

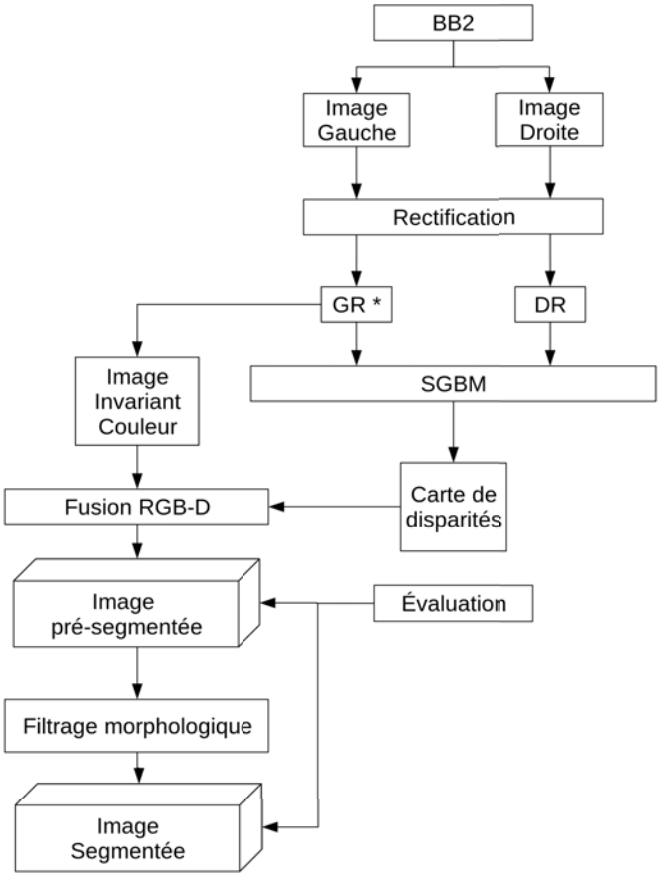



FIGURE 3.6 – Diagramme illustrant la strat gie de fusion Codebook RGB-D avec l'utilisation d'invariants colorim triques (pour la cam ra st r oscopique Bumblebee 2).

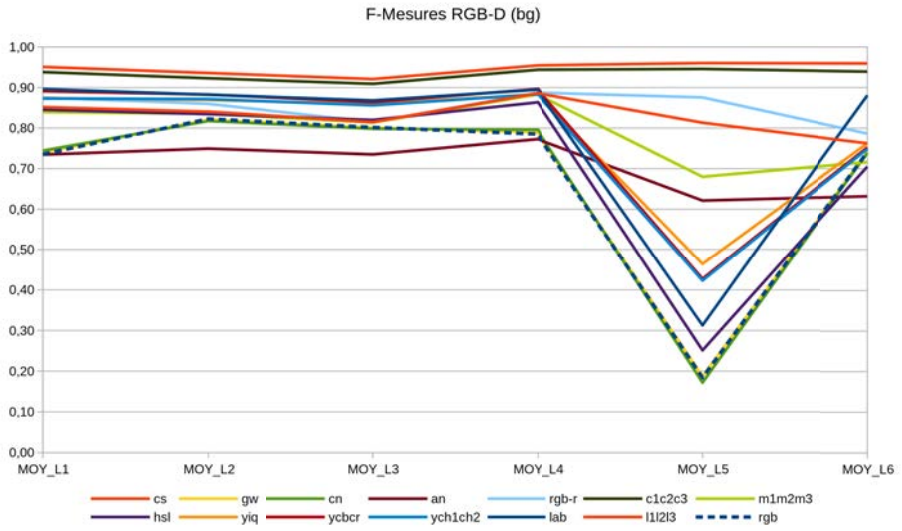
## 3.1 Détection d'objets mobiles sur la base "BUREAU"

D'apr s les figures 3.7a et 3.7b, l'utilisation d'invariants colorim triques montre un r el gain par rapport aux m mes traitements r alis s sans l'utilisation de ces derniers, et ce que ce soit pour le fond (bg) ou pour les objets mobiles (fg).

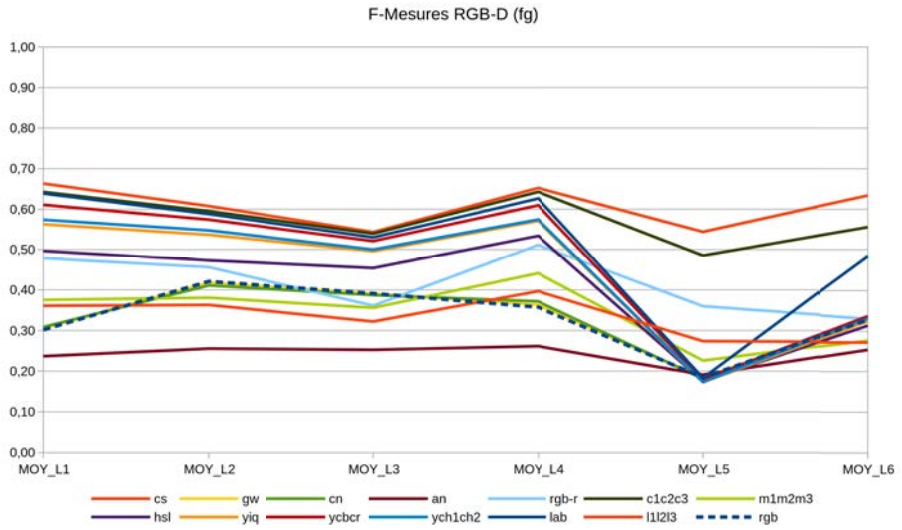
Nous pouvons constater que les invariants colorimétriques Chromaticity Space (*cs*), c1c2c3 ainsi que RGB-Rang (*rgb-r*) et l1l2l3 dans une moindre mesure augmentent significativement la qualité de la segmentation pour les régions du fond (les F-Mesures moyennes sont comprises entre 0.90 et 0.96), quelle que soit la séquence d'apprentissage utilisée. Nous constatons également que ces invariants colorimétriques apportent une stabilité dans les traitements, malgré des changements de luminosité (certes faibles mais existants) en environnement intérieur. En effet, comparé aux mêmes traitements réalisés sans invariant colorimétrique, le traitement de la séquence L5 pourtant très compliquée (de par son manque important de luminosité) affiche pour les deux meilleurs invariants un gain sur la F-Mesure d'environ 0.75.

Concernant les objets mobiles, une observation similaire peut être faite puisque le gain obtenu respectivement avec l'invariant Chromaticity Space (*cs*) et l'espace c1c2c3 est compris entre 0.15 et 0.36 et entre 0.15 et 0.34 selon la séquence d'apprentissage utilisée. Nous pouvons alors conclure que nous obtenons de meilleures segmentations sur le fond et sur les objets mobiles, lorsque nous utilisons une fusion RGB-D au sein de l'algorithme Codebook combinée avec l'utilisation d'invariants colorimétriques. Aussi, nous constatons que Chromaticity Space est l'invariant colorimétrique le plus efficace pour cette base d'images également.

La figure 3.8 illustre des résultats de segmentation pour quelques invariants colorimétriques comparé à l'espace RGB (sans invariant colorimétrique). Il apparaît clairement que le bruit présent en l'arrière-plan est réduit avec l'utilisation des invariants colorimétriques. Pour la séquence la plus complexe (L5), même si le résultat reste grandement améliorable, nous arrivons quand même à distinguer l'objet mobile (la silhouette de la personne) par rapport au fond. Celui-ci reste néanmoins parsemé de nombreuses régions de faux-positifs, mais qui pourront vraisemblablement être supprimés par un filtrage réalisé en post-traitement.




(a) BG




(b) FG

FIGURE 3.7 – F-Mesures moyennes calcul es sur le fond (bg) et les objets mobiles (fg) avec des apprentissages sur les s quences L1, L2, L3, L4, L5 et L6 de la base “BUREAU” avec di rents invariants colorim triques.

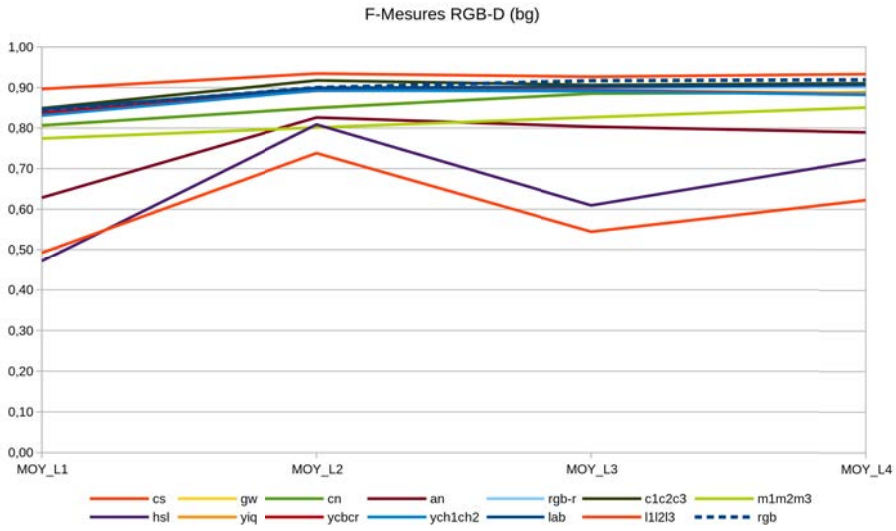
FIGURE 3.8 – R sultats de segmentations obtenues avec l'algorithme Codebook RGB-D avec dif- f rents invariants colorim triques et avec un apprentissage sur la base L2 de la base "BUREAU".

## 3.2 Détection d'objets mobiles sur la base "JARDIN"

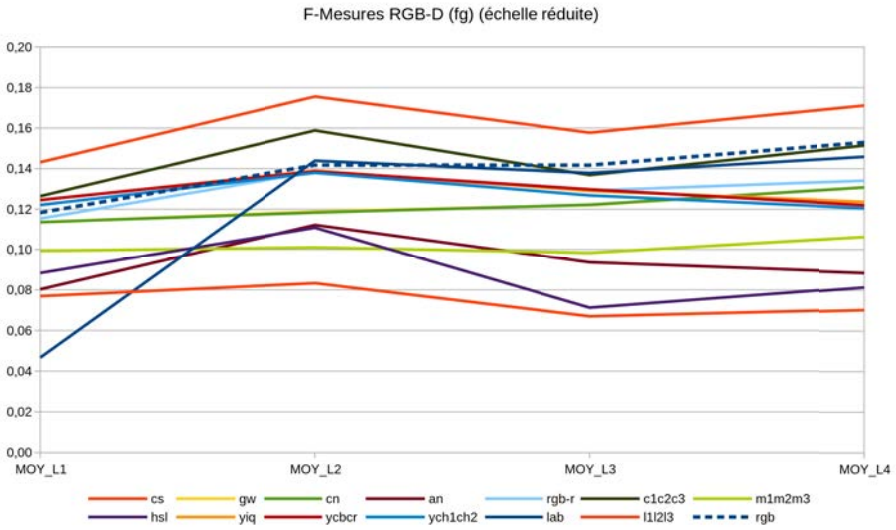
Concernant la base extérieure "JARDIN", nous observons que les différences sur les résultats entre les différents invariants colorimétriques sont, pour les régions du fond comme pour les objets mobiles, beaucoup moins contrastées. Comme l'illustre figure 3.9, seul l'invariant colorimétrique "Chromaticity Space" ($cs$) améliore les résultats de segmentation par rapport à l'utilisation du Codebook RGB-D sans invariant colorimétrique.

Comme précédemment, nous observons une amélioration pour les deux classes, même si elle est très faible : entre 0 et 0.1 pour le fond, et entre +0.02 et +0.04 pour les objets mobiles. Les espaces et invariants colorimétriques c1c2c3, RGB-Rang, YCbCr, YCh1Ch2 et L*a*b* donnent des résultats extrêmement proches (légèrement supérieurs ou inférieurs selon la séquence considérée) de RGB.

La figure 3.10 illustre quelques résultats de segmentation. Nous constatons globalement que l'utilisation d'invariants colorimétriques n'a pas apporté de gain important par rapport à l'utilisation d'un Codebook sans invariant colorimétrique. Le bruit présent dans le fond n'est que très faiblement réduit, mais les objets mobiles restent relativement bien segmentés. Ainsi, si nous n'observons pas d'amélioration notable dans le traitement de cette base d'images, nous n'observons pas non plus de dégradation.



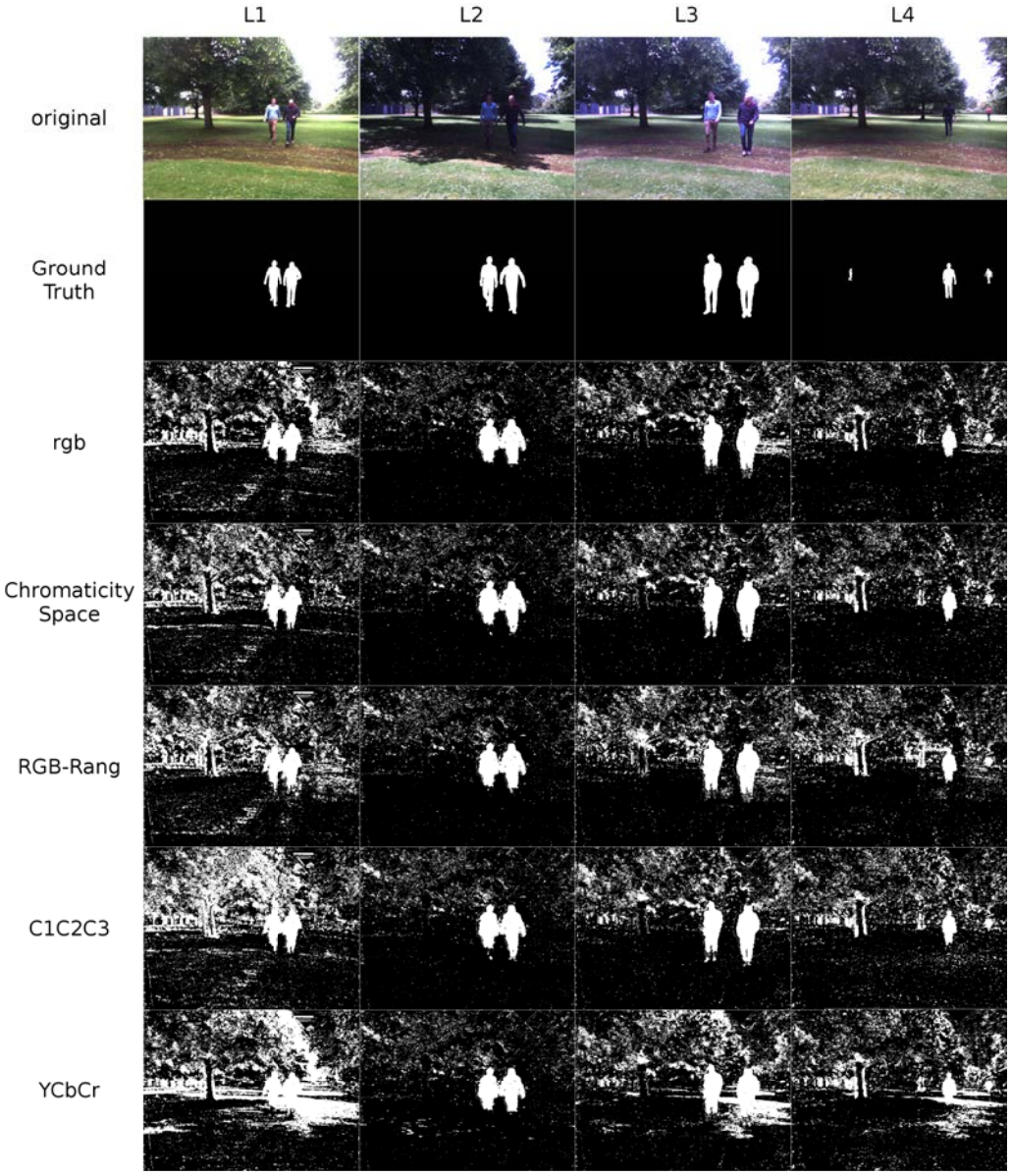

(a) BG




(b) FG

FIGURE 3.9 – F-Mesures moyennes calcul es sur le fond (bg) et les objets mobiles (fg) avec des apprentissages sur les s quences L1, L2, L3 et L4 de la base "JARDIN" et di rents invariants colorim triques.

FIGURE 3.10 – R sultats de segmentations obtenues avec l'algorithme Codebook RGB-D avec di rents invariants colorim triques et avec un apprentissage sur la base L2 de la base "JARDIN".



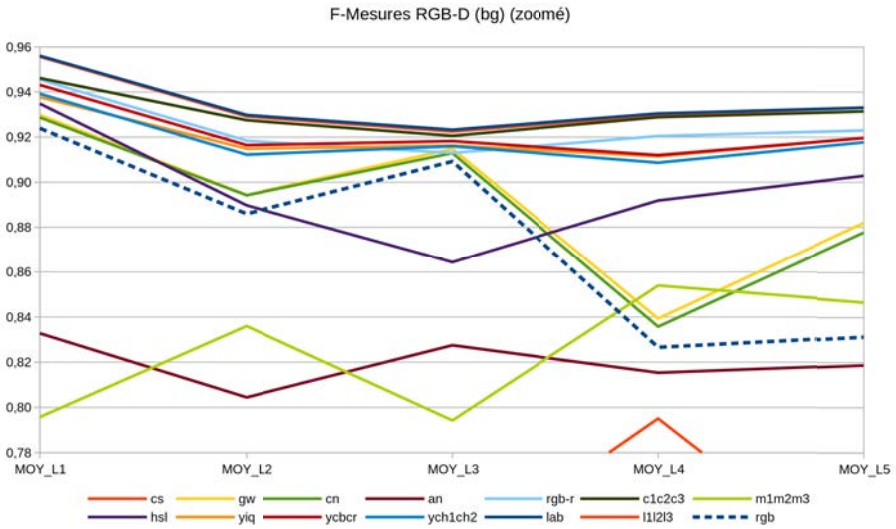
## 3.3 Détection d'objets mobiles sur la base "RUE"

Comme pour la base d'images "JARDIN" précédente, l'utilisation d'invariant colorimétrique apporte en moyenne un gain très faible sur la base "RUE". En effet, comme l'illustre la figure 3.11a, la plupart des invariants colorimétriques apportent une amélioration sur le fond par rapport aux résultats obtenus sans invariant (seuls les espaces couleur HSL, m1m2m3 et Affine Normalization sont en dessous de RGB). Néanmoins, les améliorations apportées par ces invariants colorimétriques restent faibles puisque l'amélioration maximale est comprise entre 0.03 et 0.10. C'est l'espace couleur L*a*b* qui offre les meilleurs résultats dans les régions du fond, suivi des invariants Chromaticity Space ($cs$), RGB-Rang ($rgb$-$r$) et c1c2c3 avec une F-Mesure supérieure à 0.92.

Concernant les régions appartenant aux objets mobiles, le gain est moins important que pour les régions du fond puisqu'il se situe entre 0.01 et 0.06. La figure 3.11b nous permet de montrer que les résultats obtenus avec les invariants colorimétriques Greyworld ($gw$), Comprehensive Normalization ($cn$) et RGB-Rang ($rgb$-$r$) surpassent ceux obtenus sans invariant colorimétrique, et ce quelle que soit la séquence d'apprentissage utilisée. Nous pouvons également noter que les invariants Chromaticity Space ($cs$) et c1c2c3 ne produisent un résultat inférieur à RGB que dans le cas où la séquence d'apprentissage est $L3$. Cela corrobore l'hypothèse que le choix de la séquence d'apprentissage a effectivement une importance considérable dans l'obtention du meilleur résultat.

La figure 3.12 illustre de manière plus visuelle quelques résultats de segmentation. Ainsi, nous pouvons observer que les invariants colorimétriques ne se trouvant pas dans le haut du classement dégradent la détection, ce qui n'est pas le cas avec RGB-Rang (par exemple les jambes de la silhouette ne sont plus détectées avec certains invariants colorimétriques). Malgré cette dégradation, nous remarquons une légère diminution du bruit dans le fond, notamment pour la séquence $L5$, réputée difficile. Ceci est en revanche moins visible pour les autres séquences, et particulièrement pour $L1$, où le bruit reste fortement présent dans toute l'image. La cause de ce bruit est probablement causée par la faible qualité des cartes de disparités calculées à partir des images gauche et droite, fortement bruitées en raison du manque de luminosité.

(a) F-Mesures du fond (bg).




(b) F-Mesures des objets mobiles (fg).

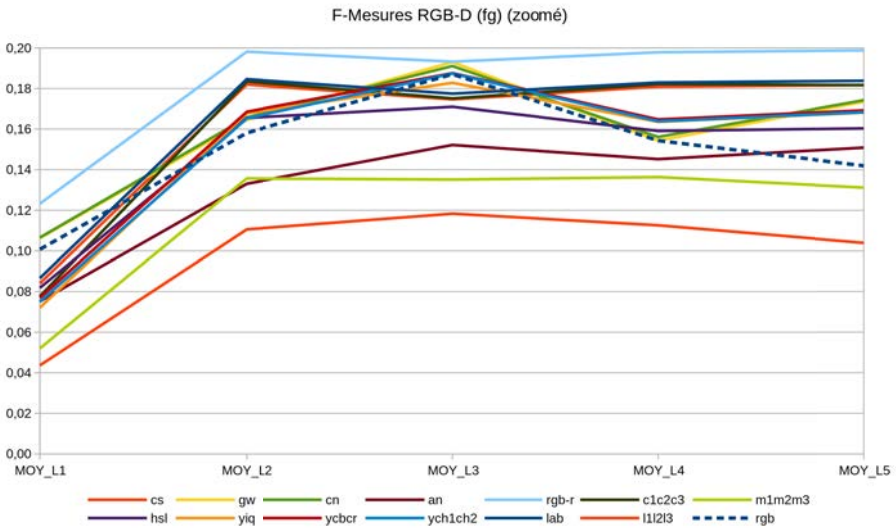
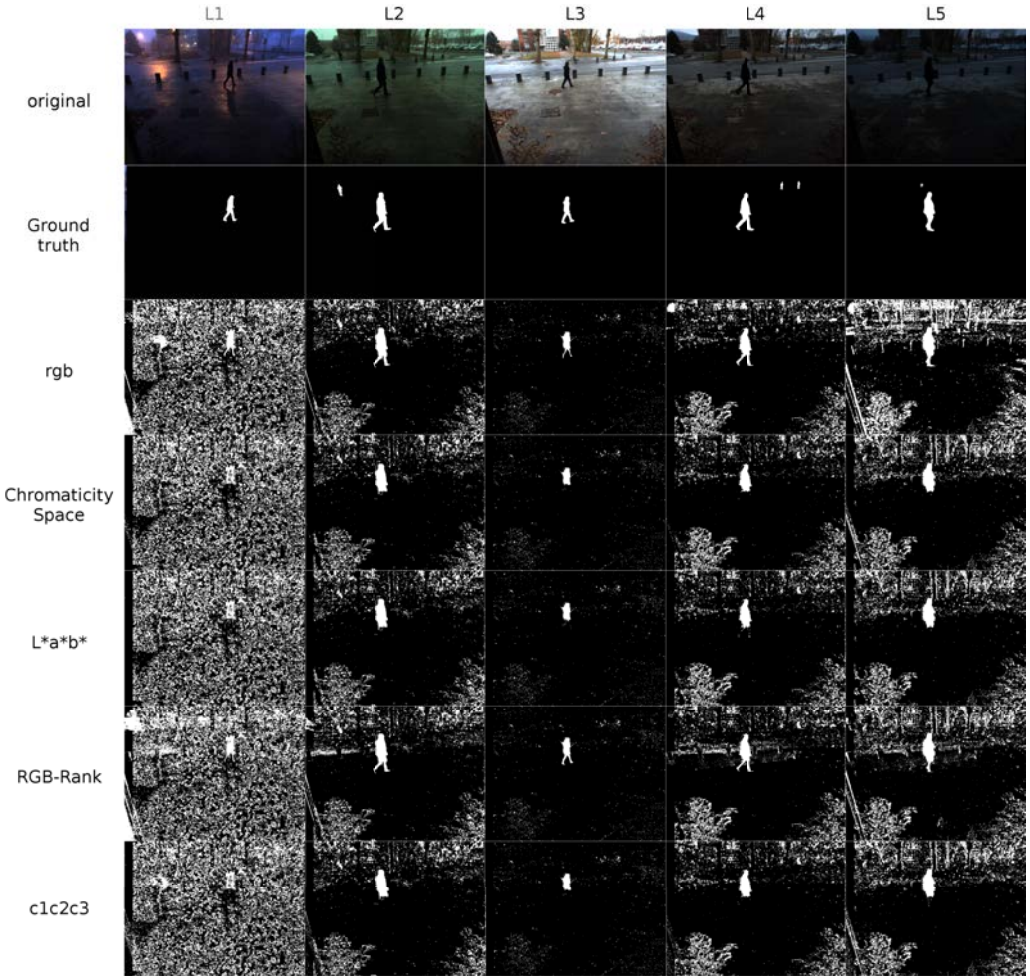



FIGURE 3.11 – F-Mesures calcul es sur le fond (bg) et les objets mobiles (fg) avec des apprentissages sur les s quences L1, L2, L3, L4 et L5 avec di rents invariants colorim triques pour la base “RUE”.

FIGURE 3.12 – R sultats de segmentations obtenus avec l'algorithme Codebook RGB-D avec dif- f rents invariants colorim triques et avec un apprentissage sur la base L3 de la base "RUE".



## 3.4 Conclusion

La méthode Codebook RGB-D (intégrant une information de couleur, d'intensité et de profondeur) couplée à l'utilisation d'invariants colorimétriques a démontré son efficacité pour la détection d'objets mobiles en environnement intérieur et dans une moindre mesure, en environnement extérieur.

Les résultats obtenus ici confirment les résultats obtenus dans le chapitre précédent sur l'apport des invariants colorimétriques. L'information de profondeur ajoutée au Codebook a elle aussi en moyenne apporté une amélioration des résultats même s'il convient de réfléchir davantage sur la méthode de fusion.

Cependant, nous relevons plusieurs points importants :

— Sur les trois bases d'images testées, au moins un des invariants colorimétriques testés obtient de meilleurs résultats que l'algorithme Codebook utilisé sans invariant.

— L'invariant colorimétrique Chromaticity Space a donné les meilleurs résultats pour deux bases d'images sur trois. L'invariant RGB-Rang obtient avec Chromaticity Space de bonnes performances sur la base "RUE", mais RGB-Rang s'avère légèrement meilleur que Chromaticity Space pour les objets mobiles.

— Pour des cas relativement simples, à savoir en environnement intérieur avec l'utilisation de la caméra Microsoft Kinect, l'amélioration est sensible et comprise en moyenne entre 0.08 et 0.75 selon la séquence d'apprentissage utilisée.

— Pour des cas plus complexes, à savoir des environnements extérieurs comme la base "RUE qui présente des séquences aux aspects colorimétriques relativement différents les uns des autres, l'amélioration est présente mais malgré tout assez faible, la F-Mesure étant comprise en moyenne entre 0.02 et 0.10.

