

Vers une meilleure détection des objets mobiles par Codebook et invariance couleur

J. Murgia¹, C. Meurie² et Y. Ruichek¹

¹IRTES-SeT, UTBM, 90010 Belfort Cedex, France

{julian.murgia, yassine.ruichek}@utbm.fr

²Univ Lille Nord de France, F-59000 Lille, IFSTTAR, LEOST, F59650, Villeneuve d'Ascq

cyril.meurie@ifsttar.fr

Résumé

Nous proposons dans cet article une amélioration de la détection d'objets mobiles basée sur l'algorithme du Codebook par une utilisation des invariants colorimétriques ou d'espaces couleur adaptés à l'application visée. Ce procédé a pour finalité d'améliorer la détection des objets mobiles dans des cas de scènes complexes, et ce afin de permettre un meilleur suivi des véhicules et piétons sur les passages à niveaux. Nous observons les effets de différents invariants sur des séquences acquises à plusieurs moments de la journée pour gérer les changements d'illumination et de couleurs. Notre objectif est de démontrer l'efficacité de l'usage des invariants colorimétriques avec une méthode de soustraction d'arrière-plan largement utilisée, tout en proposant un classement des meilleurs invariants.

Mots clefs

vision par ordinateur, détection, tracking, codebook, invariants colorimétriques

1 Introduction

Le concept de soustraction d'arrière-plan (en anglais "Background Subtraction" : BGS) regroupe de multiples méthodes permettant une détection efficace des objets mobiles dans une séquence d'images. Nous les retrouvons dans des applications liées aux systèmes de transports intelligents comme par exemple la détection d'obstacle pour la conduite autonome, la détection de stationnarité au passage à niveau pour l'amélioration de leur sécurité, la détection de situations potentiellement dangereuses dans les transports urbains, etc ([1, 2, 3]). L'idée de base du concept de BGS consiste à déterminer un modèle de l'arrière-plan par apprentissage. Ce modèle correspond aux régions fixes de l'images (sans objet mobile). La différence entre ce modèle d'arrière-plan et l'image observée est ensuite effectuée afin de faire ressortir les éléments mobiles.

Plusieurs méthodes de BGS ont été proposées depuis quelques années [4, 5, 6]. Les méthodes les plus simples consistent en une distribution unimodale [7, 8], mais

souffrent de mauvaises performances dans le cas d'arrière-plans dynamiques (oscillations des feuillages d'arbres par exemple) et de forts changements d'illumination de la scène, causant ainsi beaucoup de fausses détections. Une autre méthode largement utilisée est le Generalized Mixture Of Gaussians (MOG) [9] qui consiste en une distribution multi-modale de gaussiennes pour chaque pixel, permettant ainsi une modélisation du fond plus complexe. Ce modèle est capable de s'adapter plus ou moins rapidement à l'arrière-plan en fonction d'une constante de taux d'apprentissage définie en paramètre : si ce taux est trop bas, le modèle ne peut s'adapter suffisamment rapidement à des changements brusques dans la scène. A l'inverse, si cette valeur est trop élevée, alors les objets se déplaçant lentement risquent d'être inclus dans le modèle d'arrière-plan, ce qui peut causer une augmentation du pourcentage de faux négatifs.

Dans leurs travaux, Kim et al. proposent un algorithme Codebook [10] qui n'utilise pas de paramètre d'apprentissage et qui offre généralement de bons résultats. Nous avons choisi cet algorithme pour notre étude. Comme beaucoup d'autres, cet algorithme fonctionne en 2 phases : une phase d'apprentissage et une phase de mise à jour. L'apprentissage consiste à déterminer un modèle d'arrière-plan utilisé pour la comparaison des images à traiter. Ce modèle est ensuite mis à jour avec de nouvelles images pour prendre en compte les changements opérant dans la scène. Dans ce papier, nous avons volontairement ignoré l'étape de mise à jour du modèle qui s'avère être coûteuse en temps de calcul, et ce afin de nous rapprocher d'un traitement temps réel. Nous cherchons alors à détecter les objets mobiles (piétons, voitures, deux-roues) se déplaçant dans l'environnement d'un passage à niveaux en vue de pouvoir proposer in fine un système de détection de situations potentiellement dangereuses fonctionnant quel que soit le moment de la journée (ie. les variations de luminosité). Pour ce faire, nous couplons une méthode de BGS à l'utilisation d'un invariant colorimétrique / espace couleur adapté.

Un bref rappel du fonctionnement de la méthode du Codebook est effectué en Section 2. Un panel d'invariants et

espaces couleur est introduit en Section 3. Enfin, les résultats et observations réalisées sur des données réelles sont consignés dans la Section 4.