# 4 Segmentation d'images par Codebook RGB-D flou

La fusion des informations de couleur et profondeur dans la méthode Codebook permet d'améliorer la qualité de la segmentation des objets mobiles. L'amélioration se situe au niveau des objets mobiles qui sont généralement mieux segmentés car moins "troués" lorsque leur couleur est proche de celle du fond. A contrario, l'amélioration est plus nuancée pour les régions du au fond. En effet, nous observons toujours du bruit et de nombreux faux-positifs dans ces régions lors de changements d'illumination et de couleur dans les images constituant la base.

Plusieurs solutions peuvent être envisagées pour contrebalancer ce constat :

— Modifier la durée de l'apprentissage : cette solution permet de prendre en compte un plus grand nombre d'états différents pour chaque pixel (en termes de couleur et d'intensité lumineuse, la disparité d'un pixel étant invariante à ces deux paramètres. Néanmoins, l'obtention d'une disparité valide est justement rendue difficile pour les images acquises dans des conditions d'illumination faible, ce qui génère un bruit important. Cependant, la période d'apprentissage étant généralement fixée à priori, il est difficile de prévoir la durée idéale pour que des changements d'illumination intéressants soient présents dans le modèle d'apprentissage.

— Superviser l'apprentissage : un apprentissage utilisant $N$ frames sans qu'elles soient contrôlées peut amener du bruit, même si le modèle d'arrière-plan est ensuite raffiné. C'est pourquoi un apprentissage mixte peut être envisagé : cela consiste à effectuer une répartition non chronologique mais régulière de $N$ frames obtenues sur une plus longue durée (une journée complète, par exemple) ou en plusieurs séquences. Il est d'ailleurs recommandé que les frames ne soient pas intégrées chronologiquement en raison du paramètre $\lambda$ (MNRL) qui correspond à la plus longue durée de non-observation du codeword. En effet, si les frames d'apprentissage sont ordonnées chronologiquement, alors les frames apprises en premier ne seront pas observées pendant une longue durée à la fin de cet apprentissage, et les codewords issus de ces frames risquent alors d'être filtrés.

— Influencer la mise à jour du modèle : plusieurs améliorations peuvent être apportées pour prendre en compte les changements temporaires dans les images. La principale amélioration consiste à utiliser un ou plusieurs modèles d'arrière-plan temporaires en plus du modèle d'arrière-plan principal. Une correspondance est d'abord recherchée dans le modèle principal ; en cas d'échec, la même recherche est effectuée dans le modèle temporaire. Lorsqu'une correspondance existe dans ce modèle temporaire, le codeword apparié est mis à jour puis intégré dans le modèle principal lorsqu'il a été observé un nombre suffisant de fois. Ce système est facile à mettre en œuvre, mais sa réactivité n'est pas suffisante dans le cadre de notre application et notamment à cause des changements d'illumination et de couleur réguliers de la scène. Il est en revanche adapté aux objets initialement mobiles qui s'immobilisent pendant un moment comme par exemple, une voiture qui se gare ou un sac qu'on abandonne au sol.

— Modifier la façon de gérer les paramètres de l'algorithme : l'algorithme Codebook utilise un certain nombre de seuils afin de déterminer une correspondance entre les couleurs et intensités du pixel observé et celles de l'un des codewords présents dans le modèle d'apprentissage. Ces seuils sont soit fixés ($\epsilon$), soit calculés à partir de valeurs intégrées au codeword testé ($B_{low}$, $B_{hi}$) en les multipliant par des termes qui sont également des paramètres fixés ($\alpha$, $\beta$). Une première solution consiste à modifier dynamiquement ces paramètres. Il est possible que cette solution implique la gestion de nombreuses valeurs différentes de ces paramètres, c'est-à-dire au moins une par pixel. Il serait envisageable de limiter ce nombre de valeurs à plusieurs

régions de l'image. Cette solution n'a pas été explorée dans cette thèse mais peut faire l'objet de perspectives à ces travaux. Une autre solution consiste à modifier les conditions qui valident la correspondance entre un pixel observé et un codeword du modèle d'arrière-plan. C'est cette seconde idée que nous avons retenue et que nous allons détailler ci-après.

Revenons rapidement sur un point important de l'algorithme du Codebook. Celui-ci consiste à déterminer si un pixel observé peut être mis en correspondance avec un codeword du modèle d'arrière-plan. L'existence d'une telle correspondance signifie que le pixel observé ressemble suffisamment à un pixel observé précédemment, ce qui permet de le marquer comme appartenant au fond. Cette correspondance est déterminée à partir de deux conditions : la distorsion de couleur et la distorsion d'intensité du pixel. En fonctionnement RGB-D, une troisième condition est ajoutée : la distorsion de disparité. Au finale, ces deux (ou trois) conditions doivent être validées pour que la correspondance soit établie.

La distorsion de couleur consiste à calculer l'écart entre la couleur du pixel observé et celle du codeword. Si cet écart est inférieur à un seuil donné $\epsilon$, alors la condition sur la distorsion de couleur est validée. Néanmoins, il arrive fréquemment que cette distorsion dépasse le seuil fixé, et même si ce dépassement est très faible, il empêche la correspondance. Cette augmentation de distorsion peut être dûe à un changement momentané de l'illumination de la scène. Il peut être plus ou moins durable, mais dans les deux cas, la correspondance avec le modèle n'étant pas possible, les pixels impactés seront considérés comme des objets mobiles.

De même, la distorsion d'intensité utilise les valeurs d'intensité minimum et maximum du codeword testé, multipliés par deux paramètres $\alpha$ et $\beta$ afin d'obtenir un intervalle $[B_{min}, B_{max}]$ dans lequel doit se trouver l'intensité du pixel testé pour que la correspondance soit faite. Comme précédemment, l'intensité d'un pixel donné peut évoluer en fonction des changements d'illumination, et dans le cas de sur-illumination, la valeur d'intensité du pixel peut sortir de l'intervalle déterminé.

La distorsion de disparité est quant à elle peu impactée par les changements d'illumination de la scène. En revanche, la méthode de mise en correspondance des pixels des frames gauche et droite peut l'être, surtout si elle se base uniquement sur la couleur. Nous pensons que cela impacte la qualité des cartes de disparités, mais en globalité les valeurs de disparités qu'elles contiennent restent néanmoins relativement constantes, et nous supposons donc que cette information est invariante.

Dans ce que nous avons vu précédemment, la fusion de ces trois conditions est effectuée de façon booléenne, vérifiant la validité des trois conditions précédentes. Au vu des résultats, nous estimons que cette méthode de fusion n'est pas suffisamment permissive. Elle interdit les écarts de distorsions trop importants causés par les changements d'illumination, et produit d'importants taux de faux positifs dans les segmentations. Nous formulons alors l'hypothèse qu'une fusion de ces trois conditions basée sur la logique floue permettrait de prendre une décision plus souple sur la classification finale des pixels observés (objet mobile ou arrière-plan). Nous allons donc présenter ci-après une nouvelle méthode de fusion d'informations basée sur la logique floue pouvant être intégrée à l'algorithme Codebook.

## 4.1 Principe de la logique floue

La logique floue est un outil mathématique s'opposant à la logique modale. Elle s'applique dans un grand nombre de domaines tels que l'automatique, la robotique ou encore la météorologie. Elle peut également être employée dans les problématiques d'aide à la prise de décision et d'intelligence artificielle. Son principe consiste à autoriser la formulation d'états différents du couple booléen

classique vrai/faux : des concepts incertains tels que petit/moyen/grand, clair/sombre, ou encore froid/tiède/chaud permettent de définir un ensemble de règles définissant un système que l'on souhaite contrôler.

Par exemple, un système d'éclairage automatique se basant sur un capteur de luminosité doit permettre d'éclairer plus ou moins fort en fonction de la valeur renvoyée par ce capteur. L'algèbre booléenne permet de résoudre ce problème en discrétisant le système par la définition d'états finis : en-dessous d'un certain seuil de luminosité $\gamma$, une puissance $P_{high}$ est appliquée à la lampe ; au-dessus, une puissance $P_{low}$. Dans un cas où la luminosité déterminée par le capteur est tout juste inférieure au seuil fixé, est-il nécessaire d'appliquer le même éclairage que si cette luminosité était clairement inférieure au seuil fixé ?

Ce système décisionnel peut être décrit par un ensemble de règles floues qui permettent de décrire une décision à prendre. Dans l'exemple précédent :

— SI la luminosité est faible ALORS Puissance électrique maximale

— SI la luminosité est moyenne ALORS Puissance électrique moyenne

— SI la luminosité est forte ALORS Puissance électrique faible

— SI la luminosité est très forte ALORS Puissance nulle

La luminosité est l'ensemble de référence (ou variable floue d'entrée), sur lequel on définit 4 sous-ensembles flous (valeurs de la variable floue d'entrée) : faible/moyenne/forte/très forte. A chacun de ces sous-ensembles flous est associée une fonction d'appartenance qui détermine le degré d'appartenance (dans l'intervalle $[0, 1]$) des valeurs de luminosité au sous-ensemble flou, comme l'illustre la figure 3.13. Contrairement à la logique booléenne, il n'existe pas de "SINON" dans la définition de ces règles. En effet, toutes les règles sont évaluées car plusieurs d'entre elles peuvent être vraies en même temps, chacune à un degré différent. Ainsi, une mesure peut appartenir à plusieurs sous-ensembles flous à la fois, mais à des degrés d'appartenance différents ou identiques. Ces degrés d'appartenance ne sont pas des probabilités. On utilise habituellement les fonctions d'appartenance triangulaires et/ou trapézoïdales, pour des raisons de simplicité.

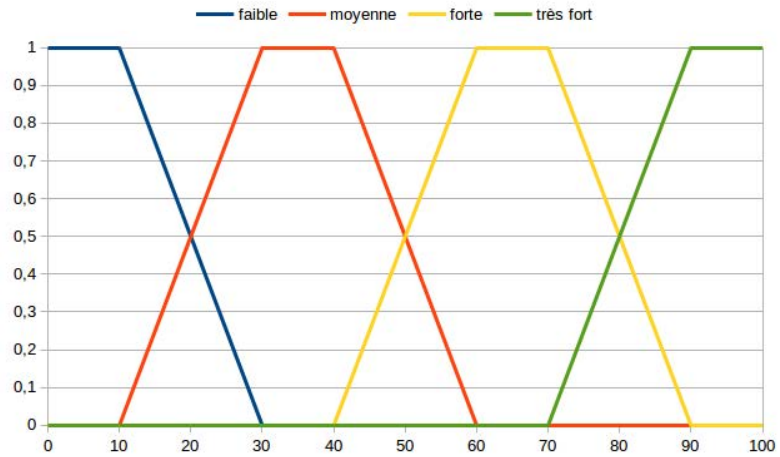



FIGURE 3.13 – Fonctions d'appartenance des 4 sous-ensembles flous

Plusieurs opérateurs flous, similaires à ceux de la logique booléenne, peuvent être définis pour la liaison de plusieurs sous-ensembles au sein d'une même règle : l'opérateur NON est habituellement défini comme le complément, l'opérateur ET par le minimum et l'opérateur OU par le maximum.

Chaque action (ou conclusion de règle) du système précédemment décrit est considéré comme un sous-ensemble flou d'un ensemble de référence de sortie (ou variable floue de sortie) Puissance. Comme pour l'entrée, les sous-ensembles flous de la variable floue de sortie sont décrits par des fonctions d'appartenance. La figure 3.14 illustre pour cet exemple les quatre fonctions d'appartenance.

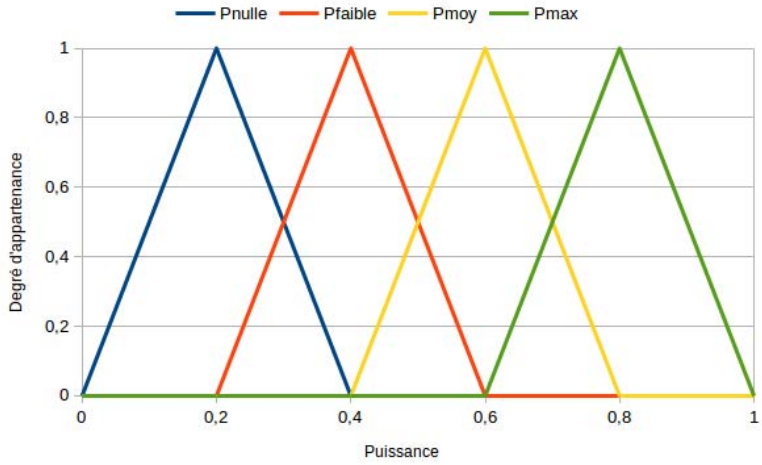



FIGURE 3.14 – Fonctions d'appartenance des 4 conclusions du système flou.

Une fois le système défini, l'analyse d'une observation est effectuée en 3 étapes illustrées par la figure 3.15.

1. Fuzzification : passage d'une mesure exacte vers une représentation floue par l'intermédiaire des fonctions d'appartenance.
2. Inférence floue : combinaison de l'ensemble des règles décrivant le processus de reconnaissance, définies précédemment.
3. Défuzzification : processus inverse de la fuzzification. Déterminaison d'une valeur numérique à partir de la fonction résultante, issue de l'inférence floue.

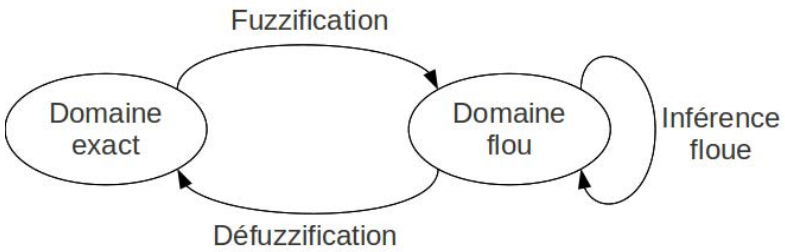



FIGURE 3.15 – Étapes d'un processus d'analyse par logique floue

L'inférence floue consiste à évaluer le degré de validité de chaque règle à l'aide des fonctions d'appartenance pour la ou les variables d'entrée (dans notre exemple, la luminosité mesurée). Il en résulte une nouvelle fonction d'appartenance pour l'action associée à chaque règle. Les nouvelles fonctions d'appartenance sont ensuite combinées pour calculer une fonction résultante. Plusieurs méthodes d'inférence sont utilisables : maximum/minimum, maximum/produit, somme/produit,

entre autres. Chaque m thode d crit la fa on d' valuer les r gles en faisant correspondre un op - rateur math matique chaque op rateur bool en de la r gle, ainsi que le passage du ALORS (condition vers la conclusion d'une r gle) et la combinaison entre les r gles. Le tableau 3.5 r capitule ces trois di rentes m thodes.

TABLE 3.5 – Tableau r capitulatif de quelques op rateurs d'inf rence.

| M thode | Condition ET | Condition OU | ALORS | Combinaison des r gles |
|---|---|---|---|---|
| max/min | min | max | min | max |
| max/produit | min | max | produit | max |
| somme/produit | produit | moyenne | produit | moyenne |

Supposons une mesure de luminosit de 55 : l'inf rence permet de calculer les deux nouvelles fonctions d'appartenance pour les $2^{ème}$ et $3^{ème}$ r gles comme l'illustre la figure 3.16. Les $1^{ère}$ et $4^{ème}$ r gles ne sont pas repr sent es car cette valeur d'inf rence ne valide aucune de ces deux r gles. Le graphique de droite de la figure 3.16 repr sente le r sultat de l'inf rence floue e ectu e sur les

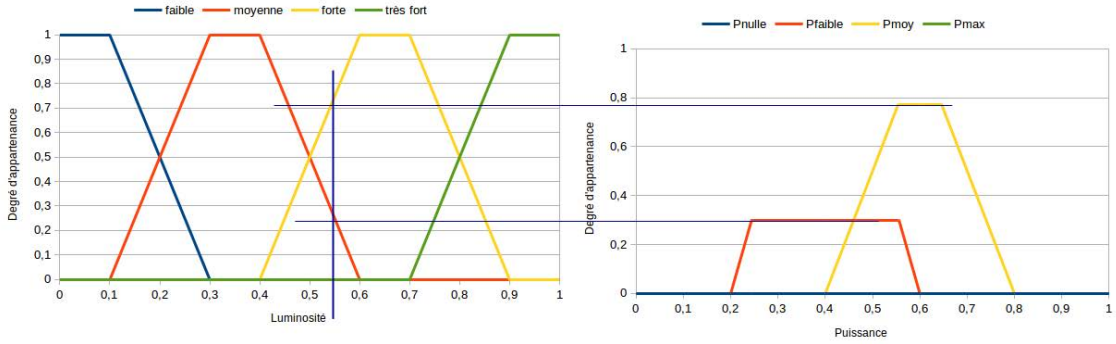



FIGURE 3.16 – Inf rence floue des quatre r gles pour une valeur de luminosit de 55.

fonctions r sultat illustr es par la figure 3.14. Il est n cessaire de d fuzzifier ce r sultat pour en d terminer une valeur num rique utilisable.

La d fuzzification est la derni re tape de l'analyse d'une observation au sein d'un syst me flou. Elle consiste d terminer une valeur num rique partir de la fonction r sultante obtenue l'issue de l'inf rence. L encore, plusieurs m thodes existent : la plus simple consiste retenir la valeur correspondant au maximum de la fonction r sultante. Si la fonction r sultante est maximum sur un intervalle, on garde la valeur m diane de cet intervalle. Une autre m thode, et qui plus est la plus utilis e, consiste retenir la valeur correspondant au centre de gravit de la surface d finie par la fonction r sultante et son domaine de d finition. Soit $f_{RES}$ la fonction r sultante d finie sur $D$ et $X^*$ le centre de gravit de la surface $S$ comprise sous la courbe de cette fonction. Analytiquement, on a :

$$X^* = \frac{\int_D X f_{RES}(X) dx}{\int_D f_{RES}(X) dx} \tag{3.14}$$

Il est galement possible de d composer la surface $S$ en plusieurs surfaces afin de simplifier les calculs de la surface comprise sous la courbe. Dans l'exemple pr c dent, le centre de gravit de la surface d finie par la fonction r sultante se situe environ 0.6.

## 4.2 Fusion RGB-D par Codebook et logique floue

Après avoir présenté le concept de Logique Floue, nous allons appliquer un principe similaire à la méthode de fusion des informations couleur, intensité et disparité pour la détection fond/forme à l'aide du modèle obtenu par la méthode Codebook. Nous rappelons que celle-ci effectue une décomposition de l'information pixellaire et la compare aux éléments présents dans le modèle d'arrière-plan obtenu après apprentissage, sur deux critères : distorsion de couleur et distorsion d'intensité. Une troisième information que nous avons souhaité prendre en compte est la disparité associée à chaque pixel après le calcul d'une carte de disparités.

Nous exprimons maintenant la fusion de ces informations par un ensemble de règles définissant un système flou, basé sur les mêmes paramètres que ceux de l'algorithme Codebook original. Contrairement à la couleur, les valeurs d'intensité et de disparité sont des informations unidimensionnelles. Il n'est donc pas nécessaire de considérer une valeur de distorsion (obtenue nécessairement par comparaison avec les valeurs internes d'un codeword). Il serait envisageable de considérer les trois composantes R, G et B séparément dans le système flou, mais pour des raisons d'optimisation, nous utilisions la distorsion de couleur obtenue en comparant la couleur d'un pixel avec celle d'un codebook.

### Définition des ensembles de référence et sous-ensembles flous

Dans le cadre de notre application, nous définissons un ensemble de référence pour chacune des trois variables (Distorsion de couleur, Intensité et Disparité) ainsi que les 2 ou 5 sous-ensembles qui les caractérisent, que nous présentons dans le tableau 3.6.

TABLE 3.6 – Tableau des ensembles et sous-ensembles flous définis.

| **Ensemble de référence** | **Sous ensemble 1** | **Sous ensemble 2** | **Sous ensemble 3** | **Sous ensemble 4** | **Sous ensemble 5** |
|---|---|---|---|---|---|
| Distorsion de couleur | Faible | Forte | | | |
| Intensité | Très inférieure | Inférieure | Similaire | Supérieure | Très supérieure |
| Disparité | Très rapprochée | Rapprochée | Équidistante | Éloignée | Très éloignée |

Dans notre cas, la distorsion de couleur prend des valeurs comprises entre 0 (couleurs identiques) et environ 442 (opposition entre le blanc et le noir). Il en est de même pour l'intensité calculée par $B = \sqrt{R^2 + G^2 + B^2}$ et qui prend des valeurs comprises entre 0 (intensité nulle, pixel noir) et 442 (intensité d'un pixel blanc). La disparité quant à elle prend des valeurs comprises entre 0 (proche de la caméra) et 255 (très éloignée de la caméra).

La similarité de couleur entre un pixel et un codeword étant mesurée par la distorsion de couleur $\delta$, le seuil $\epsilon$ est utilisé dans l'algorithme du Codebook, définissant que si $\delta \leq \epsilon$, alors la couleur du pixel est considérée comme étant proche de celle du codeword. La distorsion de couleur étant une mesure à minimiser, un seul $\epsilon$ est utilisé pour déterminer les deux sous-ensembles flous : Faible et Forte, comme l'illustre la figure 3.17.

L'ensemble de référence définissant l'intensité est caractérisé par cinq sous-ensembles flous : Très inférieure, Inférieure, Similaire, Supérieure et Très supérieure. Les seuils employés pour la définition de ces cinq sous-ensembles sont déterminés à l'aide des valeurs $B_{min}$ et $B_{max}$ provenant du codeword auquel on souhaite apparier le pixel observé d'intensité $B$. Pour élargir un peu cet intervalle $[B_{min}; B_{max}]$ et leur donner une taille minimale $E_{minimum}$, il est possible de définir un

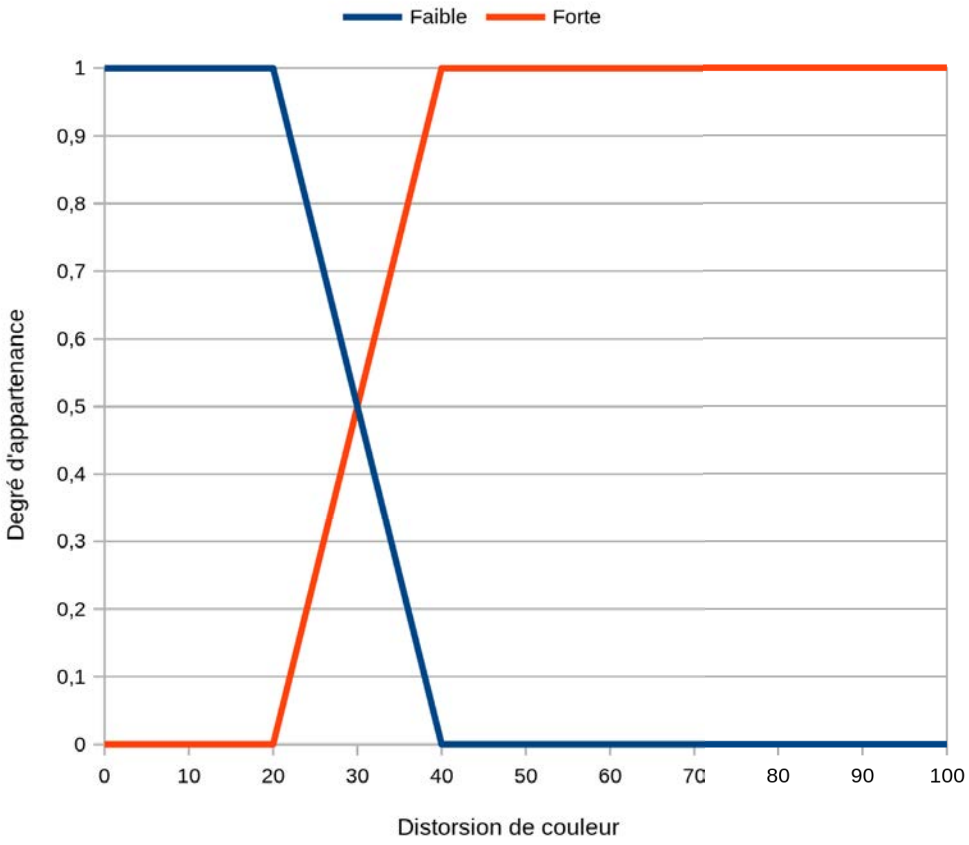



FIGURE 3.17 – Description de l'ensemble de r f rence Distorsion de couleur.

nouvel intervalle $[B'_{min}; B'_{max}]$ tel que $B'_{min} = B_{min} - E$ et $B'_{max} = B_{max} + E$ o $E$ est donn par l' quation 3.15.

$$E = \begin{cases} E_{minimum} + (B_{min} - B_{max})/2 & \text{si } B_{max} - B_{min} < E_{minimum} \\ 0 & \text{sinon} \end{cases} \tag{3.15}$$

Le sous-ensemble flou Similaire est d fini sous la forme d'une fonction d'appartenance trap - zo dale dont les seuils forment l'intervalle $[B^{Sim}_{low}; B^{Sim}_{high}]$ d termin par l' quation 3.16. La fonction d'appartenance associ e ce sous-ensemble flou renvoie 1 si $B \in [B^{Sim}_{low}; B^{Sim}_{high}]$.

$$\begin{aligned} B^{Sim}_{low} &= B_{min} - \frac{B_{min} - (B_{min} - \gamma)}{2}, \\ B^{Sim}_{high} &= B_{max} + \frac{B_{max} - (B_{max} - \gamma)}{2} \end{aligned} \tag{3.16}$$

avec $\gamma$ qui correspond une valeur d finissant l' cartement des sous-ensembles flous Inf rieur et Sup rieur par rapport l'intervalle $[B'_{min}, B'_{max}]$.

Les deux sous-ensembles Inf rieur et Sup rieur sont d finis par des fonctions triangulaires dans des intervalles form s par des seuils calcul s respectivement selon l' quation 3.17 pour le sous-

ensemble Inf rieur, et selon l' quation 3.18 pour le sous-ensemble Sup rieur.

$$\begin{aligned} B_{low}^{Inf} &= B_{min} - \gamma, \\ B_{high}^{Inf} &= B_{min} \end{aligned} \tag{3.17}$$

$$\begin{aligned} B_{low}^{Sup} &= B_{max}, \\ B_{high}^{Sup} &= B_{max} + \gamma \end{aligned} \tag{3.18}$$

Enfin, les deux sous-ensembles flous Tr s inf rieur et Tr s sup rieur qui d limitent les valeurs consid r es comme tr s loign es de l'intervalle de similarit , sont d finis par deux fonctions d'appartenance trap zo dales, dont le r sultat vaut 1 respectivement dans les intervalles $[0; B_{low}^{Inf}]$ et $[B_{high}^{Sim}; 442]$.

Une repr sentation possible de cet ensemble de r f rence Intensit est illustr e sur la figure 3.18. Comme les intervalles sont d crits par les valeurs d'intensit minimum et maximum de chaque codeword, la taille des sous-ensembles flous ainsi que les carts entre eux sont variables selon le pixel consid r et d pendent de l' tat du codeword test .

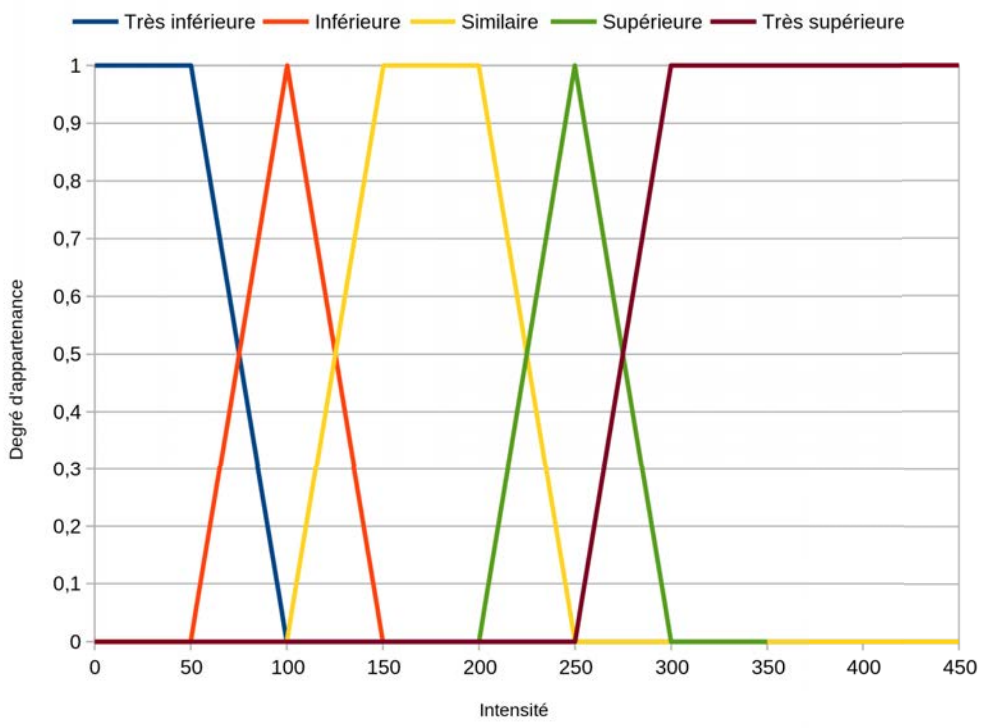



FIGURE 3.18 – Description de l'ensemble de r f rence Intensit .

Enfin, l'ensemble de r f rence d finissant la disparit est caract ris par cinq sous-ensembles flous Tr s rapproch e, quidistante, loign e, Tr s loign e. Ils sont d finis de la m me mani re que les sous-ensembles caract risant l'intensit , mais l'aide des valeurs $D_{min}$ et $D_{max}$ issues du codeword que l'on souhaite apparier avec un pixel de disparit $D$. Ici l' cartement entre les sous-ensembles flous Rapproch e, quidistante et loign e est d pendant de la valeur de disparit maximale des cartes de disparit s calcul es pr c demment. Cet ensemble de r f rence Disparit est

illustr sur la figure 3.19.

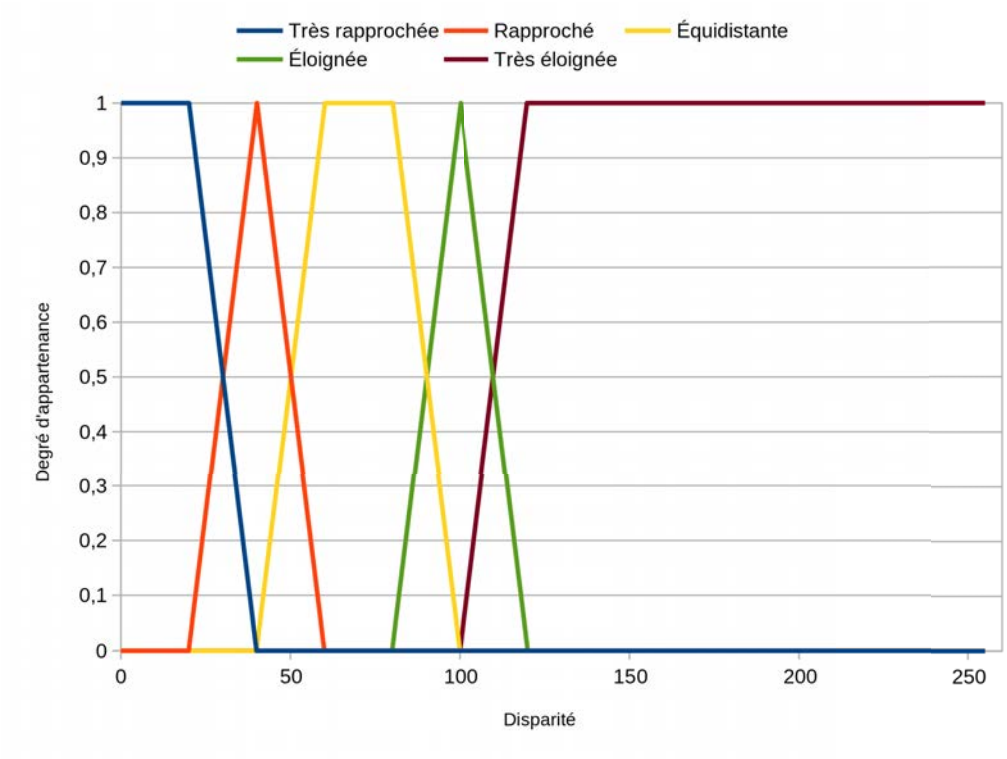



FIGURE 3.19 – Description de l'ensemble de r f rence Disparit .

Pour finir, il est n cessaire de d finir la ou les variables de sortie du syst me flou. Ce syst me permet de d terminer une correspondance entre un pixel observ et un codeword issu du codebook de ce pixel. C'est cette correspondance que nous cherchons caract riser. La di cult de ce syst me est que le r sultat est n cessairement une correspondance ou une non-correspondance, soit une sortie binaire, ce qui ne correspond pas la nature floue souhait e pour ce syst me. Ainsi, nous proposons de caract riser la sortie du syst me par trois sous-ensembles flous d finissant la confiance envers l'existence d'une correspondance entre le pixel observ et le codeword test . Ces trois sous-ensembles sont d finis par les variables : peu probable, probable et fortement probable comme sur figure 3.20.

### Définition des règles d'inférence floue

Pour certains pixels, la valeur de disparit est nulle, ce qui indique que la m thode de mise en correspondance n'a pas t en mesure d'apparier un pixel de l'image gauche avec un pixel de l'image droite. Dans un tel cas, la disparit ne devrait pas tre prise en compte comme entr e du syst me flou, ce qui revient d finir un second syst me flou n'employant comme valeur d'entr e que la distorsion de couleur et l'intensit du pixel. Les r gles d'inf rence de chacun de ces syst mes sont forc ment di rentes puisque l'inf rence doit se faire sur un nombre de variables di rent. Ainsi, l'emploi d'un ensemble de r gles ou d'un autre d pend simplement de la valeur de disparit du pixel test .

Le premier ensemble de r gles concerne l'utilisation de l'information d'intensit et de distorsion de couleur.

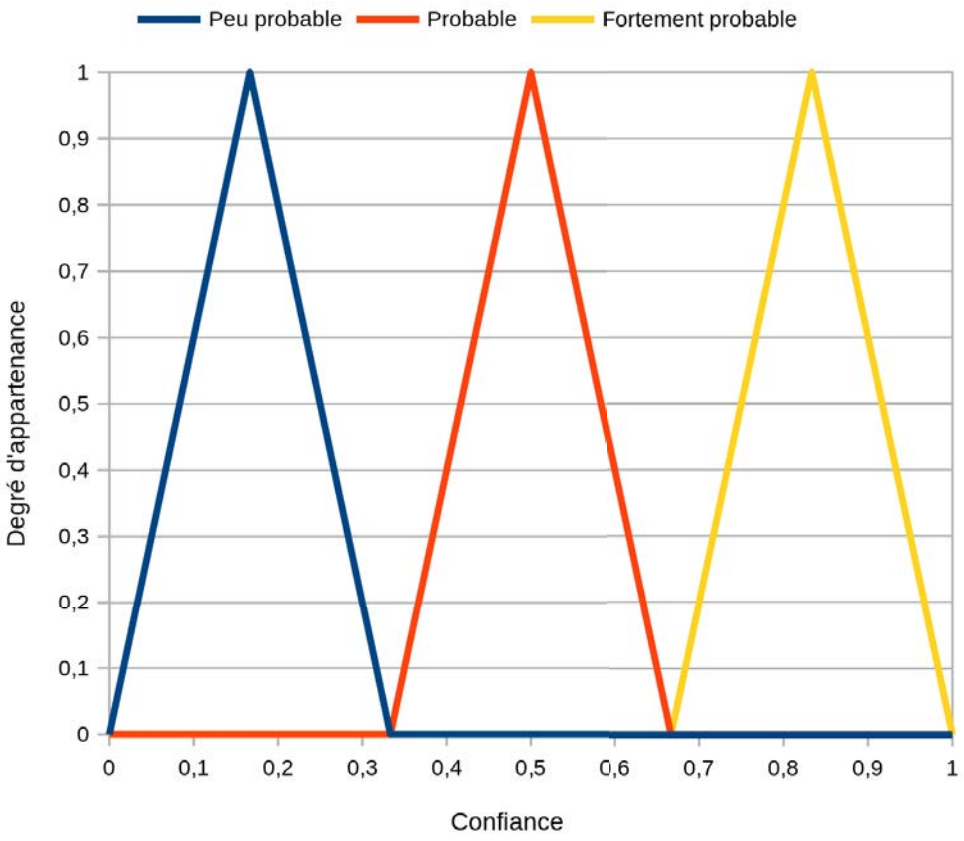



FIGURE 3.20 – Description de la variable de sortie du syst me flou.

— SI distorsion de couleur est forte ET intensit n'est PAS similaire ALORS correspondance est peu probable.

— SI distorsion de couleur est faible ET intensit est inf rieure OU intensit est sup rieure ALORS correspondance est probable.

— SI distorsion de couleur est forte ET intensit est tr s inf rieure OU intensit est tr s sup - rieure ALORS correspondance est probable.

— SI distorsion de couleur est faible ET intensit n'est PAS tr s inf rieure OU intensit n'est PAS tr s sup rieure ALORS correspondance est fortement probable.

Le second ensemble ajoute une troisi me variable : la distorsion de disparit . Lors de l' valuation d'un pixel par le syst me flou d fini pr c demment, ces r gles ne sont utilis es que si la disparit du pixel est di rente de 0. Si cette disparit est gale 0, alors elle est consid r e comme invalide et l'ensemble de r gles pr c demment nonc est alors utilis .

— SI distorsion de disparit est tr s loign ou tr s rapproch ALORS correspondance est faible.

— SI distorsion de disparit n'est PAS tr s loign ou tr s rapproch ET distorsion de couleur est forte ET intensit est tr s inf rieure ou tr s sup rieure ALORS correspondance est peu probable.

— SI distorsion de disparit n'est PAS tr s loign ou tr s rapproch ET distorsion de couleur est forte ET intensit n'est PAS tr s inf rieure ou tr s sup rieure ALORS correspondance est probable.

— SI distorsion de disparité n'est PAS très éloigné ou très rapproché ET distorsion de couleur est faible ET intensité n'est PAS très inférieure ou très supérieure ALORS correspondance est fortement probable.

Pour des raisons de performances, nous avons choisi d'employer la méthode d'inférence max/min (rappelée dans le tableau 3.5), plus économe en ressources. La méthode de défuzzification employée ici est celle du centre de gravité.

## 4.3 Détection d'objets mobiles par Codebook RGB flou

Dans cette section, nous présentons les résultats de la détection d'objets mobiles obtenus par la méthode de fusion floue décrite précédemment. Les tests ont été réalisés sur les mêmes bases de données images que précédemment. Nous informons dès à présent le lecteur que les paramétrages n'ont pas pu être optimisés et ce en raison du temps de traitement nécessaire, mais l'idée est ici de donner une tendance à l'utilisation de cette nouvelle méthode. Une perspective de ces travaux consistera à optimiser les paramètres de la méthode et à l'implémenter sur des processus légers (*threads* ou de la parallélisation par GPGPU (*General-purpose Processing on Graphics Processing Units*).

#### Détection d'objets mobiles sur la base "BUREAU"

D'après les résultats fournis dans le tableau 3.7 qui vise à comparer les résultats obtenus par le Codebook RGB (couleur uniquement), le Codebook RGB-D (couleur et profondeur) et la fusion par Logique Floue (LF, couleur uniquement), nous constatons une très nette amélioration des F-Mesures du fond (bg) avec un gain compris entre 0.08 et 0.27 selon la séquence d'apprentissage considérée, par rapport aux meilleurs résultats obtenus avec la méthode Codebook RGB ou Codebook RGB-D. Pour les objets mobiles (fg), la stratégie proposée améliore ou dégrade la F-Mesure selon la séquence d'apprentissage considérée. Il apparaît que les améliorations sont comprises entre 0.11 et 0.19 (pour les séquences $L1$, $L3$ et $L4$) et les dégradations entre 0.01 et 0.12 pour les deux autres séquences. Il est intéressant de noter que les améliorations les plus notables (à la fois sur le fond et sur les objets mobiles) sont obtenues lorsque l'apprentissage est réalisé sur les séquences $L1$ et $L4$. Toutefois, l'amélioration la plus notable sur le fond est obtenue avec la séquence d'apprentissage $L5$. Ces résultats nous permettent de situer très favorablement notre méthode de fusion par Logique Floue par rapport aux autres méthodes Codebook testées précédemment lorsqu'une base intérieure est considérée.

La figure 3.21 illustre les résultats obtenus avec la méthode Codebook RGB standard et la méthode de fusion par Logique floue proposée. Ces résultats utilisent la séquence $L1$ comme base d'apprentissage (50 frames). Nous remarquons là encore clairement une amélioration notable des segmentations avec la méthode proposée. En particulier, nous avons une diminution importante des faux-positifs qui permet une meilleure segmentation de la silhouette. Cependant, comme observé précédemment, la séquence $L5$ reste très difficile à exploiter en raison de sa luminosité extrêmement faible.

En ce qui concerne la séquence $L6$, nous observons une région importante de faux-positifs localisée à gauche de l'image, causée par une sur-illumination. La méthode par logique floue telle que proposée échoue à détecter ces pixels comme du fond. La différence d'intensité étant trop importante par rapport aux séquences précédentes, l'adaptation du modèle ne peut se faire : ces régions sont composées de pixels très clairs, proches du blanc. Considérer une intensité aussi importante comme "acceptable" dans l'algorithme de détection conduirait dans le pire des cas à la non-détection des objets mobiles. Cependant, nous améliorons grandement la segmentation de l'image par rapport à celle fournie par le Codebook RGB standard. Aussi, nous rappelons que seuls les pixels pour lesquels une correspondance a été établie sont mis à jour, fonctionnement que nous avons volontairement limité pour simuler d'importantes différences entre le modèle d'arrière-plan déterminé par l'algorithme, et les images observées.

Concernant les régions caractérisant les objets mobiles, il est clair que l'absence de faux-positifs

autour des r gions concern es permet une meilleure d tection de la silhouette, m me si dans la majorit des cas, les jambes de la silhouette ne sont pas d tect es. Ce point est notamment caus par l'apparente similitude de la couleur et d'intensit entre le pantalon et le sol. Dans la s quence $L4$, nous remarquons qu'une partie de la jambe n'est pas d tect e alors qu'elle est devant le mur blanc : le param trage de l'algorithme est ici mettre en cause, car certainement un peu trop permissif sur ces pixels en particulier.

| BG | RGB | RGB-D | RGB LF |
|---|---|---|---|
| MOY_L1 | 0,72 | 0,74 | **0,91** |
| MOY_L2 | 0,82 | 0,82 | **0,90** |
| MOY_L3 | 0,81 | 0,80 | **0,90** |
| MOY_L4 | 0,75 | 0,79 | **0,90** |
| MOY_L5 | 0,17 | 0,18 | **0,45** |
| MOY_L6 | 0,73 | 0,74 | **0,83** |

(a) Fond (bg)

| FG | RGB | RGB-D | RGB LF |
|---|---|---|---|
| MOY_L1 | 0,25 | 0,30 | **0,49** |
| MOY_L2 | 0,35 | **0,42** | 0,30 |
| MOY_L3 | 0,34 | 0,39 | **0,50** |
| MOY_L4 | 0,29 | 0,36 | **0,49** |
| MOY_L5 | 0,16 | **0,18** | 0,11 |
| MOY_L6 | 0,30 | **0,33** | 0,32 |

(b) Objets mobiles (fg)

TABLE 3.7 – F-Mesures moyennes avec des apprentissages sur les s quences L1, L2, L3, L4, L5 et L6 sans invariants pour la base "BUREAU". Comparatif des r sultats obtenus avec les algorithmes Codebook RGB, Codebook RGB-D et Codebook RGB avec Logique Floue.

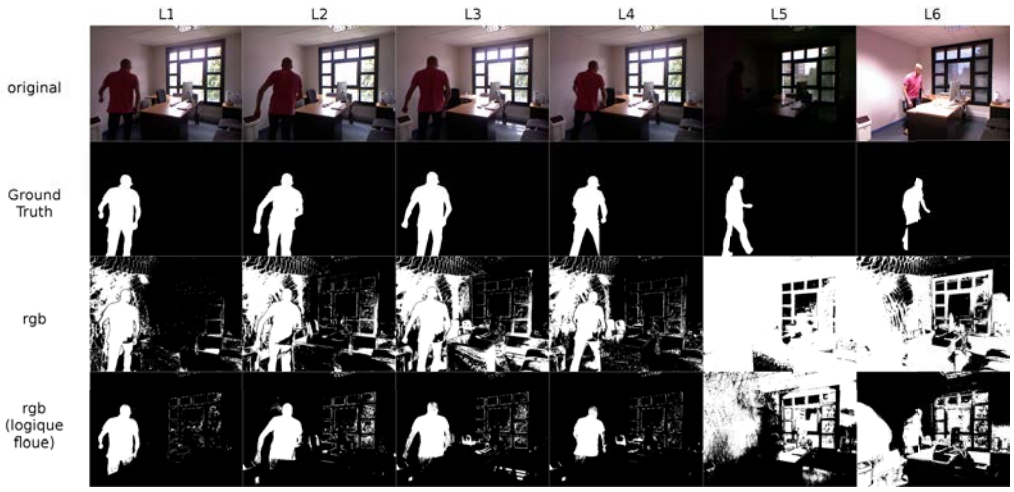



FIGURE 3.21 – Illustration de segmentations de la base "BUREAU" obtenues avec les algorithmes Codebook RGB, RGB-D et Logique Floue pour un apprentissage sur la base L1 sans utilisation de la disparit .

### Détection d'objets mobiles sur la base "JARDIN"

Au vu des r sultats fournis dans le tableau 3.8 nous constatons que la m thode bas e sur la Logique Floue n'arrive pas am liorer les r sultats obtenus pr c demment avec les Codebook RGB et RGB-D. En e et, elle d grade la segmentation du fond entre 0.02 et 0.2 et celle des objets mobiles entre 0.03 et 0.09 par rapport aux r sultats obtenus par le Codebook RGB-D. Il en d coule que la m thode, bien qu'o rant des F-Mesures relativement lev es sur le fond, et proches des r sultats obtenus avec les deux autres m thodes pr -cit es, ne permet pas d'obtenir une meilleure segmentation de l'image.

La figure 3.22 illustre les divers r sultats de segmentation. Sur cette figure, les r sultats pr sent s ont utilis la s quence $L2$ comme apprentissage. Il est alors vident que les meilleures segmentations

du fond soient obtenues sur cette m me s quence, voire tendues la s quence $L3$ qui pr sente des couleurs similaires. En revanche, les objets mobiles sont moins bien d tect s, et un nombre important de faux-n gatifs sont pr sents dans les silhouettes. Cette mauvaise performance peut s'expliquer par un param trage de l'algorithme qui s'av re plus compliqu pour des environnements ext rieurs, savoir tre la fois robuste des changements de luminosit importants pour la d tection du fond (ce qui est le cas) mais aussi assez flexible pour pouvoir d tecter des changements tels que des objets mobiles. Un ra nement des param tres peut s'av rer n cessaire dans ce cas. Une id e pourrait tre de d terminer des r gions de l'image dont la flexibilit serait moindre que pour d'autres (comme le sol par rapport au ciel).

| BG | RGB | RGB-D | RGB LF |
|---|---|---|---|
| MOY_L1 | **0,91** | 0,84 | 0,71 |
| MOY_L2 | **0,92** | 0,90 | 0,90 |
| MOY_L3 | **0,96** | 0,92 | 0,90 |
| MOY_L4 | **0,95** | 0,92 | 0,86 |

(a) Fond (bg)

| FG | RGB | RGB-D | RGB LF |
|---|---|---|---|
| MOY_L1 | 0,10 | **0,12** | 0,09 |
| MOY_L2 | 0,08 | **0,14** | 0,08 |
| MOY_L3 | 0,12 | **0,14** | 0,07 |
| MOY_L4 | 0,13 | **0,15** | 0,06 |

(b) Objets mobiles (fg)

TABLE 3.8 – F-Mesures moyennes avec des apprentissages sur les s quences L1, L2, L3 et L4 sans invariants pour la base "JARDIN". Comparatif des r sultats obtenus avec les algorithmes Codebook RGB, Codebook RGB-D et Codebook RGB avec Logique Floue.

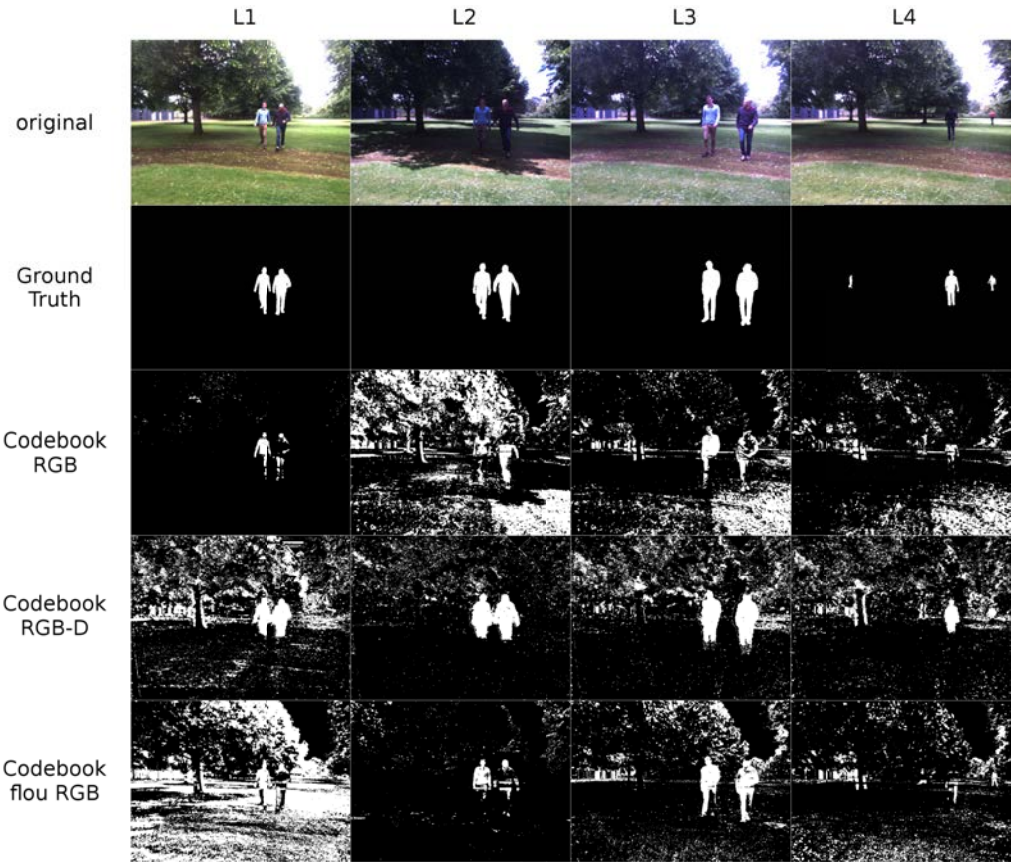



FIGURE 3.22 – Illustration de segmentations de la base "JARDIN" obtenues avec les algorithmes Codebook RGB, RGB-D et Logique Floue pour un apprentissage sur la base L2 sans utilisation de la disparit .

**Détection d'objets mobiles sur la base "RUE"**

Comme nous l'avons noté précédemment, la base "RUE" est composée de séquences très hétérogènes et est de fait assez délicate à traiter. D'après le tableau 3.9, nous constatons que la méthode par Logique Floue proposée dégrade les résultats à la fois pour le fond (bg) et pour les objets mobiles (fg) en comparaison avec ceux obtenus avec le Codebook RGB et RGB-D. En effet, la diminution de la F-Mesure s'étend entre 0.13 et 0.49 pour le fond et entre -0.11 et +0.03 pour les objets mobiles. Cependant, il est important de noter que les résultats donnés ici sont des moyennes et que le détail des tests est donné en annexe. Pour ce cas précis, il s'agit de tableaux 3.50 à 3.54. Nous y observons que dans le cas où la séquence testée est la même que la séquence d'apprentissage (par exemple $L3\_P3$ dans le tableau 3.52) nous obtenons une F-Mesure moyenne de 0.99 pour le fond et de 0.67 pour les objets mobiles, ce qui est meilleur que le résultat obtenu avec la méthode Codebook RGB-D qui offrent respectivement 0.98 pour le fond et 0.50 les objets mobiles, soit une amélioration de la détection des objets mobiles de 0.17.

Ainsi, si la méthode de fusion par logique floue n'est pas efficiente dans la globalité de l'objectif que nous nous sommes fixés, à savoir améliorer la qualité de la détection des objets mobiles sur une journée complète, il améliore cette segmentation lorsque le modèle de l'arrière-plan est à jour, et il ne la dégrade généralement que dans le cas où la séquence d'apprentissage est $L2$ (de 0.19) ou $L5$ (une séquence très sombre). Il semblerait donc que la méthode proposée de fusion par logique floue se comporte généralement de façon plus efficace lorsque la séquence d'apprentissage est d'une luminosité suffisante.

La figure 3.23 illustre les résultats de segmentations d'objets mobiles obtenus avec le Codebook RGB et RGB-D ainsi qu'avec la méthode de fusion par logique floue proposée. Hormis pour la séquence L3, la méthode proposée ne permet pas d'obtenir de meilleurs résultats sur cette base très compliquée, comme nous venons de le constater suite à l'analyse du tableau 3.9. Ceci laisse à penser qu'une perspective de ces travaux serait de tenir compte de la disparité dans la méthode de fusion.

| BG | RGB | RGB-D | RGB LF |
|---|---|---|---|
| MOY_L1 | **0,94** | 0,92 | 0,70 |
| MOY_L2 | **0,93** | 0,89 | 0,72 |
| MOY_L3 | **0,97** | 0,91 | 0,42 |
| MOY_L4 | **0,83** | **0,83** | 0,70 |
| MOY_L5 | **0,83** | **0,83** | 0,62 |

(a) Fond (bg)

| FG | RGB | RGB-D | RGB LF |
|---|---|---|---|
| MOY_L1 | **0,13** | 0,10 | **0,13** |
| MOY_L2 | 0,13 | **0,16** | 0,08 |
| MOY_L3 | **0,27** | 0,19 | 0,18 |
| MOY_L4 | **0,21** | 0,15 | 0,08 |
| MOY_L5 | 0,13 | **0,14** | 0,03 |

(b) Objets mobiles (fg)

TABLE 3.9 – F-Mesures background/foreground moyennes avec des apprentissages sur les séquences L1, L2, L3, L4 et L5 sans invariants pour la base "RUE". Comparatif des résultats obtenus avec les algorithmes Codebook RGB, Codebook RGB-D et Codebook RGB avec Logique Floue.