2 Codebook

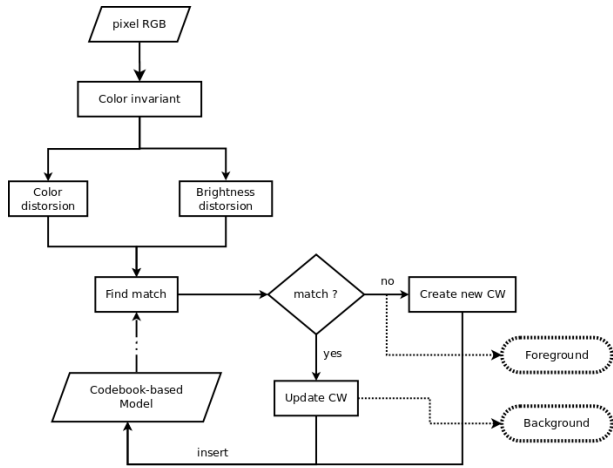


Figure 1 – Diagramme de flux de l'algorithme Codebook (les lignes pleines correspondent à la phase d'apprentissage; les lignes pointillées correspondent à la phase de détection de objets mobiles)

La modélisation de l'arrière-plan d'une scène par la méthode du Codebook proposée par Kim et al. [10] est devenue une référence de plus en plus utilisée dans les domaines de la détection et du suivi d'objets mobiles. Cette méthode est robuste et efficace dans un grand nombre de cas d'utilisation, y-compris les arrière-plans dynamiques (tels que les feuillages, les fontaines, les bords de mers, les drapeaux...) et les légers changements d'illumination.

La remarque qui a mené à la création de cet algorithme est que les fausses détections sont généralement situées dans les zones sombres de l'image. Or, les couleurs sombres (et donc moins lumineuses) sont par nature plus difficiles à différencier et conduisent à une incertitude plus grande sur leur classification finale (fond ou forme). Ainsi, on suppose que la luminosité devrait être un facteur important dans la comparaison des ratios de couleurs entre deux pixels. On observe également au fil du temps que la couleur d'un pixel donné est distribuée sur une droite alignée avec l'origine (0,0,0). Ces observations motivent la création d'un nouveau modèle de couleur permettant une évaluation séparée de la distorsion de la couleur et de la luminosité du pixel.

Le diagramme de flux de l'algorithme Codebook est illustré sur la Figure 1. Nous y retrouvons les phases d'apprentissage et de détection et notre proposition d'ajout d'utilisation d'invariants ou espaces couleur.

2.1 Apprentissage

La méthode du Codebook consiste en un partitionnement de l'image afin de construire un modèle d'arrière-plan pen-

dant une période d'apprentissage. Ce modèle est représenté par une liste de codebooks (1 par pixel) contenant chacun N codewords. Un codeword est créé (ou mis à jour si le pixel observé est similaire à un codeword existant) à chaque itération de l'apprentissage. Il est défini par deux vecteurs contenant respectivement les valeurs RGB du codeword, et plusieurs autres données telles que les valeurs de luminosité minimum et maximum, des informations temporelles et de fréquence d'observation du codeword.

Pendant cette phase d'apprentissage, tout nouveau codeword obtenu (pour un pixel donné) est intégré dans le modèle d'arrière-plan s'il satisfait deux conditions détaillées dans [10] : 1) une contrainte de distorsion de luminosité et 2) une contrainte de distorsion de couleur. La première contrainte nécessite que l'intensité du pixel soit comprise entre les intensités minimum et maximum observées pour ce même pixel. La seconde contrainte fait en sorte que la distorsion de couleur (Formule 1) entre les vecteurs couleur du pixel x_t et du codeword v_i , soit inférieure à un seuil déterminé empiriquement ϵ . La distorsion de couleur revient à caractériser le "décalage" entre ces deux couleurs, en normalisant le vecteur du codeword v_i à l'intensité du pixel x_t .

$$color_{dist}(x_t, v_t) = \sqrt{\|x_t\|^2 - \frac{\langle x_t, v_i \rangle^2}{\|v_i\|^2}} \quad (1)$$

La taille du modèle d'arrière-plan ainsi obtenu (qui est une matrice de Codebooks) est assez importante à l'issue de l'étape d'apprentissage. Cependant, d'après les observations de Kim et al., une taille de 6.5 codewords par codebook (i.e. pour chaque pixel) est suffisante pour avoir un bon compromis entre mémoire et qualité du modèle. Aussi, ce dernier est "nettoyé" des codewords dont on détecte qu'ils pourraient appartenir à des objets mobiles observés pendant l'apprentissage. Pour ce faire, l'algorithme utilise la valeur Maximum Negative Run-Length (MNRL) définie comme étant le plus long intervalle dans la période d'apprentissage pendant lequel le codeword n'a pas été observé. Ainsi, si la valeur MNRL d'un codeword est faible, cela signifie que ce codeword est fréquemment observé; inversement si cette valeur est importante, cela signifie que ce codeword est moins fréquemment observé et donc qu'il n'appartient *a priori* pas à l'arrière-plan.

Notons qu'un codeword avec un MNRL élevé et une fréquence d'apparition importante signifie probablement qu'il s'agit d'un objet resté immobile pendant un moment.

2.2 Détection des objets mobiles

Le modèle obtenu durant la phase d'apprentissage représente les parties de l'image "sans mouvement". Il est alors possible de l'utiliser pour caractériser chaque pixel d'une nouvelle image. Plus simplement, il s'agit de vérifier l'existence dans le modèle d'un codeword répondant aux mêmes contraintes précédemment décrites que celles du pixel observé. S'il existe un codeword du modèle correspondant au pixel observé, alors il est étiqueté comme