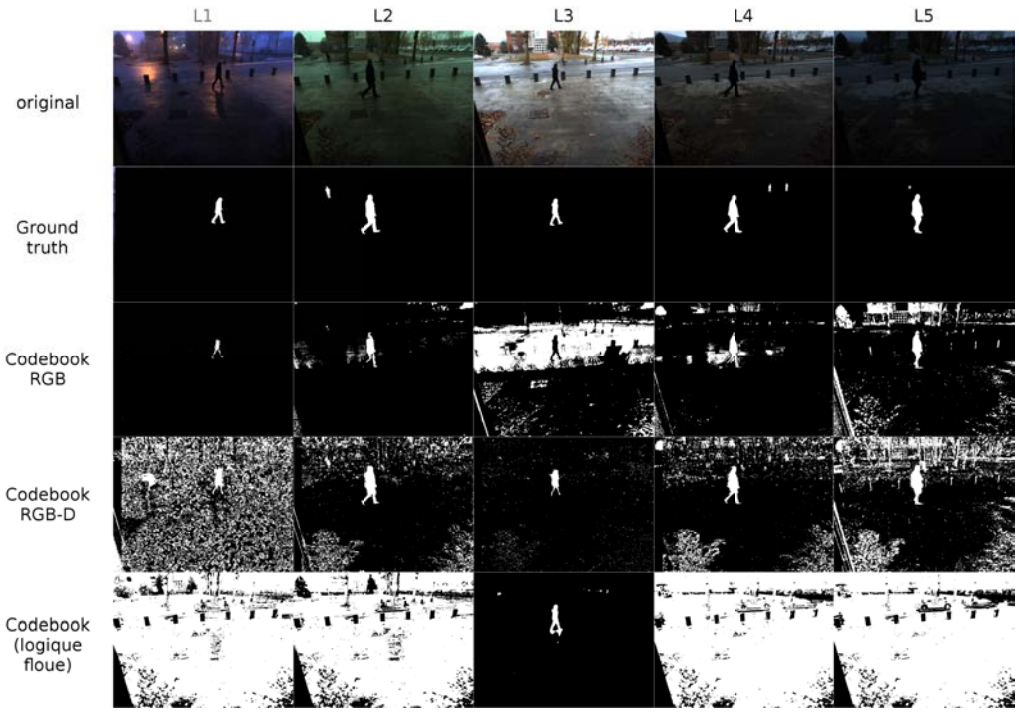



FIGURE 3.23 – Illustration de segmentations de la base “RUE” obtenues avec les algorithmes Codebook RGB, RGB-D et Logique Floue pour un apprentissage sur la base L3 sans utilisation de la disparit .

## 4.4 Conclusion

Dans ce chapitre, nous avons présenté une méthode de segmentation intitulée Codebook intégrant des informations de couleur, d'intensité et de profondeur au sein d'un même algorithme intitulé RGB-D. Nous avons dans un premier temps proposé une fusion booléenne de ces trois informations couplée à l'utilisation d'invariants colorimétriques, puis proposé une fusion par logique floue. Comme dans le chapitre précédent, les tests ont été réalisés sur trois bases d'images (1 base intérieure et 2 bases extérieures délicates). Pour ces dernières, nous avons utilisé un capteur stéréoscopique BumbleBee 2 et la méthode SGBM pour le calcul des cartes de disparités.

Nous avons visé à démontrer que la fusion de données couleur et profondeur permet d'améliorer la qualité des segmentations des objets mobiles avec l'algorithme Codebook. Cette conclusion peut être tirée de nos résultats qui confirment les avancées récentes en la matière, particulièrement avec la base acquise en milieu intérieur. Nous observons dans certains cas des améliorations notables à la fois dans les régions du fond et celles des objets mobiles. Nous souhaitions également mettre l'accent sur l'utilisation de cette méthode sur des bases d'images acquises en milieu extérieur à l'aide d'une caméra stéréoscopique. Les résultats obtenus sur ces images en particulier ont montré qu'ils étaient hautement dépendants du capteur employé, et surtout de la qualité des cartes de disparités calculées.

Ces dernières étant calculées à partir des images gauche et droite du capteur stéréoscopique, l'obscurité (et de manière générale, tout facteur impactant l'aspect colorimétrique et l'intensité de la scène) n'est pas négligeable pour la maximisation de leur justesse. Des différents et nombreux travaux ayant montré des avancées dans ce domaine, il serait intéressant de comparer nos résultats avec les mêmes segmentations obtenues grâce à de meilleures cartes de disparité. Néanmoins nous avons pu constater que nos résultats ne montraient parfois qu'un impact faible sur la qualité des segmentations obtenues, mais globalement nous pouvions observer plusieurs améliorations.

Aussi, nous avons testé et comparé un nombre important d'invariants colorimétriques et d'espaces couleurs possédant des caractéristiques invariantes. Nous avons utilisé ces invariants colorimétriques avec succès et nous avons montré l'efficacité de plusieurs d'entre eux : Chromaticity Space se retrouve notamment le plus souvent en première place dans le classement, ainsi que RGB-Rang et l'espace colorimétrique c1c2c3. Leur impact positif est autant présent en milieu intérieur qu'en milieu extérieur. Ces résultats sont très encourageants pour la résolution des problématiques traitées, car ils montrent que le choix d'un invariant colorimétrique peut être particulièrement bénéfique même si l'aspect colorimétrique de la scène est amené à changer.

Quant à la méthode de fusion à base de logique floue, nous pensons que malgré une complexité accrue, elle a la capacité d'améliorer grandement la qualité des segmentations. La difficulté de son utilisation réside bien sûr dans son paramétrage, qui est global et qui impacte la flexibilité des règles d'inférence floue. Malgré cette difficulté, cette méthode est parvenue à améliorer dans certains cas les résultats obtenus avec l'algorithme Codebook classique dans plusieurs cas de figure. Néanmoins, cette méthode a l'inconvénient de nécessiter un temps de traitement plus important. De plus, il s'avère que la flexibilité souhaitée est dépendante des régions des images : certaines régions sombres sont généralement trop permissives et entraînent quelques faux-négatifs, alors que certaines régions claires ne sont pas assez permissives et entraînent des régions de faux-positifs, notamment les sur-illuminations. Pour éviter cela, nous pourrions envisager en perspective de ces travaux l'utilisation de deux facteurs permettant d'élargir les sous-ensembles flous pour les pixels clairs, et de les réduire pour les pixels sombres.

Une autre perspective importante de ces travaux consisterait à tenir compte de l'information de disparité dans la méthode de fusion par logique floue. Pour cela, une parallélisation de l'algorithme et de la méthode développée est à envisager.

# Conclusions et perspectives

Dans le cadre d'applications liées aux systèmes de vidéo-surveillance ou aux systèmes de transports intelligents, il est depuis de nombreuses années admis que la détection, le suivi voire la réidentification d'objets mobiles est généralement impactée de façon importante par les changements de luminosité de la scène, ce qui peut conduire à des décisions erronées, ce qui peut avoir des conséquences graves dans des applications à caractère sécuritaire. Dans cette thèse, nous avons cherché à répondre à cette problématique en proposant une stratégie de détection d'objets mobiles opérable en tous lieux (intérieur, extérieur) et à tout moment de la journée (en s'affranchissant des changements de luminosité, ombres portées, etc).

Plusieurs pistes ont été explorées en vue d'améliorer la qualité des segmentations d'objets mobiles, tout d'abord en exploitant conjointement l'algorithme Codebook avec différents invariants colorimétriques. Nous avons montré lors de nos tests réalisés sur trois bases de données réelles acquises en intérieur et en extérieur, que l'utilisation de ces invariants colorimétriques comme prétraitement est une stratégie gagnante puisqu'elle permet d'améliorer les résultats dans de nombreux cas. Nous avons notamment constaté l'efficacité de plusieurs d'entre eux, comme c'est le cas pour Chromaticity Space, RGB-Rang et espace c1c2c3 notamment. Afin d'affiner cette stratégie, nous avons testé plusieurs paramétrages pour l'algorithme de détection Codebook, et ce pour chacune des bases vidéo. Nous avons déterminé empiriquement les valeurs de ces paramètres permettant d'obtenir les meilleurs résultats possibles, notamment sur les bases de données extérieures, ce qui n'avait pas été réalisé selon nous dans les travaux de référence dans le domaine.

De nombreux tests ont également été réalisés en modifiant la séquence d'apprentissage de l'algorithme, afin de déterminer l'impact de celle-ci sur les segmentations d'objets mobiles. Nous avons déterminé celle(s) qui permet(tent) d'obtenir les résultats les plus efficients et donc à privilégier pour l'initialisation de l'algorithme de détection des objets mobiles.

Nous avons ensuite exploré et démontré les bénéfices de l'utilisation d'une information supplémentaire au sein même de l'algorithme Codebook, à savoir la profondeur des objets dans la scène obtenue à l'aide d'une caméra Microsoft Kinect pour la base intérieure, et d'une caméra stéréoscopique BumbleBee 2 (après un calcul de mise en correspondance) pour les bases extérieures. Nous avons effectué une fusion booléenne d'informations RGB-D au sein de l'algorithme de détection Codebook. Il en ressort que la prise en compte de cette information supplémentaire a bien un impact positif sur la qualité des segmentations obtenues, mais relatif à cause de la qualité des cartes de disparités. Ces dernières ont été calculées à l'aide de la méthode Semi-Global Block Matching bien connue de la littérature pour son efficacité et sa rapidité. Néanmoins, force est de constater que la méthode de fusion RGB-D est très dépendante de la qualité des cartes de disparités obtenues avec parfois la présence de bruit additionnel, en particulier pour les scènes extérieures, et celles où la luminosité est très faible. En revanche dans les autres cas cette méthode a démontré son efficacité et permis d'améliorer les résultats. Comme précédemment, nous avons couplé cet algorithme intégrant l'information de profondeur à un pré-traitement des images basé sur les invariants colorimétriques, afin de valider une nouvelle fois l'impact positif de certains d'entre eux.

Enfin, sur la base de ces résultats positifs apportés par la fusion RGB-D dans l'algorithme Codebook, nous avons proposé une nouvelle méthode de fusion utilisant la Logique Floue. Les tests effectués pour cette méthode n'ont pu être aussi nombreux que pour les précédentes stratégies

proposées, mais si cette nouvelle méthode de fusion n'a pas forcément permis de surpasser ces dernières, elle a pu s'en approcher suffisamment pour valider son intérêt. Nous avons la conviction que cette méthode peut être améliorée et optimisée pour permettre des résultats meilleurs, voire ouvrir la voie à la résolution d'autres problématiques rencontrées dans la détection des objets mobiles.

Les résultats présentés dans cette thèse permettent d'envisager des perspectives porteuses à court et moyen termes.

— La stratégie d'utiliser conjointement un ou plusieurs invariants colorimétriques avec une méthode de détection mobile étant validée, il serait intéressant de la tester avec d'autres algorithmes de détection d'objets mobiles qui ont pu être proposés ultérieurement aux travaux décrits dans le présent mémoire. Nous avons utilisé l'algorithme Codebook en raison de son efficacité, notamment en comparaison avec d'autres méthodes comme les Mixtures de Gaussiennes (MOG) également réputées, mais aussi en raison de son évolutivité vers la fusion d'informations supplémentaires comme la profondeur. Cependant, il serait intéressant maintenant de considérer le cas de l'algorithme ViBe [4] qui semble démontrer des performances accrues par rapport à toutes les méthodes existantes.

— Nous avons proposé de considérer l'information de profondeur dans la méthode de fusion RGB-D de l'algorithme Codebook. Nous avions noté précédemment la sensibilité de la méthode proposée au bruit causé par la qualité variable des cartes de disparités. Nous avons tenté de calculer des cartes de disparités à l'aide d'images pré-traitées avec des invariants colorimétriques, dont l'aspect avait été jugé qualitativement. L'impact réel de l'utilisation des invariants colorimétriques dans cet objectif est une perspective future intéressante, étant donnés les résultats qu'ils ont permis d'obtenir dans le cadre de la détection des objets mobiles.

— D'autres informations peuvent présenter un intérêt non négligeable pour améliorer la segmentation des objets mobiles. Nous pensons que cela pourrait être le cas avec les contours des objets mobiles et leur intégration dans une nouvelle couche de la méthode de détection afin d'affiner la détection finale des objets d'intérêt.

— Il existe aussi de nouveaux invariants colorimétriques qui devraient être testés afin de comparer leurs performances par rapport à ceux qui ont été testés au cours de cette thèse. Nous avons remarqué la forte sensibilité de la stratégie utilisée à la présence d'ombres dans la scène. Un autre pré-traitement à envisager serait une suppression efficace de ces ombres dans les images couleur à l'aide d'une méthode adéquate.

— Plusieurs des résultats présentés dans cette thèse peuvent être améliorés par une étape de post-traitement qui n'a pas été abordée au cours de ces travaux. Une étape de filtrage peut être envisagée afin de retirer un grand nombre de faux positifs, particulièrement dans les régions du fond.

Concernant notre dernière proposition de fusion par logique floue des informations de couleur et de profondeur au sein même de l'algorithme Codebook, nous avons conscience que la totalité des tests possibles n'ont pu être réalisés pendant cette thèse. Aussi, nous envisageons d'approfondir les règles de fusion et d'optimiser la méthode globale en parallélisant le traitement dans plusieurs processus légers (*multithreading*), voire de les partager sur la carte graphique de l'ordinateur effectuant les traitements à l'aide d'une méthode de GPGPU (*General-Purpose Computing on Graphics Processing Units*). Ceci permettrait aussi de pouvoir tester d'autres méthodes de mise en corres-

pondance et permettre ainsi d'obtenir de meilleures cartes de disparités et *in fine* une meilleure détection des objets mobiles. Ajoutons à ces propositions l'éventualité de gérer dynamiquement les paramètres de la méthode de fusion par logique floue, par exemple pour chaque pixel ou pour des régions spécifiques de l'image (comme les régions les plus sensibles aux changements de luminosité), ou encore l'emploi de deux facteurs permettant de rendre l'algorithme plus flexible pour les régions sur-illuminées et moins flexible pour les régions sombres.

Enfin, comme perspective à long terme, il serait intéressant d'adapter cette méthode avec l'utilisation d'un capteur infra-rouge et permettre une détection des objets de nuit afin que le système puisse être, comme indiqué en préambule, opérable en tous lieux et à tout moment de la journée (et de la nuit).

# Glossaire

**Albédo** On définit l'albédo d'une surface comme son pouvoir réfléchissant. Il correspond au rapport entre l'énergie lumineuse d'un rayon incident à la surface et l'énergie lumineuse du rayon réfléchi. Dans sa définition la plus courante, il a une valeur comprise entre 0 et 1, 0 correspondant à une absorption totale de toutes les longueurs d'onde par un corps noir parfait, et 1 correspondant à un miroir parfait qui réfléchirait la totalité des longueurs d'onde sans en absorber une seule.. 29

**Chrominance** La chrominance désigne l'information de couleur codée dans un signal vidéo qui distingue cette information de l'information de luminance. C'est le cas de plusieurs espaces couleur utilisés pour la transmission vidéo, comme YUV, YCbCr ou YIQ.. 32, 33

**Clarté** La clarté (**brightness** en anglais) est un attribut subjectif de la lumière. Il s'agit d'une perception et ne peut être mesurée objectivement. En revanche, la clarté peut être exprimée sur une échelle (par exemple en %), ce qui implique la définition de clartés minimum et maximum. Une valeur de 0 représente le noir (foncé), 100 représente le blanc (clair). On peut également parler d'intensité.. 32

**Luminance** La luminance représente l'intensité de la lumière, projetée sur une surface et dans une direction données. Il s'agit d'un attribut mesurable, exprimé en Candela par mètre carré (Cd/m2). Une valeur faible indique une luminosité faible ; inversement, une valeur élevée indique une luminosité intense. A titre d'exemple, la luminance du Soleil à midi est équivalente à $1.6 \times 10^9 cd/m^2$.. 32, 33

# Bibliographie


[1] Cie colorimetry - part 4 : 1976 l*a*b* colour space, iso 11664-4 :2008(e)/cie s 014-4/e :2007, 1976.

[2] T. Aydin and Y. S. Akgul. Stereo depth estimation using synchronous optimization with segment based regularization. *Pattern Recognition Letters*, 31 :2389 – 2396, 2010.

[3] S. T. Barnard. A stochastic approch to stereo vision. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision*, pages 21 – 25. Morgan Kaufmann, San Francisco (CA), 1987.

[4] O. Barnich and M. Van Droogenbroeck. Vibe : A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6) :1709 –1724, Juin 2011.

[5] S. Bianco, G. Ciocca, C. Cusano, and R. Schettini. Automatic color constancy algorithm selection and combination. *Pattern Recogn.*, 43(3) :695–705, Mars 2010.

[6] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV)*, pages 1073–1080, Jan. 1998.

[7] A. Boucher, O. Martinot, and N. Vincent. Depth camera to improve segmenting people in indoor environments - real time rgb-depth video segmentation. In *VISAPP 2015 - Proceedings of the 10th International Conference on Computer Vision Theory and Applications, Volume 3, Berlin, Germany, 11-14 March, 2015.*, pages 55–62, 2015.

[8] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 1980.

[9] D. Butler, S. Sridharan, and J. Bove, V.M. Real-time adaptive background segmentation. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 3, pages III–349–52 vol.3, April 2003.

[10] T. Carron. *Segmentation d'images couleur dans la base Teinte Luminance Saturation : approche numérique et symbolique.* PhD thesis, Université de Stanford, 1995.

[11] T. Chalidabhongse, K. Kim, D. Harwood, and L. S. Davis. A perturbation method for evaluating background subtraction algorithms. *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, 2003/// 2003.

[12] S. Chambon. *Mise en correspondance stéréoscopique d'images couleur en présence d'occultation.* PhD thesis, Université Paul Sabatier, Toulouse, décembre 2005.

[13] G. Chiron, P. Gomez-Kramer, and M. Menard. Detecting and tracking honeybees in 3d at the beehive entrance using stereo vision. *EURASIP Journal on Image and Video Processing*, 2013(1) :59, 2013.

[14] C. Cigla. Recursive edge-aware filters for stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2015.

[15] D. Craciun, A. F. Serna-Morales, J.-E. Deschaud, B. Marcotegui, and F. Goulette. Scalable and Detail-Preserving Ground Surface Reconstruction from Large 3D Point Clouds Acquired by Mobile Mapping Systems. In *PCV (Photogrammetric Computer Vision)*, volume 3, pages 73 – 80, Zurich, Switzerland, Sept. 2014.

[16] M. Ebner. Color constancy based on local space average color. *Machine Vision and Applications*, 20(5) :283–301, 2009.

[17] N. Einecke and J. Eggert. A two-stage correlation method for stereoscopic depth estimation. In *Proceedings of the International Conference on Digital Image Computing : Techniques and Applications, DICTA 2010*. IEEE, 2010.

[18] Y. El Merabet, C. Meurie, Y. Ruichek, A. Sbihi, and R. Touahni. Segmentation d'images aériennes par coopération lpe-régions et lpe-contours. application à la caractérisation de toitures. *Revue Francaise de Photogrammetrie et de Teledetection*, (206) :29–44, 2014.

[19] A. Elgammal, D. Harwood, and L. Davis. *Non-parametric Model for Background Subtraction*, pages 751–767. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[20] N. Fakhfakh. *Detection and 3D localization of moving and stationary obstacles by stereo vision in complex environments : application at level crossings.* phdthesis, Ecole Centrale de Lille, June 2011.

[21] Y. Fang, C. Cappelle, and Y. Ruichek. Multisensor based obstacles detection in challenging scenes. In A. Gelbukh, F. Espinoza, and S. Galicia-Haro, editors, *Human-Inspired Computing and Its Applications*, volume 8856 of *Lecture Notes in Computer Science*, pages 257–268. Springer International Publishing, 2014.

[22] E. J. Fernandez-Sanchez, J. Diaz, and E. Ros. Background subtraction based on color and depth using active sensors. *Sensors*, 13(7) :8895, 2013.

[23] G. Finlayson, S. Hordley, C. Lu, and M. Drew. On the removal of shadows from images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1) :59–68, Jan 2006.

[24] G. D. Finlayson, S. D. Hordley, and M. Drew. Removing shadows from images using retinex. *Color and Imaging Conference*, 2002(1) :73–79, 2002.

[25] G. D. Finlayson, S. D. Hordley, and M. S. Drew. Removing shadows from images. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, ECCV '02, pages 823–836, London, UK, UK, 2002. Springer-Verlag.

[26] G. D. Finlayson, S. D. Hordley, G. Schaefer, and G. Y. Tian. Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 38(2) :179–190, 2005.

[27] G. D. Finlayson, B. Schiele, and J. L. Crowley. *Comprehensive colour image normalization*, pages 475–490. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[28] S. Forstmann, Y. Kanou, J. Ohya, S. Thuering, and A. Schmitt. Real-time stereo by using dynamic programming. In *In Conference on Computer Vision and Pattern Recognition Workshop*, page 29. IEEE Computer Society, 2004.

[29] U. Franke, C. Rabe, H. Badino, and S. K. Gehrig. 6d-vision : Fusion of stereo and motion for robust environment perception. In *Proceedings of the 27th DAGM Symposium*, pages 216–223, Vienna, Austria, August 2005. Springer.

[30] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 06 2008.

[31] S. K. Gehrig, F. Eberli, and T. Meyer. A real-time low-power stereo vision engine using semi-global matching. In M. Fritz, B. Schiele, and J. Piater, editors, *Proceedings of the 7th International Conference on Computer Vision Systems : Computer Vision Systems*, volume 5815 of *Lecture Notes in Computer Science*, pages 134–143. Springer, 2009.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics : The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[33] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part I*, ACCV'10, pages 25–38, Berlin, Heidelberg, 2011. Springer-Verlag.

[34] T. Gevers and A. W. Smeulders. Color-based object recognition. *Pattern Recognition*, 1999.

[35] T. Gevers and A. W. M. Smeulders. Object recognition based on photometric color invariants. In *Pattern Recognition Society of Finland*, volume SCIA, Lappeenranta, Finland, 1997.

[36] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2010.

[37] A. Gijsenij, R. Lu, and T. Gevers. Color constancy for multiple light sources. *Image Processing, IEEE Transactions on*, 21(2) :697–707, Feb 2012.

[38] M. Greiffenhagen, V. Ramesh, D. Comaniciu, and H. Niemann. Statistical modeling and performance characterization of a real-time dual camera surveillance system. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 335–342 vol.2, 2000.

[39] Heidelberg Collaboratory for Image Processing. *HCI Robust Vision Challenge*, 2015. Available at `http://hci.iwr.uni-heidelberg.de/Static/challenge2012/`.

[40] S. Hermann and R. Klette. Iterative semi-global matching for robust driver assistance systems. In K. Lee, Y. Matsushita, J. Rehg, and Z. Hu, editors, *Computer Vision – ACCV 2012*, volume 7726 of *Lecture Notes in Computer Science*, pages 465–478. Springer Berlin Heidelberg, 2013.

[41] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(2) :328–341, 2008.

[42] H. Hirschmuller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3) :229–246, 2002.

[43] E. Hsu, T. Mertens, S. Paris, S. Avidan, and F. Durand. Light mixture estimation for spatially varying white balance. *ACM Trans. Graph.*, 27(3) :70 :1–70 :7, Aug. 2008.

[44] I. Huerta, M. Holte, T. Moeslund, and J. Gonzalez. Detection and removal of chromatic moving shadows in surveillance scenarios. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1499–1506, Sept 2009.

[45] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Computer Vision - ECCV'98*, pages 232–248. Springer Berlin Heidelberg, 1998.

[46] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion : real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[47] K. Kim, T. H. Chalidabhongse, D. Hanuood, and L. Davis. Background modeling and substraction by codebook construction. *International Conference on Image Processing (ICIP 2004)*, 2004.

[48] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 508–515. IEEE, 2001.

[49] J. Kowalczuk, E. T. Psota, and L. C. Perez. Real-time stereo matching on cuda using an iterative refinement method for adaptive support-weight correspondences. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1) :94–104, Jan 2013.

[50] E. H. Land. The retinex theory of color vision. *Scientific American*, 237, no. 6 :108–128, December 1977.

[51] J. Marais, C. Meurie, D. Attia, Y. Ruichek, and A. Flancquart. Toward accurate localization in guided transport : combining gnss data and imaging information. *Transportation Research Part C : Emerging Technologies*, 43 :188–197, 2014.

[52] N. McFarlane and C. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3) :187–193, 1995.

[53] S. Messelodi, C. Modena, N. Segata, and M. Zanin. A kalman filter based background updating algorithm robust to sharp illumination changes. In F. Roli and S. Vitulano, editors, *Image Analysis and Processing – ICIAP 2005*, volume 3617 of *Lecture Notes in Computer Science*, pages 163–170. Springer Berlin Heidelberg, 2005.

[54] C. Meurie. *Segmentation d'images couleur par classification pixellaire et hiérarchie de partitions*. phdthesis, Université de Caen Basse-Normandie, Oct. 2005. Thèse de doctorat dirigée par Elmoataz, Abderrahim Informatique Caen 2005.

[55] Middlebury Insitute. *Middlebury Stereo Vision Page*, 2015. Available at `vision.middlebury.edu/stereo/`.

[56] K.-i. Min, J.-S. Oh, and B.-W. Kimm. Traffic sign extract and recognition on unmanned vehicle using image processing based on support vector machine. In *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*, pages 750–753, Oct 2011.

[57] N. Neverova, D. Muselet, and A. Trémeau. 21/2 d scene reconstruction of indoor scenes from single rgb-d images. 7786 :281–295, 2013.

[58] S. Obdrzalek, J. Matas, and O. Chum. On the interaction between object recognition and colour constancy. *Proc. International Workshop on Color and Photometric Methods in Computer Vision*, 2003.

[59] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8) :831–843, Aug 2000.

[60] OpenCV. *OpenCV's SGBM implementation documentation*, 2008. Available at `http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#stereosgbm`.

[61] J. Pearl. Reverend bayes on inference engines : a distributed hierarchical approach. In *in Proceedings of the National Conference on Artificial Intelligence*, pages 133–136, 1982.

[62] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara. Detecting moving shadows : Formulation, algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 :2003, 2003.

[63] V. Prinet, D. Lischinski, and M. Werman. Illuminant chromaticity from image sequences. December 2013.

[64] E. T. Psota, J. Kowalczuk, M. Mittek, and L. C. Perez. Map disparity estimation using hidden markov trees. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[65] C. Rabe, U. Franke, and S. Gehrig. Fast detection of moving objects in complex scenarios. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 398–403, Istanbul, June 2007.

[66] C. Rabe, T. Müller, A. Wedel, and U. Franke. Dense, robust, and accurate motion field estimation from stereo image sequences in real-time. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Proceedings of the 11th European Conference on Computer Vision*, volume 6314 of *Lecture Notes in Computer Science*, pages 582–595. Springer, September 2010.

[67] C. R. Rosenberg, T. P. Minka, and A. Ladsariya. Bayesian color constancy with non-gaussian models. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *NIPS*. MIT Press, 2003.

[68] Y. Ruichek. *Perception de l'environnement par stéréovision : Application à la sécurité dans les systèmes de transports terrestres*. PhD thesis, Université des Sciences et Technologies de Lille, 2005.

[69] S. A. Shafer. Using color to separate reflection components. In G. E. Healey, S. A. Shafer, and L. B. Wolff, editors, *Physics-Based Vision : Principles and Practice : Color, Volume 2*, chapter Using Color to Separate Reflection Components, pages 43–51. Jones and Bartlett Publishers, Inc., USA, 1992.

[70] Y. Shan, F. Yang, and R. Wang. Color space selection for moving shadow elimination. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pages 496–501, Aug 2007.

[71] M. H. Sigari, N. Mozayani, and H. R. Pourreza. Fuzzy running average and fuzzy background subtraction : Concepts and application, 2005.

[72] S. N. Sinha, D. Scharstein, and R. Szeliski. Efficient high-resolution stereo matching using local plane sweeps. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1582–1589, June 2014.

[73] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2 :2246, 1999.

[74] C. Stentoumis, L. Grammatikopoulos, I. Kalisperakis, and G. Karras. On accurate dense stereo-matching using a local adaptive multi-cost approach. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, 91 :29 – 49, 2014.

[75] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6) :1068–1080, 2008.

[76] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower : principles and practice of background maintenance. 1 :255 –261 vol.1, 1999.

[77] J. van de Weijer and T. Gevers. Color constancy based on the grey-edge hypothesis. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–722–5, Sept 2005.

[78] J. Vazquez-Corral, M. Vanrell, R. Baldrich, and F. Tous. Color constancy by category correlation. *Image Processing, IEEE Transactions on*, 21(4) :1997 –2007, april 2012.

[79] L. Wei, C. Cappelle, and Y. Ruichek. Unscented information filter based multi-sensor data fusion using stereo camera, laser range finder and gps receiver for vehicle localization. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1923–1928, Oct 2011.

[80] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder : real-time tracking of the human body. pages 51 –56, oct 1996.

[81] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui. Meshstereo : A global stereo model with mesh alignment regularization for view interpolation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2057–2065, Dec 2015.

[82] J. Zheng, Y. Wang, N. Nihan, and E. Hallenbeck. Extracting roadway background image : A mode based approach. *Journal of Transportation Research Report*, 1944 :82–88, 2006.

[83] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31 Vol.2, Aug 2004.

# Annexes

# 1 Codebook RGB : résultats détaillés L1 à L6 pour la base “BUREAU”

TABLE 3.10 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L1 (base “BUREAU”) et les paramétrages optimaux de l’algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,82 | 0,33 | 0,98 | 0,66 | 0,82 | 0,34 | 0,82 | 0,35 | 0,78 | 0,19 | 0,92 | 0,55 | 0,98 | 0,71 |
| L1_P2 | 0,80 | 0,25 | 0,98 | 0,55 | 0,81 | 0,26 | 0,82 | 0,27 | 0,75 | 0,15 | 0,91 | 0,37 | 0,98 | 0,61 |
| L1_P3 | 0,72 | 0,22 | 0,97 | 0,53 | 0,73 | 0,23 | 0,74 | 0,24 | 0,72 | 0,16 | 0,91 | 0,44 | 0,97 | 0,62 |
| L1_P4 | 0,82 | 0,25 | 0,97 | 0,53 | 0,83 | 0,26 | 0,83 | 0,27 | 0,79 | 0,16 | 0,92 | 0,42 | 0,97 | 0,56 |
| L1_P5 | 0,16 | 0,17 | 0,74 | 0,23 | 0,17 | 0,17 | 0,15 | 0,16 | 0,70 | 0,20 | 0,66 | 0,20 | 0,66 | 0,21 |
| L1_P6 | 0,51 | 0,17 | 0,96 | 0,39 | 0,52 | 0,17 | 0,53 | 0,17 | 0,57 | 0,14 | 0,69 | 0,19 | 0,93 | 0,37 |
| **MOY** | **0,72** | **0,25** | **0,96** | **0,53** | **0,72** | **0,26** | **0,73** | **0,26** | **0,73** | **0,16** | **0,87** | **0,40** | **0,95** | **0,58** |
| +/- | 0,00 | 0,00 | 0,24 | 0,28 | 0,01 | 0,01 | 0,01 | 0,01 | 0,02 | -0,08 | 0,16 | 0,15 | 0,23 | 0,33 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,78 | 0,28 | 0,88 | 0,41 | 0,94 | 0,63 | 0,97 | 0,72 | 0,95 | 0,65 | 0,98 | 0,74 | 0,90 | 0,23 |
| L1_P2 | 0,75 | 0,22 | 0,89 | 0,35 | 0,92 | 0,49 | 0,96 | 0,57 | 0,93 | 0,52 | 0,98 | 0,62 | 0,89 | 0,19 |
| L1_P3 | 0,75 | 0,24 | 0,84 | 0,32 | 0,90 | 0,44 | 0,94 | 0,58 | 0,90 | 0,48 | 0,97 | 0,58 | 0,86 | 0,19 |
| L1_P4 | 0,75 | 0,22 | 0,86 | 0,32 | 0,97 | 0,57 | 0,98 | 0,66 | 0,98 | 0,60 | 0,98 | 0,66 | 0,89 | 0,18 |
| L1_P5 | 0,39 | 0,17 | 0,19 | 0,17 | 0,39 | 0,16 | 0,39 | 0,18 | 0,37 | 0,17 | 0,19 | 0,17 | 0,76 | 0,15 |
| L1_P6 | 0,60 | 0,17 | 0,46 | 0,16 | 0,73 | 0,18 | 0,69 | 0,20 | 0,69 | 0,18 | 0,79 | 0,20 | 0,79 | 0,12 |
| **MOY** | **0,71** | **0,23** | **0,78** | **0,32** | **0,87** | **0,47** | **0,89** | **0,55** | **0,87** | **0,49** | **0,90** | **0,57** | **0,87** | **0,19** |
| +/- | 0,00 | -0,02 | 0,06 | 0,07 | 0,16 | 0,22 | 0,18 | 0,30 | 0,16 | 0,24 | 0,19 | 0,32 | 0,15 | -0,06 |

TABLE 3.11 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L2 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,92 | 0,42 | 0,98 | 0,63 | 0,92 | 0,43 | 0,92 | 0,42 | 0,84 | 0,21 | 0,94 | 0,50 | 0,98 | 0,71 |
| L2_P3 | 0,85 | 0,31 | 0,97 | 0,54 | 0,85 | 0,31 | 0,84 | 0,30 | 0,77 | 0,18 | 0,92 | 0,46 | 0,97 | 0,63 |
| L2_P4 | 0,92 | 0,39 | 0,98 | 0,55 | 0,92 | 0,39 | 0,92 | 0,38 | 0,80 | 0,17 | 0,91 | 0,37 | 0,97 | 0,59 |
| L2_P5 | 0,16 | 0,17 | 0,75 | 0,23 | 0,16 | 0,17 | 0,15 | 0,17 | 0,70 | 0,19 | 0,65 | 0,19 | 0,66 | 0,21 |
| L2_P6 | 0,63 | 0,20 | 0,96 | 0,40 | 0,63 | 0,20 | 0,63 | 0,20 | 0,57 | 0,14 | 0,68 | 0,19 | 0,93 | 0,38 |
| L2_P1 | 0,90 | 0,43 | 0,97 | 0,61 | 0,89 | 0,42 | 0,88 | 0,41 | 0,77 | 0,19 | 0,89 | 0,41 | 0,97 | 0,65 |
| **MOY** | **0,82** | **0,35** | **0,96** | **0,54** | **0,81** | **0,35** | **0,81** | **0,34** | **0,76** | **0,18** | **0,87** | **0,39** | **0,95** | **0,59** |
| +/- | 0,00 | 0,00 | 0,14 | 0,19 | 0,00 | 0,00 | -0,01 | -0,01 | -0,05 | -0,17 | 0,05 | 0,04 | 0,13 | 0,24 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,82 | 0,26 | 0,92 | 0,42 | 0,94 | 0,63 | 0,97 | 0,71 | 0,96 | 0,67 | 0,98 | 0,69 | 0,92 | 0,24 |
| L2_P3 | 0,76 | 0,25 | 0,84 | 0,32 | 0,91 | 0,49 | 0,96 | 0,62 | 0,92 | 0,52 | 0,97 | 0,60 | 0,87 | 0,19 |
| L2_P4 | 0,77 | 0,23 | 0,87 | 0,33 | 0,97 | 0,61 | 0,98 | 0,70 | 0,98 | 0,64 | 0,98 | 0,69 | 0,89 | 0,18 |
| L2_P5 | 0,40 | 0,17 | 0,20 | 0,17 | 0,39 | 0,16 | 0,39 | 0,18 | 0,37 | 0,17 | 0,19 | 0,17 | 0,77 | 0,15 |
| L2_P6 | 0,61 | 0,17 | 0,49 | 0,17 | 0,74 | 0,19 | 0,69 | 0,20 | 0,69 | 0,19 | 0,80 | 0,21 | 0,80 | 0,12 |
| L2_P1 | 0,77 | 0,28 | 0,88 | 0,40 | 0,96 | 0,60 | 0,97 | 0,67 | 0,97 | 0,62 | 0,98 | 0,68 | 0,89 | 0,21 |
| **MOY** | **0,74** | **0,24** | **0,79** | **0,33** | **0,89** | **0,51** | **0,90** | **0,59** | **0,89** | **0,53** | **0,90** | **0,58** | **0,87** | **0,19** |
| +/- | -0,08 | -0,11 | -0,03 | -0,02 | 0,07 | 0,16 | 0,08 | 0,23 | 0,07 | 0,18 | 0,09 | 0,23 | 0,06 | -0,16 |

TABLE 3.12 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L3 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,89 | 0,39 | 0,97 | 0,56 | 0,89 | 0,39 | 0,89 | 0,39 | 0,82 | 0,23 | 0,92 | 0,47 | 0,98 | 0,68 |
| L3_P4 | 0,90 | 0,34 | 0,98 | 0,53 | 0,89 | 0,34 | 0,89 | 0,34 | 0,80 | 0,16 | 0,87 | 0,30 | 0,97 | 0,57 |
| L3_P5 | 0,14 | 0,17 | 0,75 | 0,23 | 0,15 | 0,17 | 0,14 | 0,16 | 0,70 | 0,19 | 0,63 | 0,19 | 0,67 | 0,21 |
| L3_P6 | 0,61 | 0,19 | 0,96 | 0,39 | 0,61 | 0,19 | 0,61 | 0,19 | 0,56 | 0,13 | 0,66 | 0,18 | 0,93 | 0,36 |
| L3_P1 | 0,89 | 0,41 | 0,97 | 0,62 | 0,89 | 0,41 | 0,88 | 0,41 | 0,78 | 0,18 | 0,85 | 0,34 | 0,97 | 0,65 |
| L3_P2 | 0,89 | 0,34 | 0,98 | 0,56 | 0,89 | 0,35 | 0,89 | 0,34 | 0,79 | 0,16 | 0,87 | 0,29 | 0,98 | 0,63 |
| **MOY** | **0,81** | **0,34** | **0,96** | **0,53** | **0,81** | **0,34** | **0,81** | **0,34** | **0,77** | **0,18** | **0,84** | **0,32** | **0,95** | **0,58** |
| +/- | 0,00 | 0,00 | 0,15 | 0,19 | 0,00 | 0,00 | 0,00 | 0,00 | -0,04 | -0,16 | 0,03 | -0,02 | 0,14 | 0,24 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,82 | 0,28 | 0,89 | 0,38 | 0,94 | 0,63 | 0,96 | 0,72 | 0,94 | 0,65 | 0,98 | 0,68 | 0,88 | 0,21 |
| L3_P4 | 0,79 | 0,23 | 0,85 | 0,28 | 0,97 | 0,59 | 0,98 | 0,65 | 0,97 | 0,59 | 0,98 | 0,63 | 0,86 | 0,16 |
| L3_P5 | 0,40 | 0,17 | 0,23 | 0,17 | 0,41 | 0,16 | 0,42 | 0,17 | 0,40 | 0,17 | 0,22 | 0,17 | 0,72 | 0,14 |
| L3_P6 | 0,61 | 0,16 | 0,46 | 0,15 | 0,74 | 0,19 | 0,67 | 0,19 | 0,68 | 0,19 | 0,80 | 0,21 | 0,76 | 0,11 |
| L3_P1 | 0,78 | 0,28 | 0,87 | 0,36 | 0,96 | 0,57 | 0,97 | 0,61 | 0,96 | 0,57 | 0,97 | 0,65 | 0,89 | 0,20 |
| L3_P2 | 0,79 | 0,23 | 0,88 | 0,33 | 0,94 | 0,53 | 0,97 | 0,56 | 0,95 | 0,53 | 0,97 | 0,62 | 0,88 | 0,17 |
| **MOY** | **0,75** | **0,24** | **0,78** | **0,31** | **0,89** | **0,51** | **0,90** | **0,56** | **0,89** | **0,51** | **0,91** | **0,57** | **0,85** | **0,18** |
| +/- | -0,07 | -0,10 | -0,03 | -0,03 | 0,08 | 0,17 | 0,09 | 0,22 | 0,08 | 0,17 | 0,10 | 0,23 | 0,04 | -0,16 |

TABLE 3.13 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L4 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,89 | 0,34 | 0,98 | 0,54 | 0,89 | 0,34 | 0,89 | 0,36 | 0,85 | 0,20 | 0,94 | 0,52 | 0,98 | 0,58 |
| L4_P5 | 0,17 | 0,17 | 0,77 | 0,23 | 0,17 | 0,17 | 0,16 | 0,16 | 0,69 | 0,19 | 0,68 | 0,20 | 0,68 | 0,20 |
| L4_P6 | 0,54 | 0,18 | 0,96 | 0,39 | 0,55 | 0,18 | 0,57 | 0,18 | 0,58 | 0,14 | 0,69 | 0,19 | 0,94 | 0,37 |
| L4_P1 | 0,85 | 0,37 | 0,97 | 0,59 | 0,85 | 0,37 | 0,85 | 0,38 | 0,80 | 0,20 | 0,90 | 0,47 | 0,97 | 0,62 |
| L4_P2 | 0,84 | 0,29 | 0,98 | 0,55 | 0,85 | 0,30 | 0,86 | 0,32 | 0,77 | 0,16 | 0,93 | 0,43 | 0,98 | 0,61 |
| L4_P3 | 0,76 | 0,25 | 0,97 | 0,51 | 0,78 | 0,27 | 0,80 | 0,28 | 0,75 | 0,17 | 0,92 | 0,48 | 0,97 | 0,60 |
| **MOY** | **0,75** | **0,29** | **0,96** | **0,51** | **0,76** | **0,29** | **0,77** | **0,31** | **0,76** | **0,18** | **0,88** | **0,43** | **0,95** | **0,55** |
| +/- | 0,00 | 0,00 | 0,20 | 0,23 | 0,01 | 0,01 | 0,02 | 0,02 | 0,01 | -0,11 | 0,13 | 0,14 | 0,20 | 0,26 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,83 | 0,24 | 0,91 | 0,35 | 0,98 | 0,66 | 0,98 | 0,76 | 0,98 | 0,69 | 0,98 | 0,68 | 0,93 | 0,17 |
| L4_P5 | 0,46 | 0,16 | 0,26 | 0,17 | 0,43 | 0,16 | 0,44 | 0,18 | 0,42 | 0,17 | 0,23 | 0,17 | 0,86 | 0,11 |
| L4_P6 | 0,65 | 0,16 | 0,54 | 0,16 | 0,73 | 0,18 | 0,69 | 0,20 | 0,71 | 0,19 | 0,76 | 0,18 | 0,88 | 0,10 |
| L4_P1 | 0,79 | 0,27 | 0,89 | 0,39 | 0,95 | 0,55 | 0,97 | 0,62 | 0,96 | 0,58 | 0,97 | 0,62 | 0,92 | 0,16 |
| L4_P2 | 0,79 | 0,22 | 0,90 | 0,35 | 0,93 | 0,51 | 0,96 | 0,56 | 0,94 | 0,53 | 0,97 | 0,60 | 0,93 | 0,17 |
| L4_P3 | 0,78 | 0,24 | 0,84 | 0,31 | 0,90 | 0,46 | 0,94 | 0,57 | 0,90 | 0,47 | 0,96 | 0,55 | 0,91 | 0,16 |
| **MOY** | **0,76** | **0,23** | **0,80** | **0,32** | **0,88** | **0,47** | **0,90** | **0,55** | **0,88** | **0,49** | **0,90** | **0,53** | **0,91** | **0,16** |
| +/- | 0,00 | -0,06 | 0,05 | 0,03 | 0,13 | 0,19 | 0,14 | 0,26 | 0,13 | 0,21 | 0,14 | 0,24 | 0,16 | -0,13 |

TABLE 3.14 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L5 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L5_P5 | 0,92 | 0,40 | 0,95 | 0,30 | 0,91 | 0,38 | 0,86 | 0,33 | 0,95 | 0,30 | 0,94 | 0,30 | 0,94 | 0,21 |
| L5_P6 | 0,14 | 0,15 | 0,96 | 0,25 | 0,16 | 0,15 | 0,16 | 0,15 | 0,55 | 0,11 | 0,88 | 0,21 | 0,93 | 0,24 |
| L5_P1 | 0,12 | 0,16 | 0,94 | 0,31 | 0,12 | 0,16 | 0,12 | 0,16 | 0,61 | 0,07 | 0,86 | 0,23 | 0,93 | 0,29 |
| L5_P2 | 0,12 | 0,12 | 0,95 | 0,30 | 0,12 | 0,12 | 0,12 | 0,12 | 0,63 | 0,05 | 0,88 | 0,21 | 0,95 | 0,27 |
| L5_P3 | 0,09 | 0,14 | 0,95 | 0,29 | 0,10 | 0,15 | 0,10 | 0,14 | 0,61 | 0,06 | 0,87 | 0,21 | 0,94 | 0,27 |
| L5_P4 | 0,13 | 0,13 | 0,95 | 0,29 | 0,14 | 0,13 | 0,13 | 0,13 | 0,63 | 0,06 | 0,88 | 0,20 | 0,94 | 0,26 |
| **MOY** | **0,17** | **0,16** | **0,95** | **0,29** | **0,18** | **0,16** | **0,17** | **0,15** | **0,63** | **0,08** | **0,88** | **0,22** | **0,94** | **0,27** |
| +/- | 0,00 | 0,00 | 0,78 | 0,13 | 0,01 | 0,00 | 0,00 | 0,00 | 0,46 | -0,08 | 0,71 | 0,06 | 0,77 | 0,11 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L5_P5 | 0,91 | 0,21 | 0,93 | 0,22 | 0,95 | 0,11 | 0,97 | 0,24 | 0,96 | 0,17 | 0,96 | 0,40 | 0,94 | 0,13 |
| L5_P6 | 0,79 | 0,17 | 0,74 | 0,15 | 0,38 | 0,12 | 0,09 | 0,11 | 0,18 | 0,13 | 0,20 | 0,14 | 0,83 | 0,13 |
| L5_P1 | 0,78 | 0,20 | 0,71 | 0,16 | 0,38 | 0,05 | 0,36 | 0,07 | 0,35 | 0,05 | 0,27 | 0,12 | 0,84 | 0,13 |
| L5_P2 | 0,80 | 0,16 | 0,72 | 0,13 | 0,42 | 0,03 | 0,41 | 0,04 | 0,40 | 0,03 | 0,29 | 0,08 | 0,86 | 0,11 |
| L5_P3 | 0,79 | 0,18 | 0,72 | 0,15 | 0,40 | 0,04 | 0,41 | 0,06 | 0,39 | 0,04 | 0,29 | 0,11 | 0,84 | 0,12 |
| L5_P4 | 0,79 | 0,16 | 0,73 | 0,14 | 0,45 | 0,03 | 0,45 | 0,06 | 0,43 | 0,04 | 0,34 | 0,10 | 0,86 | 0,12 |
| **MOY** | **0,80** | **0,18** | **0,74** | **0,15** | **0,44** | **0,05** | **0,41** | **0,07** | **0,40** | **0,06** | **0,33** | **0,13** | **0,85** | **0,13** |
| +/- | 0,63 | 0,02 | 0,57 | -0,01 | 0,27 | -0,11 | 0,24 | -0,08 | 0,23 | -0,10 | 0,16 | -0,03 | 0,68 | -0,03 |

TABLE 3.15 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L6 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L6_P6 | 0,98 | 0,73 | 0,98 | 0,52 | 0,98 | 0,72 | 0,97 | 0,71 | 0,94 | 0,49 | 0,97 | 0,66 | 0,98 | 0,66 |
| L6_P1 | 0,74 | 0,28 | 0,97 | 0,58 | 0,74 | 0,28 | 0,74 | 0,29 | 0,57 | 0,17 | 0,75 | 0,27 | 0,95 | 0,52 |
| L6_P2 | 0,73 | 0,23 | 0,97 | 0,52 | 0,74 | 0,23 | 0,74 | 0,23 | 0,58 | 0,14 | 0,74 | 0,22 | 0,96 | 0,46 |
| L6_P3 | 0,73 | 0,25 | 0,97 | 0,50 | 0,74 | 0,25 | 0,74 | 0,25 | 0,55 | 0,15 | 0,74 | 0,25 | 0,96 | 0,50 |
| L6_P4 | 0,75 | 0,23 | 0,97 | 0,52 | 0,75 | 0,23 | 0,75 | 0,23 | 0,60 | 0,14 | 0,76 | 0,22 | 0,96 | 0,50 |
| L6_P5 | 0,15 | 0,17 | 0,75 | 0,24 | 0,16 | 0,17 | 0,14 | 0,17 | 0,54 | 0,18 | 0,62 | 0,20 | 0,67 | 0,21 |
| **MOY** | **0,73** | **0,30** | **0,96** | **0,51** | **0,73** | **0,30** | **0,73** | **0,30** | **0,61** | **0,19** | **0,76** | **0,29** | **0,94** | **0,50** |
| +/- | 0,00 | 0,00 | 0,23 | 0,21 | 0,00 | 0,00 | 0,00 | 0,00 | -0,11 | -0,11 | 0,04 | -0,01 | 0,21 | 0,20 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L6_P6 | 0,85 | 0,30 | 0,91 | 0,40 | 0,97 | 0,71 | 0,99 | 0,82 | 0,99 | 0,80 | 0,98 | 0,65 | 0,92 | 0,18 |
| L6_P1 | 0,61 | 0,22 | 0,60 | 0,23 | 0,73 | 0,25 | 0,72 | 0,28 | 0,73 | 0,28 | 0,87 | 0,37 | 0,84 | 0,15 |
| L6_P2 | 0,60 | 0,18 | 0,61 | 0,19 | 0,76 | 0,22 | 0,77 | 0,24 | 0,75 | 0,23 | 0,93 | 0,41 | 0,84 | 0,13 |
| L6_P3 | 0,62 | 0,20 | 0,58 | 0,21 | 0,75 | 0,23 | 0,75 | 0,27 | 0,72 | 0,25 | 0,94 | 0,45 | 0,83 | 0,14 |
| L6_P4 | 0,61 | 0,18 | 0,63 | 0,20 | 0,79 | 0,23 | 0,79 | 0,26 | 0,78 | 0,24 | 0,94 | 0,43 | 0,85 | 0,13 |
| L6_P5 | 0,43 | 0,16 | 0,27 | 0,17 | 0,37 | 0,18 | 0,30 | 0,18 | 0,29 | 0,18 | 0,17 | 0,17 | 0,76 | 0,14 |
| **MOY** | **0,63** | **0,20** | **0,62** | **0,23** | **0,75** | **0,28** | **0,75** | **0,32** | **0,74** | **0,31** | **0,87** | **0,43** | **0,84** | **0,14** |
| +/- | -0,10 | -0,09 | -0,11 | -0,07 | 0,03 | -0,01 | 0,02 | 0,02 | 0,02 | 0,01 | 0,15 | 0,13 | 0,12 | -0,16 |

## 2 Codebook RGB : résultats détaillés L1 à L4 pour la base “JARDIN”

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,97 | 0,23 | 0,99 | 0,57 | 0,97 | 0,29 | 0,97 | 0,29 | 0,89 | 0,18 | 0,97 | 0,30 | 0,98 | 0,56 |
| L1_P2 | 0,89 | 0,05 | 0,93 | 0,19 | 0,83 | 0,08 | 0,83 | 0,08 | 0,61 | 0,09 | 0,89 | 0,07 | 0,76 | 0,10 |
| L1_P3 | 0,96 | 0,15 | 0,97 | 0,30 | 0,92 | 0,13 | 0,92 | 0,13 | 0,69 | 0,11 | 0,93 | 0,14 | 0,89 | 0,16 |
| L1_P4 | 0,99 | 0,08 | 0,98 | 0,17 | 0,94 | 0,05 | 0,94 | 0,05 | 0,72 | 0,02 | 0,94 | 0,05 | 0,96 | 0,10 |
| **MOY** | **0,96** | **0,11** | **0,97** | **0,26** | **0,92** | **0,11** | **0,92** | **0,11** | **0,72** | **0,08** | **0,93** | **0,11** | **0,92** | **0,19** |
| +/- | 0,00 | 0,00 | 0,01 | 0,15 | -0,04 | -0,01 | -0,04 | -0,01 | -0,24 | -0,04 | -0,03 | 0,00 | -0,05 | 0,07 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,88 | 0,17 | 0,96 | 0,33 | 0,98 | 0,44 | 0,98 | 0,43 | 0,98 | 0,43 | 0,98 | 0,50 | 0,92 | 0,21 |
| L1_P2 | 0,73 | 0,08 | 0,32 | 0,09 | 0,78 | 0,10 | 0,78 | 0,10 | 0,77 | 0,10 | 0,77 | 0,11 | 0,56 | 0,08 |
| L1_P3 | 0,78 | 0,11 | 0,37 | 0,09 | 0,91 | 0,20 | 0,91 | 0,20 | 0,90 | 0,19 | 0,93 | 0,22 | 0,54 | 0,07 |
| L1_P4 | 0,82 | 0,03 | 0,55 | 0,02 | 0,93 | 0,06 | 0,93 | 0,07 | 0,93 | 0,06 | 0,94 | 0,08 | 0,60 | 0,02 |
| **MOY** | **0,80** | **0,08** | **0,54** | **0,10** | **0,91** | **0,16** | **0,91** | **0,16** | **0,90** | **0,15** | **0,92** | **0,18** | **0,64** | **0,07** |
| +/- | -0,16 | -0,04 | -0,42 | -0,02 | -0,06 | 0,04 | -0,05 | 0,04 | -0,06 | 0,04 | -0,05 | 0,07 | -0,33 | -0,04 |

TABLE 3.16 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L1 (base “JARDIN”) et les paramétrages optimaux de l’algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,96 | 0,05 | 0,96 | 0,28 | 0,96 | 0,10 | 0,96 | 0,10 | 0,94 | 0,11 | 0,96 | 0,09 | 0,96 | 0,25 |
| L2_P3 | 0,97 | 0,15 | 0,98 | 0,36 | 0,92 | 0,12 | 0,92 | 0,12 | 0,89 | 0,13 | 0,95 | 0,15 | 0,97 | 0,28 |
| L2_P4 | 0,98 | 0,06 | 0,98 | 0,18 | 0,93 | 0,04 | 0,93 | 0,04 | 0,88 | 0,04 | 0,95 | 0,06 | 0,97 | 0,11 |
| L2_P1 | 0,93 | 0,09 | 0,96 | 0,30 | 0,86 | 0,12 | 0,86 | 0,12 | 0,74 | 0,12 | 0,90 | 0,16 | 0,89 | 0,18 |
| **MOY** | **0,97** | **0,08** | **0,98** | **0,25** | **0,92** | **0,08** | **0,92** | **0,08** | **0,87** | **0,08** | **0,94** | **0,10** | **0,95** | **0,17** |
| +/- | 0,00 | 0,00 | 0,01 | 0,17 | -0,04 | 0,00 | -0,04 | 0,00 | -0,10 | 0,00 | -0,02 | 0,02 | -0,01 | 0,09 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,88 | 0,12 | 0,93 | 0,16 | 0,95 | 0,12 | 0,95 | 0,13 | 0,95 | 0,13 | 0,95 | 0,20 | 0,86 | 0,08 |
| L2_P3 | 0,79 | 0,11 | 0,89 | 0,16 | 0,93 | 0,15 | 0,93 | 0,16 | 0,93 | 0,15 | 0,93 | 0,20 | 0,80 | 0,07 |
| L2_P4 | 0,75 | 0,03 | 0,89 | 0,05 | 0,94 | 0,06 | 0,94 | 0,06 | 0,94 | 0,06 | 0,96 | 0,09 | 0,81 | 0,02 |
| L2_P1 | 0,66 | 0,11 | 0,51 | 0,10 | 0,85 | 0,15 | 0,85 | 0,15 | 0,86 | 0,16 | 0,85 | 0,17 | 0,78 | 0,09 |
| **MOY** | **0,77** | **0,07** | **0,83** | **0,10** | **0,93** | **0,10** | **0,92** | **0,10** | **0,93** | **0,11** | **0,93** | **0,14** | **0,82** | **0,05** |
| +/- | -0,20 | -0,01 | -0,13 | 0,02 | -0,04 | 0,02 | -0,04 | 0,02 | -0,04 | 0,02 | -0,03 | 0,06 | -0,15 | -0,03 |

TABLE 3.17 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L2 (base "JARDIN") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,98 | 0,16 | 0,98 | 0,33 | 0,97 | 0,20 | 0,97 | 0,19 | 0,87 | 0,12 | 0,97 | 0,20 | 0,96 | 0,25 |
| L3_P4 | 0,99 | 0,08 | 0,99 | 0,18 | 0,98 | 0,08 | 0,98 | 0,08 | 0,86 | 0,03 | 0,97 | 0,09 | 0,97 | 0,12 |
| L3_P1 | 0,94 | 0,10 | 0,96 | 0,30 | 0,92 | 0,15 | 0,92 | 0,15 | 0,76 | 0,12 | 0,91 | 0,17 | 0,91 | 0,20 |
| L3_P2 | 0,95 | 0,10 | 0,96 | 0,28 | 0,93 | 0,12 | 0,93 | 0,11 | 0,89 | 0,08 | 0,94 | 0,11 | 0,92 | 0,18 |
| **MOY** | **0,97** | **0,10** | **0,97** | **0,24** | **0,96** | **0,12** | **0,96** | **0,12** | **0,85** | **0,07** | **0,96** | **0,13** | **0,95** | **0,17** |
| +/- | 0,00 | 0,00 | 0,00 | 0,14 | -0,01 | 0,02 | -0,01 | 0,02 | -0,12 | -0,03 | -0,02 | 0,02 | -0,02 | 0,06 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,85 | 0,11 | 0,51 | 0,09 | 0,94 | 0,17 | 0,95 | 0,19 | 0,94 | 0,16 | 0,96 | 0,27 | 0,57 | 0,07 |
| L3_P4 | 0,80 | 0,03 | 0,71 | 0,03 | 0,95 | 0,07 | 0,95 | 0,07 | 0,95 | 0,07 | 0,97 | 0,10 | 0,65 | 0,02 |
| L3_P1 | 0,71 | 0,11 | 0,68 | 0,12 | 0,86 | 0,16 | 0,86 | 0,16 | 0,87 | 0,17 | 0,87 | 0,18 | 0,79 | 0,13 |
| L3_P2 | 0,82 | 0,10 | 0,68 | 0,10 | 0,94 | 0,13 | 0,95 | 0,14 | 0,94 | 0,13 | 0,95 | 0,20 | 0,59 | 0,08 |
| **MOY** | **0,80** | **0,07** | **0,66** | **0,07** | **0,93** | **0,11** | **0,94** | **0,12** | **0,93** | **0,11** | **0,95** | **0,16** | **0,65** | **0,06** |
| +/- | -0,17 | -0,03 | -0,31 | -0,03 | -0,04 | 0,01 | -0,04 | 0,02 | -0,04 | 0,01 | -0,03 | 0,06 | -0,33 | -0,04 |

TABLE 3.18 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L3 (base "JARDIN") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,99 | 0,14 | 0,99 | 0,28 | 0,99 | 0,13 | 0,99 | 0,13 | 0,87 | 0,03 | 0,99 | 0,13 | 0,99 | 0,19 |
| L4_P1 | 0,95 | 0,12 | 0,96 | 0,30 | 0,91 | 0,15 | 0,91 | 0,15 | 0,75 | 0,11 | 0,91 | 0,17 | 0,93 | 0,24 |
| L4_P2 | 0,92 | 0,07 | 0,95 | 0,23 | 0,89 | 0,09 | 0,89 | 0,09 | 0,84 | 0,07 | 0,92 | 0,10 | 0,87 | 0,14 |
| L4_P3 | 0,97 | 0,18 | 0,98 | 0,34 | 0,95 | 0,16 | 0,95 | 0,16 | 0,83 | 0,11 | 0,95 | 0,17 | 0,95 | 0,23 |
| **MOY** | **0,97** | **0,13** | **0,98** | **0,28** | **0,95** | **0,13** | **0,95** | **0,13** | **0,84** | **0,07** | **0,96** | **0,14** | **0,95** | **0,20** |
| +/- | 0,00 | 0,00 | 0,01 | 0,15 | -0,02 | 0,00 | -0,02 | 0,00 | -0,13 | -0,06 | -0,02 | 0,01 | -0,02 | 0,06 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,88 | 0,04 | 0,82 | 0,04 | 0,95 | 0,06 | 0,94 | 0,06 | 0,94 | 0,06 | 0,98 | 0,15 | 0,73 | 0,02 |
| L4_P1 | 0,73 | 0,11 | 0,66 | 0,12 | 0,85 | 0,15 | 0,86 | 0,15 | 0,87 | 0,16 | 0,87 | 0,19 | 0,75 | 0,11 |
| L4_P2 | 0,79 | 0,09 | 0,73 | 0,11 | 0,91 | 0,10 | 0,91 | 0,11 | 0,91 | 0,10 | 0,91 | 0,14 | 0,62 | 0,08 |
| L4_P3 | 0,80 | 0,11 | 0,74 | 0,12 | 0,94 | 0,17 | 0,94 | 0,18 | 0,93 | 0,17 | 0,96 | 0,26 | 0,61 | 0,08 |
| **MOY** | **0,83** | **0,07** | **0,76** | **0,08** | **0,92** | **0,10** | **0,92** | **0,11** | **0,92** | **0,10** | **0,95** | **0,18** | **0,69** | **0,06** |
| +/- | -0,15 | -0,06 | -0,21 | -0,05 | -0,05 | -0,03 | -0,05 | -0,03 | -0,05 | -0,03 | -0,02 | 0,04 | -0,28 | -0,07 |

TABLE 3.19 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L4 (base "JARDIN") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

# 3 Codebook RGB : résultats détaillés L1 à L5 pour la base “RUE”

TABLE 3.20 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L1 (base “RUE”) et les paramétrages optimaux de l’algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,99 | 0,07 | 1,00 | 0,00 | 0,99 | 0,09 | 0,99 | 0,11 | 0,99 | 0,19 | 1,00 | 0,16 | 1,00 | 0,02 |
| L1_P2 | 0,99 | 0,19 | 0,99 | 0,03 | 0,98 | 0,19 | 0,98 | 0,20 | 0,88 | 0,08 | 0,98 | 0,23 | 0,95 | 0,03 |
| L1_P3 | 0,83 | 0,06 | 0,98 | 0,10 | 0,85 | 0,07 | 0,85 | 0,08 | 0,72 | 0,09 | 0,97 | 0,20 | 0,97 | 0,06 |
| L1_P4 | 0,98 | 0,20 | 0,99 | 0,01 | 0,97 | 0,18 | 0,97 | 0,18 | 0,85 | 0,07 | 0,97 | 0,16 | 0,99 | 0,07 |
| L1_P5 | 0,94 | 0,15 | 0,99 | 0,04 | 0,96 | 0,20 | 0,95 | 0,19 | 0,83 | 0,07 | 0,96 | 0,19 | 0,99 | 0,12 |
| **MOY** | **0,94** | **0,13** | **0,99** | **0,04** | **0,95** | **0,15** | **0,95** | **0,15** | **0,85** | **0,10** | **0,98** | **0,19** | **0,98** | **0,06** |
| +/- | 0,00 | 0,00 | 0,05 | -0,10 | 0,01 | 0,01 | 0,01 | 0,01 | -0,09 | -0,04 | 0,03 | 0,05 | 0,04 | -0,07 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L1_P1 | 0,90 | 0,03 | 0,99 | 0,03 | 1,00 | 0,00 | 1,00 | 0,00 | 1,00 | 0,00 | 1,00 | 0,00 | 0,97 | 0,06 |
| L1_P2 | 0,76 | 0,04 | 0,99 | 0,08 | 0,99 | 0,03 | 0,99 | 0,06 | 0,99 | 0,06 | 0,99 | 0,04 | 0,67 | 0,03 |
| L1_P3 | 0,70 | 0,07 | 0,88 | 0,06 | 0,89 | 0,05 | 0,92 | 0,06 | 0,90 | 0,05 | 0,98 | 0,11 | 0,71 | 0,07 |
| L1_P4 | 0,72 | 0,03 | 0,98 | 0,12 | 0,99 | 0,05 | 0,99 | 0,06 | 0,99 | 0,06 | 0,99 | 0,23 | 0,74 | 0,03 |
| L1_P5 | 0,71 | 0,04 | 0,97 | 0,13 | 0,99 | 0,01 | 0,99 | 0,01 | 0,99 | 0,02 | 0,99 | 0,06 | 0,74 | 0,03 |
| **MOY** | **0,76** | **0,04** | **0,96** | **0,09** | **0,97** | 0,03 | **0,98** | **0,04** | **0,97** | **0,04** | **0,99** | **0,09** | **0,76** | **0,04** |
| +/- | -0,19 | -0,09 | 0,02 | -0,05 | 0,03 | -0,11 | 0,03 | -0,10 | 0,03 | -0,10 | 0,05 | -0,04 | -0,18 | -0,09 |

TABLE 3.21 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L2 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,99 | 0,26 | 0,99 | 0,05 | 0,99 | 0,29 | 0,99 | 0,30 | 0,99 | 0,39 | 1,00 | 0,53 | 0,99 | 0,08 |
| L2_P3 | 0,81 | 0,07 | 0,98 | 0,11 | 0,84 | 0,08 | 0,84 | 0,08 | 0,73 | 0,09 | 0,98 | 0,38 | 0,98 | 0,14 |
| L2_P4 | 0,95 | 0,12 | 0,99 | 0,01 | 0,95 | 0,15 | 0,95 | 0,15 | 0,86 | 0,07 | 0,98 | 0,20 | 0,99 | 0,09 |
| L2_P5 | 0,95 | 0,19 | 0,99 | 0,06 | 0,96 | 0,21 | 0,96 | 0,21 | 0,85 | 0,09 | 0,97 | 0,25 | 0,99 | 0,18 |
| L2_P1 | 0,94 | 0,02 | 1,00 | 0,00 | 0,95 | 0,02 | 0,94 | 0,02 | 0,86 | 0,03 | 0,96 | 0,03 | 0,99 | 0,02 |
| **MOY** | **0,93** | **0,13** | **0,99** | **0,05** | **0,94** | **0,15** | **0,94** | **0,15** | **0,86** | **0,13** | **0,98** | **0,27** | **0,99** | **0,10** |
| +/- | 0,00 | 0,00 | 0,06 | -0,08 | 0,01 | 0,02 | 0,01 | 0,02 | -0,07 | 0,00 | 0,05 | 0,14 | 0,06 | -0,03 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L2_P2 | 0,96 | 0,17 | 0,99 | 0,09 | 0,99 | 0,11 | 0,99 | 0,09 | 0,99 | 0,08 | 0,99 | 0,07 | 0,99 | 0,48 |
| L2_P3 | 0,81 | 0,10 | 0,94 | 0,15 | 0,92 | 0,07 | 0,93 | 0,09 | 0,91 | 0,07 | 0,98 | 0,14 | 0,25 | 0,06 |
| L2_P4 | 0,83 | 0,05 | 0,97 | 0,15 | 0,99 | 0,07 | 0,99 | 0,06 | 0,98 | 0,06 | 1,00 | 0,22 | 0,28 | 0,02 |
| L2_P5 | 0,80 | 0,07 | 0,94 | 0,14 | 0,99 | 0,01 | 0,98 | 0,01 | 0,99 | 0,02 | 0,99 | 0,11 | 0,22 | 0,03 |
| L2_P1 | 0,71 | 0,01 | 0,87 | 0,02 | 0,99 | 0,00 | 0,99 | 0,00 | 0,99 | 0,00 | 1,00 | 0,04 | 0,35 | 0,01 |
| **MOY** | **0,82** | **0,08** | **0,94** | **0,11** | **0,98** | **0,05** | **0,98** | **0,05** | **0,97** | **0,04** | **0,99** | **0,12** | **0,41** | **0,11** |
| +/- | -0,11 | -0,05 | 0,01 | -0,02 | 0,05 | -0,08 | 0,05 | -0,08 | 0,04 | -0,08 | 0,06 | -0,01 | -0,52 | -0,02 |

TABLE 3.22 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L3 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,99 | 0,36 | 0,98 | 0,12 | 0,99 | 0,40 | 0,99 | 0,40 | 0,98 | 0,53 | 0,98 | 0,52 | 0,98 | 0,15 |
| L3_P4 | 0,99 | 0,29 | 0,99 | 0,02 | 0,99 | 0,30 | 0,99 | 0,27 | 0,88 | 0,10 | 0,99 | 0,29 | 0,99 | 0,08 |
| L3_P5 | 0,93 | 0,18 | 0,99 | 0,06 | 0,96 | 0,31 | 0,96 | 0,30 | 0,83 | 0,09 | 0,98 | 0,32 | 0,99 | 0,14 |
| L3_P1 | 0,98 | 0,11 | 1,00 | 0,00 | 0,98 | 0,09 | 0,98 | 0,09 | 0,86 | 0,03 | 0,96 | 0,04 | 0,99 | 0,01 |
| L3_P2 | 0,99 | 0,43 | 0,99 | 0,04 | 0,99 | 0,44 | 0,99 | 0,42 | 0,90 | 0,10 | 0,98 | 0,24 | 0,99 | 0,07 |
| **MOY** | **0,97** | **0,27** | **0,99** | **0,05** | **0,98** | **0,30** | **0,98** | **0,29** | **0,89** | **0,16** | **0,98** | **0,27** | **0,99** | **0,09** |
| +/- | 0,00 | 0,00 | 0,02 | -0,22 | 0,01 | 0,03 | 0,01 | 0,02 | -0,09 | -0,10 | 0,01 | 0,01 | 0,02 | -0,18 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L3_P3 | 0,95 | 0,22 | 0,98 | 0,45 | 0,99 | 0,45 | 0,99 | 0,47 | 0,99 | 0,48 | 0,98 | 0,13 | 0,97 | 0,29 |
| L3_P4 | 0,82 | 0,04 | 0,96 | 0,13 | 0,97 | 0,19 | 0,99 | 0,25 | 0,99 | 0,24 | 1,00 | 0,26 | 0,88 | 0,05 |
| L3_P5 | 0,73 | 0,06 | 0,87 | 0,09 | 0,99 | 0,05 | 0,97 | 0,03 | 0,97 | 0,04 | 0,99 | 0,13 | 0,80 | 0,04 |
| L3_P1 | 0,66 | 0,01 | 0,78 | 0,01 | 0,99 | 0,02 | 0,99 | 0,00 | 0,99 | 0,01 | 1,00 | 0,04 | 0,60 | 0,01 |
| L3_P2 | 0,80 | 0,05 | 0,98 | 0,07 | 0,99 | 0,21 | 0,99 | 0,25 | 0,99 | 0,30 | 0,99 | 0,05 | 0,48 | 0,02 |
| **MOY** | **0,78** | **0,07** | **0,91** | **0,14** | **0,99** | **0,18** | **0,99** | **0,19** | **0,99** | **0,21** | **0,99** | **0,12** | **0,75** | **0,08** |
| +/- | -0,19 | -0,20 | -0,06 | -0,12 | 0,01 | -0,09 | 0,01 | -0,07 | 0,01 | -0,06 | 0,02 | -0,14 | -0,23 | -0,19 |

TABLE 3.23 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L4 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 1,00 | 0,42 | 0,99 | 0,01 | 0,99 | 0,42 | 0,99 | 0,38 | 0,98 | 0,38 | 1,00 | 0,53 | 0,99 | 0,11 |
| L4_P5 | 0,99 | 0,50 | 0,99 | 0,04 | 0,98 | 0,33 | 0,98 | 0,30 | 0,92 | 0,13 | 0,98 | 0,26 | 0,99 | 0,11 |
| L4_P1 | 0,85 | 0,02 | 1,00 | 0,02 | 0,86 | 0,02 | 0,85 | 0,02 | 0,83 | 0,04 | 0,96 | 0,07 | 0,99 | 0,05 |
| L4_P2 | 0,81 | 0,03 | 0,99 | 0,06 | 0,80 | 0,03 | 0,79 | 0,03 | 0,85 | 0,08 | 0,97 | 0,15 | 0,99 | 0,06 |
| L4_P3 | 0,48 | 0,03 | 0,98 | 0,07 | 0,53 | 0,04 | 0,50 | 0,04 | 0,73 | 0,08 | 0,95 | 0,15 | 0,98 | 0,09 |
| **MOY** | **0,83** | **0,21** | **0,99** | **0,04** | **0,84** | **0,17** | **0,83** | **0,16** | **0,87** | **0,15** | **0,97** | **0,24** | **0,99** | **0,08** |
| +/- | 0,00 | 0,00 | 0,16 | -0,17 | 0,01 | -0,03 | 0,00 | -0,05 | 0,03 | -0,06 | 0,14 | 0,03 | 0,16 | -0,12 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L4_P4 | 0,94 | 0,12 | 0,98 | 0,17 | 0,99 | 0,02 | 0,99 | 0,02 | 0,99 | 0,02 | 0,99 | 0,13 | 0,97 | 0,14 |
| L4_P5 | 0,85 | 0,07 | 0,95 | 0,08 | 0,99 | 0,01 | 0,98 | 0,02 | 0,98 | 0,02 | 0,99 | 0,03 | 0,87 | 0,04 |
| L4_P1 | 0,70 | 0,03 | 0,88 | 0,05 | 0,99 | 0,00 | 0,99 | 0,01 | 0,99 | 0,01 | 1,00 | 0,03 | 0,69 | 0,02 |
| L4_P2 | 0,85 | 0,06 | 0,97 | 0,07 | 0,98 | 0,07 | 0,98 | 0,07 | 0,98 | 0,08 | 0,99 | 0,08 | 0,68 | 0,04 |
| L4_P3 | 0,81 | 0,08 | 0,87 | 0,06 | 0,90 | 0,03 | 0,91 | 0,04 | 0,90 | 0,04 | 0,98 | 0,10 | 0,92 | 0,11 |
| **MOY** | **0,83** | **0,07** | **0,93** | **0,09** | **0,97** | **0,03** | **0,97** | **0,03** | **0,97** | **0,03** | **0,99** | **0,07** | **0,83** | **0,07** |
| +/- | 0,00 | -0,13 | 0,10 | -0,12 | 0,14 | -0,18 | 0,14 | -0,18 | 0,14 | -0,17 | 0,16 | -0,13 | -0,01 | -0,14 |

TABLE 3.24 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L5 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L5_P5 | 1,00 | 0,51 | 0,99 | 0,05 | 1,00 | 0,58 | 1,00 | 0,60 | 0,99 | 0,49 | 1,00 | 0,64 | 0,99 | 0,08 |
| L5_P1 | 0,81 | 0,02 | 1,00 | 0,02 | 0,92 | 0,03 | 0,91 | 0,03 | 0,84 | 0,04 | 0,97 | 0,06 | 0,99 | 0,04 |
| L5_P2 | 0,83 | 0,03 | 0,99 | 0,05 | 0,92 | 0,07 | 0,92 | 0,07 | 0,85 | 0,07 | 0,97 | 0,17 | 0,99 | 0,06 |
| L5_P3 | 0,64 | 0,05 | 0,98 | 0,07 | 0,74 | 0,06 | 0,71 | 0,06 | 0,72 | 0,09 | 0,96 | 0,20 | 0,98 | 0,07 |
| L5_P4 | 0,86 | 0,04 | 0,99 | 0,01 | 0,97 | 0,21 | 0,96 | 0,21 | 0,90 | 0,10 | 0,98 | 0,22 | 0,99 | 0,08 |
| **MOY** | **0,83** | **0,13** | **0,99** | **0,04** | **0,91** | **0,19** | **0,91** | **0,20** | **0,86** | **0,16** | **0,98** | **0,26** | **0,99** | **0,07** |
| +/- | 0,00 | 0,00 | 0,16 | -0,09 | 0,08 | 0,06 | 0,07 | 0,07 | 0,03 | 0,03 | 0,15 | 0,13 | 0,16 | -0,06 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L5_P5 | 0,94 | 0,15 | 0,99 | 0,16 | 0,99 | 0,04 | 0,99 | 0,01 | 0,99 | 0,04 | 0,99 | 0,14 | 0,97 | 0,17 |
| L5_P1 | 0,71 | 0,02 | 0,94 | 0,04 | 0,99 | 0,00 | 0,99 | 0,00 | 0,99 | 0,00 | 1,00 | 0,06 | 0,61 | 0,02 |
| L5_P2 | 0,83 | 0,05 | 0,98 | 0,05 | 0,98 | 0,05 | 0,98 | 0,07 | 0,98 | 0,06 | 0,99 | 0,07 | 0,49 | 0,02 |
| L5_P3 | 0,73 | 0,08 | 0,85 | 0,05 | 0,91 | 0,06 | 0,92 | 0,06 | 0,91 | 0,06 | 0,98 | 0,10 | 0,79 | 0,08 |
| L5_P4 | 0,83 | 0,05 | 0,97 | 0,09 | 0,99 | 0,06 | 0,99 | 0,06 | 0,99 | 0,05 | 1,00 | 0,23 | 0,81 | 0,04 |
| **MOY** | **0,81** | **0,07** | **0,95** | **0,08** | **0,98** | **0,04** | **0,98** | **0,04** | **0,97** | **0,04** | **0,99** | **0,12** | **0,73** | **0,07** |
| +/- | -0,03 | -0,06 | 0,12 | -0,05 | 0,14 | -0,09 | 0,14 | -0,09 | 0,14 | -0,09 | 0,16 | -0,01 | -0,10 | -0,06 |

# 4 Codebook RGB-D : résultats détaillés L1 à L6 pour la base "BUREAU"

TABLE 3.25 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L1 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,85 | 0,40 | 0,99 | 0,81 | 0,85 | 0,41 | 0,86 | 0,41 | 0,79 | 0,29 | 0,94 | 0,64 | 0,99 | 0,81 |
| L1_P2 | 0,84 | 0,31 | 0,98 | 0,73 | 0,84 | 0,31 | 0,85 | 0,32 | 0,76 | 0,22 | 0,93 | 0,46 | 0,98 | 0,71 |
| L1_P3 | 0,77 | 0,28 | 0,98 | 0,74 | 0,78 | 0,29 | 0,79 | 0,30 | 0,73 | 0,23 | 0,93 | 0,55 | 0,98 | 0,72 |
| L1_P4 | 0,85 | 0,32 | 0,98 | 0,63 | 0,85 | 0,32 | 0,85 | 0,32 | 0,79 | 0,24 | 0,93 | 0,48 | 0,97 | 0,60 |
| L1_P5 | 0,08 | 0,17 | 0,65 | 0,23 | 0,08 | 0,17 | 0,09 | 0,17 | 0,66 | 0,22 | 0,57 | 0,21 | 0,54 | 0,21 |
| L1_P6 | 0,45 | 0,18 | 0,91 | 0,38 | 0,46 | 0,18 | 0,47 | 0,18 | 0,53 | 0,17 | 0,64 | 0,21 | 0,87 | 0,35 |
| **MOY** | **0,74** | **0,30** | **0,95** | **0,66** | **0,74** | **0,30** | **0,74** | **0,31** | **0,73** | **0,24** | **0,88** | **0,48** | **0,94** | **0,64** |
| +/- | 0,00 | 0,00 | 0,22 | 0,36 | 0,00 | 0,00 | 0,01 | 0,01 | 0,00 | -0,06 | 0,14 | 0,18 | 0,20 | 0,34 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,91 | 0,49 | 0,96 | 0,68 | 0,96 | 0,75 | 0,98 | 0,80 | 0,96 | 0,76 | 0,99 | 0,83 | 0,93 | 0,50 |
| L1_P2 | 0,89 | 0,37 | 0,95 | 0,56 | 0,93 | 0,61 | 0,96 | 0,67 | 0,94 | 0,63 | 0,98 | 0,72 | 0,91 | 0,38 |
| L1_P3 | 0,88 | 0,39 | 0,92 | 0,50 | 0,91 | 0,56 | 0,95 | 0,64 | 0,91 | 0,57 | 0,98 | 0,70 | 0,87 | 0,35 |
| L1_P4 | 0,88 | 0,36 | 0,94 | 0,50 | 0,98 | 0,64 | 0,98 | 0,68 | 0,98 | 0,66 | 0,98 | 0,67 | 0,90 | 0,35 |
| L1_P5 | 0,39 | 0,19 | 0,08 | 0,17 | 0,35 | 0,18 | 0,34 | 0,18 | 0,33 | 0,18 | 0,16 | 0,17 | 0,48 | 0,18 |
| L1_P6 | 0,72 | 0,24 | 0,58 | 0,20 | 0,69 | 0,22 | 0,67 | 0,22 | 0,68 | 0,22 | 0,72 | 0,23 | 0,71 | 0,22 |
| **MOY** | **0,84** | **0,38** | **0,85** | **0,50** | **0,87** | **0,56** | **0,89** | **0,61** | **0,87** | **0,57** | **0,90** | **0,64** | **0,85** | **0,36** |
| +/- | 0,10 | 0,07 | 0,11 | 0,19 | 0,14 | 0,26 | 0,16 | 0,31 | 0,14 | 0,27 | 0,16 | 0,34 | 0,12 | 0,06 |

TABLE 3.26 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L2 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,95 | 0,57 | 0,99 | 0,82 | 0,95 | 0,57 | 0,95 | 0,56 | 0,86 | 0,31 | 0,96 | 0,65 | 0,99 | 0,82 |
| L2_P3 | 0,90 | 0,41 | 0,98 | 0,75 | 0,89 | 0,41 | 0,89 | 0,40 | 0,79 | 0,27 | 0,94 | 0,57 | 0,98 | 0,72 |
| L2_P4 | 0,92 | 0,42 | 0,95 | 0,50 | 0,92 | 0,41 | 0,91 | 0,41 | 0,78 | 0,24 | 0,89 | 0,37 | 0,95 | 0,49 |
| L2_P5 | 0,04 | 0,17 | 0,61 | 0,22 | 0,04 | 0,17 | 0,05 | 0,17 | 0,61 | 0,21 | 0,55 | 0,20 | 0,50 | 0,20 |
| L2_P6 | 0,58 | 0,20 | 0,87 | 0,33 | 0,58 | 0,20 | 0,58 | 0,20 | 0,48 | 0,17 | 0,61 | 0,20 | 0,84 | 0,31 |
| L2_P1 | 0,91 | 0,48 | 0,96 | 0,61 | 0,91 | 0,47 | 0,90 | 0,46 | 0,77 | 0,26 | 0,89 | 0,44 | 0,96 | 0,60 |
| **MOY** | **0,82** | **0,42** | **0,94** | **0,61** | **0,82** | **0,42** | **0,82** | **0,41** | **0,75** | **0,26** | **0,86** | **0,46** | **0,92** | **0,60** |
| +/- | 0,00 | 0,00 | 0,11 | 0,19 | 0,00 | 0,00 | -0,01 | -0,01 | -0,07 | -0,17 | 0,04 | 0,04 | 0,10 | 0,17 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,93 | 0,49 | 0,97 | 0,69 | 0,95 | 0,75 | 0,98 | 0,80 | 0,96 | 0,78 | 0,99 | 0,81 | 0,95 | 0,52 |
| L2_P3 | 0,89 | 0,41 | 0,92 | 0,48 | 0,92 | 0,60 | 0,97 | 0,67 | 0,93 | 0,62 | 0,98 | 0,73 | 0,88 | 0,37 |
| L2_P4 | 0,86 | 0,33 | 0,92 | 0,43 | 0,95 | 0,50 | 0,96 | 0,53 | 0,95 | 0,50 | 0,96 | 0,52 | 0,87 | 0,31 |
| L2_P5 | 0,36 | 0,18 | 0,05 | 0,17 | 0,29 | 0,17 | 0,29 | 0,18 | 0,28 | 0,18 | 0,11 | 0,17 | 0,44 | 0,18 |
| L2_P6 | 0,69 | 0,23 | 0,56 | 0,20 | 0,66 | 0,21 | 0,63 | 0,21 | 0,64 | 0,21 | 0,69 | 0,22 | 0,68 | 0,21 |
| L2_P1 | 0,88 | 0,42 | 0,93 | 0,53 | 0,95 | 0,58 | 0,96 | 0,61 | 0,96 | 0,59 | 0,96 | 0,62 | 0,89 | 0,39 |
| **MOY** | **0,84** | **0,38** | **0,83** | **0,47** | **0,87** | **0,54** | **0,88** | **0,57** | **0,87** | **0,55** | **0,88** | **0,59** | **0,84** | **0,36** |
| +/- | 0,01 | -0,04 | 0,01 | 0,05 | 0,05 | 0,12 | 0,06 | 0,15 | 0,05 | 0,13 | 0,06 | 0,17 | 0,02 | -0,06 |

TABLE 3.27 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L3 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,93 | 0,54 | 0,98 | 0,79 | 0,93 | 0,54 | 0,93 | 0,54 | 0,84 | 0,36 | 0,93 | 0,57 | 0,99 | 0,79 |
| L3_P4 | 0,87 | 0,35 | 0,93 | 0,44 | 0,87 | 0,35 | 0,87 | 0,35 | 0,76 | 0,22 | 0,84 | 0,30 | 0,93 | 0,43 |
| L3_P5 | 0,03 | 0,17 | 0,58 | 0,21 | 0,03 | 0,17 | 0,03 | 0,17 | 0,59 | 0,20 | 0,50 | 0,20 | 0,46 | 0,20 |
| L3_P6 | 0,55 | 0,20 | 0,85 | 0,30 | 0,55 | 0,20 | 0,55 | 0,20 | 0,44 | 0,16 | 0,58 | 0,19 | 0,83 | 0,29 |
| L3_P1 | 0,88 | 0,42 | 0,94 | 0,53 | 0,88 | 0,42 | 0,87 | 0,42 | 0,75 | 0,25 | 0,84 | 0,35 | 0,94 | 0,53 |
| L3_P2 | 0,91 | 0,42 | 0,97 | 0,63 | 0,91 | 0,41 | 0,90 | 0,41 | 0,80 | 0,24 | 0,88 | 0,35 | 0,97 | 0,63 |
| **MOY** | **0,80** | **0,39** | **0,92** | **0,54** | **0,80** | **0,39** | **0,80** | **0,39** | **0,74** | **0,25** | **0,81** | **0,36** | **0,91** | **0,54** |
| +/- | 0,00 | 0,00 | 0,12 | 0,15 | 0,00 | 0,00 | 0,00 | 0,00 | -0,07 | -0,14 | 0,01 | -0,03 | 0,11 | 0,15 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,92 | 0,50 | 0,95 | 0,67 | 0,95 | 0,74 | 0,96 | 0,80 | 0,95 | 0,76 | 0,99 | 0,80 | 0,91 | 0,43 |
| L3_P4 | 0,84 | 0,30 | 0,89 | 0,37 | 0,93 | 0,43 | 0,93 | 0,45 | 0,93 | 0,43 | 0,93 | 0,44 | 0,83 | 0,27 |
| L3_P5 | 0,32 | 0,18 | 0,03 | 0,17 | 0,27 | 0,17 | 0,28 | 0,18 | 0,26 | 0,17 | 0,09 | 0,17 | 0,39 | 0,16 |
| L3_P6 | 0,66 | 0,21 | 0,51 | 0,19 | 0,63 | 0,20 | 0,59 | 0,20 | 0,59 | 0,20 | 0,67 | 0,22 | 0,62 | 0,19 |
| L3_P1 | 0,85 | 0,37 | 0,91 | 0,47 | 0,93 | 0,50 | 0,94 | 0,52 | 0,93 | 0,50 | 0,94 | 0,52 | 0,87 | 0,35 |
| L3_P2 | 0,88 | 0,36 | 0,94 | 0,51 | 0,94 | 0,56 | 0,96 | 0,59 | 0,95 | 0,56 | 0,97 | 0,62 | 0,89 | 0,35 |
| **MOY** | **0,81** | **0,36** | **0,82** | **0,45** | **0,86** | **0,50** | **0,86** | **0,52** | **0,86** | **0,50** | **0,87** | **0,53** | **0,81** | **0,32** |
| +/- | 0,01 | -0,04 | 0,02 | 0,06 | 0,06 | 0,10 | 0,06 | 0,13 | 0,05 | 0,11 | 0,07 | 0,14 | 0,01 | -0,07 |

TABLE 3.28 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L4 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,92 | 0,44 | 0,99 | 0,75 | 0,92 | 0,44 | 0,93 | 0,46 | 0,87 | 0,33 | 0,96 | 0,66 | 0,99 | 0,74 |
| L4_P5 | 0,11 | 0,17 | 0,69 | 0,24 | 0,12 | 0,17 | 0,13 | 0,17 | 0,67 | 0,22 | 0,61 | 0,22 | 0,58 | 0,21 |
| L4_P6 | 0,52 | 0,19 | 0,92 | 0,38 | 0,53 | 0,19 | 0,53 | 0,19 | 0,57 | 0,18 | 0,66 | 0,21 | 0,89 | 0,36 |
| L4_P1 | 0,88 | 0,44 | 0,98 | 0,69 | 0,88 | 0,45 | 0,89 | 0,45 | 0,82 | 0,30 | 0,92 | 0,55 | 0,97 | 0,69 |
| L4_P2 | 0,88 | 0,37 | 0,98 | 0,73 | 0,89 | 0,38 | 0,89 | 0,38 | 0,79 | 0,23 | 0,94 | 0,53 | 0,98 | 0,73 |
| L4_P3 | 0,83 | 0,34 | 0,98 | 0,73 | 0,84 | 0,35 | 0,85 | 0,36 | 0,76 | 0,25 | 0,94 | 0,58 | 0,98 | 0,72 |
| **MOY** | **0,79** | **0,36** | **0,95** | **0,65** | **0,79** | **0,36** | **0,80** | **0,37** | **0,77** | **0,26** | **0,89** | **0,51** | **0,94** | **0,64** |
| +/- | 0,00 | 0,00 | 0,17 | 0,30 | 0,00 | 0,01 | 0,01 | 0,01 | -0,01 | -0,10 | 0,10 | 0,15 | 0,16 | 0,29 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,95 | 0,53 | 0,98 | 0,70 | 0,99 | 0,77 | 0,99 | 0,83 | 0,99 | 0,79 | 0,99 | 0,80 | 0,96 | 0,49 |
| L4_P5 | 0,47 | 0,19 | 0,12 | 0,17 | 0,39 | 0,18 | 0,40 | 0,19 | 0,38 | 0,18 | 0,19 | 0,17 | 0,54 | 0,18 |
| L4_P6 | 0,79 | 0,26 | 0,64 | 0,21 | 0,70 | 0,22 | 0,68 | 0,22 | 0,69 | 0,22 | 0,70 | 0,21 | 0,76 | 0,22 |
| L4_P1 | 0,92 | 0,50 | 0,96 | 0,62 | 0,96 | 0,64 | 0,97 | 0,67 | 0,96 | 0,65 | 0,97 | 0,69 | 0,93 | 0,45 |
| L4_P2 | 0,93 | 0,45 | 0,96 | 0,60 | 0,94 | 0,63 | 0,97 | 0,67 | 0,95 | 0,64 | 0,98 | 0,72 | 0,94 | 0,43 |
| L4_P3 | 0,92 | 0,47 | 0,93 | 0,53 | 0,91 | 0,58 | 0,95 | 0,65 | 0,91 | 0,57 | 0,97 | 0,69 | 0,91 | 0,39 |
| **MOY** | **0,88** | **0,44** | **0,86** | **0,53** | **0,88** | **0,57** | **0,90** | **0,61** | **0,88** | **0,57** | **0,89** | **0,63** | **0,89** | **0,40** |
| +/- | 0,10 | 0,08 | 0,08 | 0,18 | 0,10 | 0,21 | 0,11 | 0,25 | 0,10 | 0,22 | 0,11 | 0,27 | 0,10 | 0,04 |

TABLE 3.29 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L5 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L5_P5 | 0,97 | 0,59 | 0,97 | 0,57 | 0,96 | 0,58 | 0,94 | 0,55 | 0,98 | 0,63 | 0,94 | 0,52 | 0,96 | 0,55 |
| L5_P6 | 0,16 | 0,15 | 0,95 | 0,44 | 0,17 | 0,15 | 0,17 | 0,15 | 0,51 | 0,16 | 0,82 | 0,28 | 0,92 | 0,36 |
| L5_P1 | 0,13 | 0,18 | 0,95 | 0,54 | 0,13 | 0,18 | 0,11 | 0,17 | 0,59 | 0,16 | 0,87 | 0,39 | 0,94 | 0,49 |
| L5_P2 | 0,13 | 0,14 | 0,97 | 0,57 | 0,13 | 0,14 | 0,11 | 0,14 | 0,62 | 0,15 | 0,88 | 0,34 | 0,96 | 0,51 |
| L5_P3 | 0,10 | 0,16 | 0,96 | 0,58 | 0,11 | 0,16 | 0,10 | 0,16 | 0,61 | 0,16 | 0,87 | 0,35 | 0,95 | 0,51 |
| L5_P4 | 0,14 | 0,14 | 0,97 | 0,53 | 0,14 | 0,14 | 0,12 | 0,14 | 0,62 | 0,16 | 0,89 | 0,35 | 0,95 | 0,47 |
| **MOY** | **0,18** | **0,18** | **0,96** | **0,54** | **0,19** | **0,18** | **0,17** | **0,18** | **0,62** | **0,19** | **0,88** | **0,36** | **0,95** | **0,49** |
| +/- | 0,00 | 0,00 | 0,78 | 0,36 | 0,00 | 0,00 | -0,01 | 0,00 | 0,44 | 0,01 | 0,69 | 0,18 | 0,76 | 0,30 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L5_P5 | 0,95 | 0,52 | 0,96 | 0,55 | 0,97 | 0,62 | 0,98 | 0,69 | 0,98 | 0,65 | 0,98 | 0,65 | 0,95 | 0,45 |
| L5_P6 | 0,67 | 0,19 | 0,20 | 0,15 | 0,41 | 0,16 | 0,11 | 0,14 | 0,19 | 0,14 | 0,16 | 0,15 | 0,75 | 0,22 |
| L5_P1 | 0,65 | 0,23 | 0,17 | 0,16 | 0,39 | 0,14 | 0,38 | 0,15 | 0,37 | 0,14 | 0,25 | 0,16 | 0,81 | 0,28 |
| L5_P2 | 0,66 | 0,20 | 0,21 | 0,13 | 0,44 | 0,13 | 0,43 | 0,13 | 0,42 | 0,13 | 0,28 | 0,13 | 0,82 | 0,26 |
| L5_P3 | 0,67 | 0,21 | 0,20 | 0,15 | 0,43 | 0,14 | 0,43 | 0,15 | 0,41 | 0,14 | 0,28 | 0,15 | 0,80 | 0,26 |
| L5_P4 | 0,66 | 0,20 | 0,23 | 0,14 | 0,47 | 0,14 | 0,47 | 0,15 | 0,45 | 0,14 | 0,32 | 0,15 | 0,82 | 0,26 |
| **MOY** | **0,68** | **0,23** | **0,25** | **0,17** | **0,46** | **0,17** | **0,43** | **0,18** | **0,42** | **0,17** | **0,31** | **0,18** | **0,81** | **0,27** |
| +/- | 0,50 | 0,04 | 0,07 | -0,01 | 0,28 | -0,01 | 0,24 | -0,01 | 0,24 | -0,01 | 0,13 | 0,00 | 0,63 | 0,09 |

TABLE 3.30 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L6 (base "BUREAU") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L6_P6 | 0,98 | 0,79 | 0,98 | 0,71 | 0,98 | 0,78 | 0,98 | 0,77 | 0,96 | 0,65 | 0,98 | 0,76 | 0,98 | 0,76 |
| L6_P1 | 0,76 | 0,31 | 0,97 | 0,67 | 0,76 | 0,31 | 0,75 | 0,31 | 0,59 | 0,23 | 0,77 | 0,31 | 0,95 | 0,57 |
| L6_P2 | 0,75 | 0,25 | 0,98 | 0,65 | 0,75 | 0,25 | 0,75 | 0,25 | 0,60 | 0,18 | 0,76 | 0,25 | 0,96 | 0,53 |
| L6_P3 | 0,75 | 0,28 | 0,97 | 0,67 | 0,75 | 0,28 | 0,75 | 0,28 | 0,56 | 0,20 | 0,77 | 0,28 | 0,96 | 0,58 |
| L6_P4 | 0,77 | 0,26 | 0,97 | 0,62 | 0,76 | 0,26 | 0,76 | 0,26 | 0,62 | 0,19 | 0,79 | 0,26 | 0,96 | 0,53 |
| L6_P5 | 0,16 | 0,18 | 0,74 | 0,26 | 0,16 | 0,18 | 0,15 | 0,17 | 0,55 | 0,21 | 0,64 | 0,23 | 0,64 | 0,23 |
| **MOY** | **0,74** | **0,33** | **0,96** | **0,63** | **0,74** | **0,33** | **0,74** | **0,33** | **0,63** | **0,25** | **0,79** | **0,33** | **0,94** | **0,56** |
| +/- | 0,00 | 0,00 | 0,22 | 0,31 | 0,00 | 0,00 | 0,00 | 0,00 | -0,11 | -0,07 | 0,05 | 0,00 | 0,20 | 0,23 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L6_P6 | 0,92 | 0,53 | 0,98 | 0,78 | 0,98 | 0,77 | 0,99 | 0,84 | 0,99 | 0,84 | 0,99 | 0,78 | 0,95 | 0,50 |
| L6_P1 | 0,71 | 0,28 | 0,70 | 0,28 | 0,74 | 0,29 | 0,73 | 0,30 | 0,74 | 0,30 | 0,87 | 0,41 | 0,78 | 0,28 |
| L6_P2 | 0,70 | 0,23 | 0,72 | 0,24 | 0,77 | 0,25 | 0,77 | 0,26 | 0,76 | 0,25 | 0,94 | 0,46 | 0,75 | 0,22 |
| L6_P3 | 0,72 | 0,26 | 0,72 | 0,27 | 0,76 | 0,28 | 0,75 | 0,28 | 0,73 | 0,27 | 0,95 | 0,54 | 0,74 | 0,25 |
| L6_P4 | 0,70 | 0,23 | 0,73 | 0,25 | 0,80 | 0,27 | 0,79 | 0,27 | 0,79 | 0,27 | 0,94 | 0,47 | 0,79 | 0,24 |
| L6_P5 | 0,47 | 0,19 | 0,14 | 0,17 | 0,39 | 0,19 | 0,29 | 0,19 | 0,30 | 0,18 | 0,17 | 0,18 | 0,45 | 0,18 |
| **MOY** | **0,72** | **0,28** | **0,71** | **0,31** | **0,76** | **0,32** | **0,75** | **0,34** | **0,75** | **0,33** | **0,88** | **0,48** | **0,76** | **0,27** |
| +/- | -0,02 | -0,05 | -0,04 | -0,01 | 0,02 | 0,00 | 0,01 | 0,01 | 0,01 | 0,00 | 0,14 | 0,16 | 0,02 | -0,06 |

# 5 Codebook RGB-D : résultats détaillés L1 à L4 pour la base "JARDIN"

TABLE 3.31 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L1 (base "JARDIN") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,94 | 0,29 | 0,95 | 0,32 | 0,94 | 0,30 | 0,94 | 0,30 | 0,87 | 0,20 | 0,94 | 0,29 | 0,95 | 0,32 |
| L1_P2 | 0,84 | 0,13 | 0,85 | 0,14 | 0,78 | 0,12 | 0,79 | 0,12 | 0,53 | 0,10 | 0,82 | 0,12 | 0,70 | 0,11 |
| L1_P3 | 0,88 | 0,17 | 0,89 | 0,19 | 0,85 | 0,16 | 0,85 | 0,16 | 0,62 | 0,11 | 0,86 | 0,16 | 0,82 | 0,15 |
| L1_P4 | 0,90 | 0,06 | 0,90 | 0,07 | 0,86 | 0,05 | 0,86 | 0,05 | 0,64 | 0,03 | 0,87 | 0,05 | 0,88 | 0,06 |
| **MOY** | **0,89** | **0,13** | **0,90** | **0,14** | **0,86** | **0,12** | **0,86** | **0,12** | **0,65** | **0,08** | **0,87** | **0,12** | **0,85** | **0,13** |
| +/- | 0,00 | 0,00 | 0,00 | 0,01 | -0,03 | -0,01 | -0,03 | -0,01 | -0,24 | -0,05 | -0,02 | -0,01 | -0,04 | -0,01 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,91 | 0,24 | 0,94 | 0,29 | 0,94 | 0,31 | 0,94 | 0,31 | 0,94 | 0,31 | 0,95 | 0,32 | 0,89 | 0,21 |
| L1_P2 | 0,74 | 0,11 | 0,28 | 0,09 | 0,70 | 0,11 | 0,70 | 0,11 | 0,68 | 0,11 | 0,69 | 0,11 | 0,39 | 0,09 |
| L1_P3 | 0,79 | 0,14 | 0,32 | 0,09 | 0,83 | 0,17 | 0,84 | 0,17 | 0,83 | 0,16 | 0,85 | 0,17 | 0,37 | 0,08 |
| L1_P4 | 0,81 | 0,04 | 0,49 | 0,02 | 0,85 | 0,05 | 0,85 | 0,05 | 0,85 | 0,05 | 0,86 | 0,06 | 0,44 | 0,02 |
| **MOY** | **0,81** | **0,10** | **0,49** | **0,09** | **0,84** | **0,12** | **0,84** | **0,12** | **0,83** | **0,12** | **0,85** | **0,13** | **0,49** | **0,08** |
| +/- | -0,08 | -0,03 | -0,40 | -0,04 | -0,06 | -0,01 | -0,05 | -0,01 | -0,06 | -0,01 | -0,05 | 0,00 | -0,40 | -0,05 |

TABLE 3.32 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L2 (base "JARDIN") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,94 | 0,26 | 0,94 | 0,26 | 0,94 | 0,26 | 0,94 | 0,26 | 0,93 | 0,23 | 0,94 | 0,26 | 0,94 | 0,26 |
| L2_P3 | 0,94 | 0,24 | 0,94 | 0,26 | 0,91 | 0,20 | 0,91 | 0,20 | 0,87 | 0,17 | 0,93 | 0,22 | 0,94 | 0,24 |
| L2_P4 | 0,94 | 0,09 | 0,94 | 0,10 | 0,90 | 0,07 | 0,91 | 0,07 | 0,85 | 0,05 | 0,92 | 0,08 | 0,93 | 0,08 |
| L2_P1 | 0,87 | 0,19 | 0,90 | 0,23 | 0,83 | 0,17 | 0,82 | 0,17 | 0,70 | 0,14 | 0,85 | 0,18 | 0,84 | 0,19 |
| **MOY** | **0,93** | **0,16** | **0,93** | **0,18** | **0,90** | **0,14** | **0,90** | **0,14** | **0,85** | **0,12** | **0,92** | **0,15** | **0,92** | **0,16** |
| +/- | 0,00 | 0,00 | 0,01 | 0,01 | -0,03 | -0,02 | -0,03 | -0,02 | -0,08 | -0,05 | -0,01 | -0,01 | -0,01 | 0,00 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,93 | 0,21 | 0,92 | 0,21 | 0,94 | 0,25 | 0,94 | 0,25 | 0,94 | 0,25 | 0,94 | 0,25 | 0,84 | 0,14 |
| L2_P3 | 0,86 | 0,16 | 0,87 | 0,17 | 0,90 | 0,19 | 0,91 | 0,19 | 0,90 | 0,19 | 0,90 | 0,19 | 0,75 | 0,12 |
| L2_P4 | 0,82 | 0,05 | 0,86 | 0,05 | 0,91 | 0,07 | 0,90 | 0,07 | 0,90 | 0,07 | 0,92 | 0,08 | 0,74 | 0,03 |
| L2_P1 | 0,72 | 0,14 | 0,49 | 0,12 | 0,80 | 0,17 | 0,80 | 0,17 | 0,81 | 0,17 | 0,79 | 0,17 | 0,60 | 0,12 |
| **MOY** | **0,83** | **0,11** | **0,81** | **0,11** | **0,89** | **0,14** | **0,89** | **0,14** | **0,89** | **0,14** | **0,90** | **0,14** | **0,74** | **0,08** |
| +/- | -0,10 | -0,05 | -0,12 | -0,05 | -0,04 | -0,02 | -0,04 | -0,02 | -0,04 | -0,02 | -0,03 | -0,02 | -0,19 | -0,08 |

TABLE 3.33 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L3 (base "JARDIN") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,94 | 0,23 | 0,93 | 0,23 | 0,94 | 0,23 | 0,94 | 0,23 | 0,84 | 0,15 | 0,94 | 0,23 | 0,92 | 0,21 |
| L3_P4 | 0,94 | 0,09 | 0,94 | 0,09 | 0,93 | 0,08 | 0,93 | 0,08 | 0,83 | 0,05 | 0,93 | 0,08 | 0,93 | 0,08 |
| L3_P1 | 0,87 | 0,19 | 0,89 | 0,22 | 0,86 | 0,19 | 0,86 | 0,19 | 0,70 | 0,14 | 0,85 | 0,18 | 0,85 | 0,19 |
| L3_P2 | 0,93 | 0,21 | 0,93 | 0,21 | 0,91 | 0,18 | 0,91 | 0,18 | 0,87 | 0,15 | 0,92 | 0,19 | 0,89 | 0,17 |
| **MOY** | **0,93** | **0,15** | **0,93** | **0,16** | **0,92** | **0,14** | **0,92** | **0,14** | **0,82** | **0,10** | **0,92** | **0,14** | **0,91** | **0,14** |
| +/- | 0,00 | 0,00 | 0,00 | 0,01 | -0,01 | -0,01 | -0,01 | -0,01 | -0,11 | -0,05 | -0,01 | -0,01 | -0,02 | -0,01 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,89 | 0,18 | 0,48 | 0,09 | 0,91 | 0,19 | 0,91 | 0,19 | 0,90 | 0,18 | 0,92 | 0,21 | 0,46 | 0,09 |
| L3_P4 | 0,86 | 0,05 | 0,69 | 0,03 | 0,91 | 0,07 | 0,91 | 0,07 | 0,91 | 0,07 | 0,92 | 0,08 | 0,54 | 0,02 |
| L3_P1 | 0,75 | 0,14 | 0,62 | 0,13 | 0,80 | 0,17 | 0,80 | 0,17 | 0,81 | 0,17 | 0,80 | 0,17 | 0,70 | 0,14 |
| L3_P2 | 0,87 | 0,15 | 0,66 | 0,11 | 0,92 | 0,19 | 0,92 | 0,19 | 0,92 | 0,19 | 0,92 | 0,19 | 0,50 | 0,10 |
| **MOY** | **0,85** | **0,11** | **0,63** | **0,07** | **0,89** | **0,13** | **0,89** | **0,13** | **0,89** | **0,13** | **0,90** | **0,14** | **0,54** | **0,07** |
| +/- | -0,08 | -0,04 | -0,29 | -0,08 | -0,03 | -0,02 | -0,03 | -0,02 | -0,04 | -0,02 | -0,03 | -0,01 | -0,38 | -0,08 |

TABLE 3.34 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L4 (base "JARDIN") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,97 | 0,13 | 0,96 | 0,13 | 0,96 | 0,13 | 0,96 | 0,13 | 0,85 | 0,05 | 0,96 | 0,12 | 0,96 | 0,12 |
| L4_P1 | 0,88 | 0,19 | 0,88 | 0,21 | 0,86 | 0,19 | 0,86 | 0,19 | 0,68 | 0,13 | 0,85 | 0,18 | 0,87 | 0,20 |
| L4_P2 | 0,90 | 0,16 | 0,90 | 0,18 | 0,87 | 0,15 | 0,87 | 0,15 | 0,80 | 0,12 | 0,89 | 0,16 | 0,82 | 0,13 |
| L4_P3 | 0,92 | 0,22 | 0,93 | 0,23 | 0,91 | 0,20 | 0,91 | 0,20 | 0,79 | 0,13 | 0,91 | 0,20 | 0,90 | 0,20 |
| **MOY** | **0,93** | **0,16** | **0,93** | **0,17** | **0,92** | **0,15** | **0,92** | **0,15** | **0,80** | **0,09** | **0,92** | **0,15** | **0,91** | **0,15** |
| +/- | 0,00 | 0,00 | 0,00 | 0,01 | -0,01 | -0,01 | -0,01 | -0,01 | -0,13 | -0,07 | -0,01 | -0,01 | -0,02 | -0,01 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,93 | 0,08 | 0,81 | 0,04 | 0,92 | 0,08 | 0,91 | 0,07 | 0,91 | 0,07 | 0,95 | 0,11 | 0,67 | 0,03 |
| L4_P1 | 0,76 | 0,14 | 0,60 | 0,13 | 0,79 | 0,16 | 0,79 | 0,17 | 0,80 | 0,17 | 0,81 | 0,17 | 0,70 | 0,14 |
| L4_P2 | 0,84 | 0,13 | 0,69 | 0,11 | 0,87 | 0,14 | 0,87 | 0,15 | 0,86 | 0,14 | 0,86 | 0,15 | 0,53 | 0,10 |
| L4_P3 | 0,85 | 0,15 | 0,70 | 0,12 | 0,89 | 0,18 | 0,90 | 0,19 | 0,89 | 0,18 | 0,91 | 0,20 | 0,51 | 0,09 |
| **MOY** | **0,87** | **0,11** | **0,74** | **0,08** | **0,89** | **0,12** | **0,88** | **0,12** | **0,88** | **0,12** | **0,91** | **0,15** | **0,62** | **0,07** |
| +/- | -0,06 | -0,05 | -0,20 | -0,08 | -0,05 | -0,04 | -0,05 | -0,04 | -0,05 | -0,04 | -0,03 | -0,02 | -0,31 | -0,09 |

# 6 Codebook RGB-D : résultats détaillés L1 à L5 pour la base "RUE"

TABLE 3.35 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L1 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L1_P1 | 0,95 | 0,03 | 0,95 | 0,02 | 0,95 | 0,03 | 0,95 | 0,03 | 0,94 | 0,06 | 0,95 | 0,04 | 0,95 | 0,02 |
| L1_P2 | 0,96 | 0,12 | 0,97 | 0,09 | 0,96 | 0,12 | 0,96 | 0,13 | 0,87 | 0,08 | 0,96 | 0,15 | 0,93 | 0,06 |
| L1_P3 | 0,86 | 0,08 | 0,95 | 0,13 | 0,87 | 0,10 | 0,88 | 0,10 | 0,72 | 0,09 | 0,94 | 0,16 | 0,94 | 0,11 |
| L1_P4 | 0,95 | 0,14 | 0,96 | 0,08 | 0,94 | 0,13 | 0,94 | 0,13 | 0,83 | 0,07 | 0,94 | 0,12 | 0,96 | 0,09 |
| L1_P5 | 0,92 | 0,13 | 0,95 | 0,09 | 0,93 | 0,15 | 0,93 | 0,14 | 0,81 | 0,07 | 0,94 | 0,14 | 0,95 | 0,10 |
| **MOY** | **0,92** | **0,10** | **0,96** | **0,08** | **0,93** | **0,11** | **0,93** | **0,11** | **0,83** | **0,08** | **0,95** | **0,12** | **0,95** | **0,08** |
| +/- | 0,00 | 0,00 | 0,03 | -0,02 | 0,01 | 0,01 | 0,00 | 0,01 | -0,09 | -0,03 | 0,02 | 0,02 | 0,02 | -0,02 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L1_P1 | 0,91 | 0,03 | 0,95 | 0,02 | 0,95 | 0,02 | 0,95 | 0,02 | 0,95 | 0,02 | 0,95 | 0,02 | 0,91 | 0,03 |
| L1_P2 | 0,81 | 0,05 | 0,96 | 0,10 | 0,96 | 0,08 | 0,96 | 0,09 | 0,96 | 0,09 | 0,97 | 0,09 | 0,66 | 0,04 |
| L1_P3 | 0,76 | 0,08 | 0,89 | 0,10 | 0,88 | 0,08 | 0,90 | 0,10 | 0,88 | 0,09 | 0,95 | 0,13 | 0,61 | 0,07 |
| L1_P4 | 0,76 | 0,04 | 0,95 | 0,09 | 0,96 | 0,08 | 0,96 | 0,08 | 0,96 | 0,08 | 0,96 | 0,09 | 0,66 | 0,03 |
| L1_P5 | 0,75 | 0,05 | 0,93 | 0,09 | 0,95 | 0,09 | 0,95 | 0,09 | 0,95 | 0,09 | 0,96 | 0,09 | 0,65 | 0,04 |
| **MOY** | **0,80** | **0,05** | **0,93** | **0,08** | **0,94** | **0,07** | **0,94** | **0,08** | **0,94** | **0,07** | **0,96** | **0,09** | **0,70** | **0,04** |
| +/- | -0,13 | -0,05 | 0,01 | -0,02 | 0,01 | -0,03 | 0,02 | -0,02 | 0,02 | -0,03 | 0,03 | -0,01 | -0,23 | -0,06 |

TABLE 3.36 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L2 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L2_P2 | 0,99 | 0,41 | 0,99 | 0,39 | 0,99 | 0,41 | 0,99 | 0,42 | 0,99 | 0,43 | 0,99 | 0,46 | 0,99 | 0,40 |
| L2_P3 | 0,83 | 0,12 | 0,95 | 0,24 | 0,86 | 0,14 | 0,86 | 0,14 | 0,71 | 0,10 | 0,95 | 0,26 | 0,95 | 0,24 |
| L2_P4 | 0,92 | 0,12 | 0,96 | 0,13 | 0,93 | 0,13 | 0,93 | 0,13 | 0,83 | 0,07 | 0,95 | 0,13 | 0,96 | 0,13 |
| L2_P5 | 0,91 | 0,15 | 0,94 | 0,17 | 0,92 | 0,16 | 0,92 | 0,16 | 0,81 | 0,08 | 0,93 | 0,17 | 0,94 | 0,17 |
| L2_P1 | 0,78 | 0,02 | 0,82 | 0,02 | 0,79 | 0,02 | 0,78 | 0,02 | 0,70 | 0,02 | 0,79 | 0,02 | 0,81 | 0,02 |
| **MOY** | **0,89** | **0,16** | **0,93** | **0,18** | **0,89** | **0,16** | **0,89** | **0,17** | **0,80** | **0,13** | **0,92** | **0,20** | **0,93** | **0,18** |
| +/- | 0,00 | 0,00 | 0,04 | 0,02 | 0,01 | 0,01 | 0,01 | 0,01 | -0,08 | -0,03 | 0,03 | 0,04 | 0,04 | 0,03 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L2_P2 | 0,99 | 0,39 | 0,99 | 0,40 | 0,99 | 0,40 | 0,99 | 0,40 | 0,99 | 0,40 | 0,99 | 0,40 | 0,99 | 0,45 |
| L2_P3 | 0,87 | 0,15 | 0,92 | 0,19 | 0,89 | 0,15 | 0,90 | 0,16 | 0,88 | 0,14 | 0,95 | 0,24 | 0,29 | 0,07 |
| L2_P4 | 0,87 | 0,07 | 0,94 | 0,12 | 0,95 | 0,13 | 0,95 | 0,13 | 0,95 | 0,13 | 0,96 | 0,14 | 0,32 | 0,03 |
| L2_P5 | 0,84 | 0,09 | 0,90 | 0,13 | 0,94 | 0,17 | 0,94 | 0,17 | 0,94 | 0,17 | 0,95 | 0,17 | 0,25 | 0,03 |
| L2_P1 | 0,64 | 0,01 | 0,72 | 0,02 | 0,81 | 0,02 | 0,81 | 0,02 | 0,81 | 0,02 | 0,82 | 0,02 | 0,31 | 0,01 |
| **MOY** | **0,84** | **0,14** | **0,89** | **0,17** | **0,91** | **0,17** | **0,92** | **0,17** | **0,91** | **0,17** | **0,93** | **0,18** | **0,42** | **0,11** |
| +/- | -0,05 | -0,02 | 0,00 | 0,01 | 0,03 | 0,01 | 0,03 | 0,01 | 0,03 | 0,01 | 0,04 | 0,03 | -0,46 | -0,05 |

TABLE 3.37 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L3 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L3_P3 | 0,99 | 0,50 | 0,98 | 0,47 | 0,99 | 0,51 | 0,99 | 0,51 | 0,97 | 0,52 | 0,99 | 0,53 | 0,98 | 0,48 |
| L3_P4 | 0,95 | 0,15 | 0,96 | 0,13 | 0,95 | 0,15 | 0,95 | 0,15 | 0,85 | 0,09 | 0,95 | 0,15 | 0,96 | 0,13 |
| L3_P5 | 0,89 | 0,13 | 0,93 | 0,14 | 0,91 | 0,15 | 0,91 | 0,15 | 0,79 | 0,08 | 0,92 | 0,16 | 0,93 | 0,14 |
| L3_P1 | 0,79 | 0,02 | 0,80 | 0,02 | 0,79 | 0,02 | 0,79 | 0,02 | 0,68 | 0,02 | 0,78 | 0,02 | 0,80 | 0,02 |
| L3_P2 | 0,95 | 0,17 | 0,95 | 0,14 | 0,95 | 0,17 | 0,95 | 0,17 | 0,87 | 0,09 | 0,94 | 0,15 | 0,95 | 0,14 |
| **MOY** | **0,91** | **0,19** | **0,92** | **0,17** | **0,91** | **0,19** | **0,91** | **0,19** | **0,83** | **0,15** | **0,91** | **0,19** | **0,92** | **0,18** |
| +/- | 0,00 | 0,00 | 0,01 | -0,01 | 0,01 | 0,01 | 0,00 | 0,00 | -0,08 | -0,03 | 0,00 | 0,01 | 0,01 | -0,01 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L3_P3 | 0,98 | 0,48 | 0,99 | 0,54 | 0,99 | 0,52 | 0,99 | 0,52 | 0,99 | 0,52 | 0,98 | 0,48 | 0,98 | 0,43 |
| L3_P4 | 0,85 | 0,08 | 0,93 | 0,12 | 0,94 | 0,12 | 0,95 | 0,14 | 0,95 | 0,14 | 0,96 | 0,14 | 0,88 | 0,08 |
| L3_P5 | 0,75 | 0,07 | 0,84 | 0,09 | 0,93 | 0,14 | 0,92 | 0,14 | 0,91 | 0,14 | 0,93 | 0,15 | 0,79 | 0,07 |
| L3_P1 | 0,58 | 0,01 | 0,65 | 0,01 | 0,79 | 0,02 | 0,79 | 0,02 | 0,80 | 0,02 | 0,80 | 0,02 | 0,51 | 0,01 |
| L3_P2 | 0,84 | 0,07 | 0,94 | 0,13 | 0,95 | 0,15 | 0,95 | 0,16 | 0,95 | 0,16 | 0,95 | 0,14 | 0,51 | 0,03 |
| **MOY** | **0,79** | **0,14** | **0,86** | **0,17** | **0,92** | **0,18** | **0,92** | **0,19** | **0,92** | **0,19** | **0,92** | **0,18** | **0,73** | **0,12** |
| +/- | -0,12 | -0,05 | -0,04 | -0,02 | 0,01 | 0,00 | 0,01 | 0,00 | 0,01 | 0,00 | 0,01 | -0,01 | -0,18 | -0,07 |

TABLE 3.38 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L4 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L4_P4 | 0,99 | 0,38 | 0,99 | 0,34 | 0,99 | 0,39 | 0,99 | 0,39 | 0,98 | 0,40 | 0,99 | 0,40 | 0,99 | 0,35 |
| L4_P5 | 0,95 | 0,19 | 0,95 | 0,16 | 0,95 | 0,18 | 0,95 | 0,18 | 0,89 | 0,11 | 0,95 | 0,18 | 0,95 | 0,16 |
| L4_P1 | 0,74 | 0,04 | 0,81 | 0,05 | 0,75 | 0,04 | 0,74 | 0,04 | 0,68 | 0,03 | 0,79 | 0,05 | 0,81 | 0,05 |
| L4_P2 | 0,84 | 0,08 | 0,96 | 0,17 | 0,85 | 0,08 | 0,85 | 0,08 | 0,82 | 0,08 | 0,95 | 0,17 | 0,96 | 0,17 |
| L4_P3 | 0,58 | 0,06 | 0,95 | 0,18 | 0,65 | 0,07 | 0,63 | 0,07 | 0,69 | 0,09 | 0,94 | 0,18 | 0,95 | 0,18 |
| **MOY** | **0,83** | **0,15** | **0,93** | **0,18** | **0,84** | **0,15** | **0,84** | **0,16** | **0,82** | **0,15** | **0,92** | **0,20** | **0,93** | **0,18** |
| +/- | 0,00 | 0,00 | 0,10 | 0,03 | 0,01 | 0,00 | 0,01 | 0,00 | -0,01 | -0,01 | 0,09 | 0,04 | 0,10 | 0,03 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L4_P4 | 0,98 | 0,31 | 0,98 | 0,35 | 0,99 | 0,35 | 0,99 | 0,35 | 0,99 | 0,35 | 0,99 | 0,35 | 0,97 | 0,25 |
| L4_P5 | 0,89 | 0,10 | 0,93 | 0,13 | 0,95 | 0,16 | 0,95 | 0,16 | 0,94 | 0,16 | 0,95 | 0,16 | 0,86 | 0,08 |
| L4_P1 | 0,65 | 0,04 | 0,74 | 0,05 | 0,81 | 0,05 | 0,80 | 0,05 | 0,81 | 0,05 | 0,81 | 0,05 | 0,57 | 0,02 |
| L4_P2 | 0,89 | 0,10 | 0,94 | 0,14 | 0,94 | 0,14 | 0,94 | 0,14 | 0,94 | 0,14 | 0,96 | 0,17 | 0,67 | 0,06 |
| L4_P3 | 0,88 | 0,13 | 0,87 | 0,11 | 0,87 | 0,11 | 0,88 | 0,12 | 0,87 | 0,11 | 0,95 | 0,18 | 0,90 | 0,14 |
| **MOY** | **0,85** | **0,14** | **0,89** | **0,16** | **0,91** | **0,16** | **0,91** | **0,16** | **0,91** | **0,16** | **0,93** | **0,18** | **0,80** | **0,11** |
| +/- | 0,03 | -0,02 | 0,07 | 0,00 | 0,08 | 0,01 | 0,09 | 0,01 | 0,08 | 0,01 | 0,10 | 0,03 | -0,03 | -0,04 |

TABLE 3.39 – Tableau de F-Mesures obtenues pour chaque invariant colorimétrique avec un apprentissage sur L5 (base "RUE") et les paramétrages optimaux de l'algorithme Codebook RGB-D calculés sur cette base.

| | rgb | | cs | | gw | | cn | | an | | rgb-r | | c1c2c3 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg | bg | fg |
| L5_P5 | 0,99 | 0,44 | 0,99 | 0,39 | 0,99 | 0,45 | 0,99 | 0,46 | 0,99 | 0,46 | 0,99 | 0,46 | 0,99 | 0,39 |
| L5_P1 | 0,72 | 0,03 | 0,82 | 0,04 | 0,78 | 0,04 | 0,78 | 0,04 | 0,70 | 0,03 | 0,80 | 0,04 | 0,82 | 0,04 |
| L5_P2 | 0,86 | 0,07 | 0,96 | 0,15 | 0,92 | 0,11 | 0,92 | 0,11 | 0,83 | 0,07 | 0,94 | 0,15 | 0,95 | 0,14 |
| L5_P3 | 0,69 | 0,08 | 0,94 | 0,19 | 0,78 | 0,10 | 0,76 | 0,10 | 0,69 | 0,10 | 0,93 | 0,20 | 0,94 | 0,19 |
| L5_P4 | 0,89 | 0,09 | 0,96 | 0,16 | 0,95 | 0,17 | 0,94 | 0,17 | 0,88 | 0,10 | 0,96 | 0,17 | 0,96 | 0,16 |
| **MOY** | **0,83** | **0,14** | **0,93** | **0,18** | **0,88** | **0,17** | **0,88** | **0,17** | **0,82** | **0,15** | **0,92** | **0,20** | **0,93** | **0,18** |
| +/- | 0,00 | 0,00 | 0,10 | 0,04 | 0,05 | 0,03 | 0,05 | 0,03 | -0,01 | 0,01 | 0,09 | 0,06 | 0,10 | 0,04 |
| | | | | | | | | | | | | | | |
| | m1m2m3 | | hsl | | yiq | | ycbcr | | ych1ch2 | | lab | | l1l2l3 | |
| L5_P5 | 0,98 | 0,35 | 0,98 | 0,38 | 0,99 | 0,38 | 0,99 | 0,39 | 0,99 | 0,39 | 0,99 | 0,39 | 0,97 | 0,30 |
| L5_P1 | 0,67 | 0,03 | 0,79 | 0,04 | 0,82 | 0,03 | 0,82 | 0,04 | 0,82 | 0,04 | 0,82 | 0,04 | 0,55 | 0,02 |
| L5_P2 | 0,88 | 0,08 | 0,94 | 0,13 | 0,95 | 0,13 | 0,95 | 0,14 | 0,95 | 0,14 | 0,96 | 0,15 | 0,46 | 0,03 |
| L5_P3 | 0,82 | 0,12 | 0,85 | 0,12 | 0,89 | 0,14 | 0,89 | 0,14 | 0,88 | 0,13 | 0,94 | 0,19 | 0,80 | 0,11 |
| L5_P4 | 0,89 | 0,08 | 0,95 | 0,14 | 0,96 | 0,16 | 0,96 | 0,16 | 0,96 | 0,16 | 0,96 | 0,16 | 0,83 | 0,06 |
| **MOY** | **0,85** | **0,13** | **0,90** | **0,16** | **0,92** | **0,17** | **0,92** | **0,17** | **0,92** | **0,17** | **0,93** | **0,18** | **0,72** | **0,10** |
| +/- | 0,02 | -0,01 | 0,07 | 0,02 | 0,09 | 0,03 | 0,09 | 0,03 | 0,09 | 0,03 | 0,10 | 0,04 | -0,11 | -0,04 |

# 7 Codebook RGB-LF : résultats détaillés L1 à L6 pour la base “BUREAU”

TABLE 3.40 – Tableau de F-Mesures obtenues avec un apprentissage sur L1 (base “BUREAU”) et l’algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L1_P2 | 0,96 | 0,01 |
| L1_P1 | 0,89 | 0,56 |
| L1_P3 | 0,96 | 0,54 |
| L1_P4 | 0,98 | 0,60 |
| L1_P5 | 0,64 | 0,17 |
| L1_P6 | 0,80 | 0,26 |
| **MOY** | **0,91** | **0,49** |

TABLE 3.41 – Tableau de F-Mesures obtenues avec un apprentissage sur L2 (base “BUREAU”) et l’algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L2_P2 | 0,86 | 0,16 |
| L2_P3 | 0,93 | 0,33 |
| L2_P4 | 0,96 | 0,41 |
| L2_P5 | 0,77 | 0,09 |
| L2_P6 | 0,79 | 0,23 |
| L2_P1 | 0,95 | 0,41 |
| **MOY** | **0,90** | **0,30** |

TABLE 3.42 – Tableau de F-Mesures obtenues avec un apprentissage sur L3 (base “BUREAU”) et l’algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L3_P3 | 0,90 | 0,55 |
| L3_P4 | 0,97 | 0,58 |
| L3_P5 | 0,51 | 0,16 |
| L3_P6 | 0,77 | 0,24 |
| L3_P1 | 0,97 | 0,59 |
| L3_P2 | 0,97 | 0,52 |
| **MOY** | **0,90** | **0,50** |

TABLE 3.43 – Tableau de F-Mesures obtenues avec un apprentissage sur L4 (base "BUREAU") et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L4_P4 | 0,96 | 0,55 |
| L4_P5 | 0,47 | 0,17 |
| L4_P6 | 0,75 | 0,24 |
| L4_P1 | 0,96 | 0,59 |
| L4_P2 | 0,96 | 0,52 |
| L4_P3 | 0,96 | 0,53 |
| **MOY** | **0,90** | **0,48** |

TABLE 3.44 – Tableau de F-Mesures obtenues avec un apprentissage sur L5 (base "BUREAU") et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L5_P5 | 0,97 | 0,23 |
| L5_P6 | 0,40 | 0,15 |
| L5_P1 | 0,39 | 0,13 |
| L5_P2 | 0,41 | 0,06 |
| L5_P3 | 0,42 | 0,08 |
| L5_P4 | 0,44 | 0,09 |
| **MOY** | **0,45** | **0,11** |

TABLE 3.45 – Tableau de F-Mesures obtenues avec un apprentissage sur L6 (base "BUREAU") et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L6_P6 | 0,99 | 0,86 |
| L6_P1 | 0,79 | 0,28 |
| L6_P2 | 0,82 | 0,22 |
| L6_P3 | 0,83 | 0,26 |
| L6_P4 | 0,85 | 0,25 |
| L6_P5 | 0,66 | 0,18 |
| **MOY** | **0,83** | **0,32** |

# 8 Codebook RGB-LF : résultats détaillés L1 à L4 pour la base “JARDIN”

TABLE 3.46 – Tableau de F-Mesures obtenues avec un apprentissage sur L1 (base “JARDIN”) et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L1_P1 | 0,95 | 0,28 |
| L1_P2 | 0,56 | 0,09 |
| L1_P3 | 0,62 | 0,08 |
| L1_P4 | 0,72 | 0,03 |
| **MOY** | **0.71** | **0.09** |

TABLE 3.47 – Tableau de F-Mesures obtenues avec un apprentissage sur L2 (base “JARDIN”) et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L2_P2 | 0,95 | 0,12 |
| L2_P3 | 0,93 | 0,11 |
| L2_P4 | 0,95 | 0,04 |
| L2_P1 | 0,66 | 0,09 |
| **MOY** | **0.90** | **0.08** |

TABLE 3.48 – Tableau de F-Mesures obtenues avec un apprentissage sur L3 (base “JARDIN”) et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L3_P3 | 0,93 | 0,12 |
| L3_P4 | 0,94 | 0,04 |
| L3_P1 | 0,70 | 0,09 |
| L3_P2 | 0,94 | 0,09 |
| **MOY** | **0.90** | **0.07** |

TABLE 3.49 – Tableau de F-Mesures obtenues avec un apprentissage sur L4 (base “JARDIN”) et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L4_P4 | 0,91 | 0,04 |
| L4_P1 | 0,68 | 0,10 |
| L4_P2 | 0,86 | 0,07 |
| L4_P3 | 0,86 | 0,09 |
| **MOY** | **0,85** | **0,06** |

# 9 Codebook RGB-LF : résultats détaillés L1 à L5 pour la base "RUE"

TABLE 3.50 – Tableau de F-Mesures obtenues avec un apprentissage sur L1 (base "RUE") et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L1_P1 | 1,00 | 0,50 |
| L1_P2 | 0,79 | 0,05 |
| L1_P3 | 0,13 | 0,03 |
| L1_P4 | 0,89 | 0,05 |
| L1_P5 | 0,90 | 0,05 |
| **MOY** | **0,70** | **0,13** |

TABLE 3.51 – Tableau de F-Mesures obtenues avec un apprentissage sur L2 (base "RUE") et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L2_P2 | 0,99 | 0,25 |
| L2_P3 | 0,08 | 0,03 |
| L2_P4 | 0,94 | 0,04 |
| L2_P5 | 0,94 | 0,01 |
| L2_P1 | 0,81 | 0,01 |
| **MOY** | **0,72** | **0,08** |

TABLE 3.52 – Tableau de F-Mesures obtenues avec un apprentissage sur L3 (base "RUE") et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L3_P3 | 0,99 | 0,67 |
| L3_P4 | 0,20 | 0,02 |
| L3_P5 | 0,21 | 0,03 |
| L3_P1 | 0,25 | 0,01 |
| L3_P2 | 0,26 | 0,02 |
| **MOY** | **0,42** | **0,18** |

TABLE 3.53 – Tableau de F-Mesures obtenues avec un apprentissage sur L4 (base "RUE") et l'algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L4_P4 | 1,00 | 0,31 |
| L4_P5 | 0,92 | 0,03 |
| L4_P1 | 0,81 | 0,01 |
| L4_P2 | 0,90 | 0,04 |
| L4_P3 | 0,10 | 0,05 |
| **MOY** | **0,70** | **0,08** |

TABLE 3.54 – Tableau de F-Mesures obtenues avec un apprentissage sur L5 (base “RUE”) et l’algorithme Codebook Logique Floue (sans disparité).

| | rgb | |
|---|---|---|
| | bg | fg |
| L5_P5 | 0,96 | 0,00 |
| L5_P1 | 0,67 | 0,01 |
| L5_P2 | 0,71 | 0,04 |
| L5_P3 | 0,13 | 0,05 |
| L5_P4 | 0,82 | 0,04 |
| **MOY** | **0,62** | **0,03** |

## Résumé :

Cette thèse s'inscrit dans un cadre de vidéo-surveillance, et s'intéresse plus précisément à la détection robustes d'objets mobiles dans une séquence d'images. Une bonne détection d'objets mobiles est un prérequis indispensable à tout traitement appliqué à ces objets dans de nombreuses applications telles que le suivi de voitures ou de personnes, le comptage des passagers de transports en commun, la détection de situations dangereuses dans des environnements spécifiques (passages à niveau, passages piéton, carrefours, etc.), ou encore le contrôle de véhicules autonomes. Un très grand nombre de ces applications utilise un système de vision par ordinateur. La fiabilité de ces systèmes demande une robustesse importante face à des conditions parfois difficiles souvent causées par les conditions d'illumination (jour/nuit, ombres portées), les conditions météorologiques (pluie, vent, neige...) ainsi que la topologie même de la scène observée (occultations...). Les travaux présentés dans cette thèse visent à améliorer la qualité de détection d'objets mobiles en milieu intérieur ou extérieur, et à tout moment de la journée. Pour ce faire, nous avons proposé trois stratégies combinables : i) l'utilisation d'invariants colorimétriques et/ou d'espaces de représentation couleur présentant des propriétés invariantes ; ii) l'utilisation d'une caméra stéréoscopique et d'une caméra active Microsoft Kinect en plus de la caméra couleur afin de reconstruire l'environnement 3D partiel de la scène, et de fournir une dimension supplémentaire, à savoir une information de profondeur, à l'algorithme de détection d'objets mobiles pour la caractérisation des pixels; iii) la proposition d'un nouvel algorithme de fusion basé sur la logique floue permettant de combiner les informations de couleur et de profondeur tout en accordant une certaine marge d'incertitude quant à l'appartenance du pixel au fond ou à un objet mobile.

**Mots-clés :** segmentation d'images, invariance colorimétrique, codebook, fusion floue, segmentation fond/forme, segmentation RGB-D

## Abstract:

This PhD thesis falls within the scope of video-surveillance, and more precisely focuses on the detection of moving objects in image sequences. In many applications, good detection of moving objects is an indispensable prerequisite to any treatment applied to these objects such as people or cars tracking, passengers counting, detection of dangerous situations in specific environments (level crossings, pedestrian crossings, intersections, etc.), or control of autonomous vehicles. The reliability of computer vision based systems require robustness against difficult conditions often caused by lighting conditions (day/night, shadows), weather conditions (rain, wind, snow ...) and the topology of the observed scene (occultation ...). Works detailed in this PhD thesis aim at reducing the impact of illumination conditions by improving the quality of the detection of mobile objects in indoor or outdoor environments and at any time of the day. Thus, we propose three strategies working as a combination to improve the detection of moving objects: i) using colorimetric invariants and/or color spaces that provide invariant properties ; ii) using passive stereoscopic camera (in outdoor environments) and Microsoft Kinect active camera (in outdoor environments) in order to partially reconstruct the 3D environment, providing an additional dimension (a depth information) to the background/foreground subtraction algorithm ; iii) a new fusion algorithm based on fuzzy logic in order to combine color and depth information with a certain level of uncertainty for the pixels classification.

**Keywords:** image segmentation, color constancy, codebook, fuzzy fusion, background/foreground segmentation, RGB-D segmentation

■ École doctorale SPIM - Université de Technologie Belfort-Montbéliard
F - 90010 Belfort Cedex ■ tél. +33 (0)3 84 58 31 39
■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr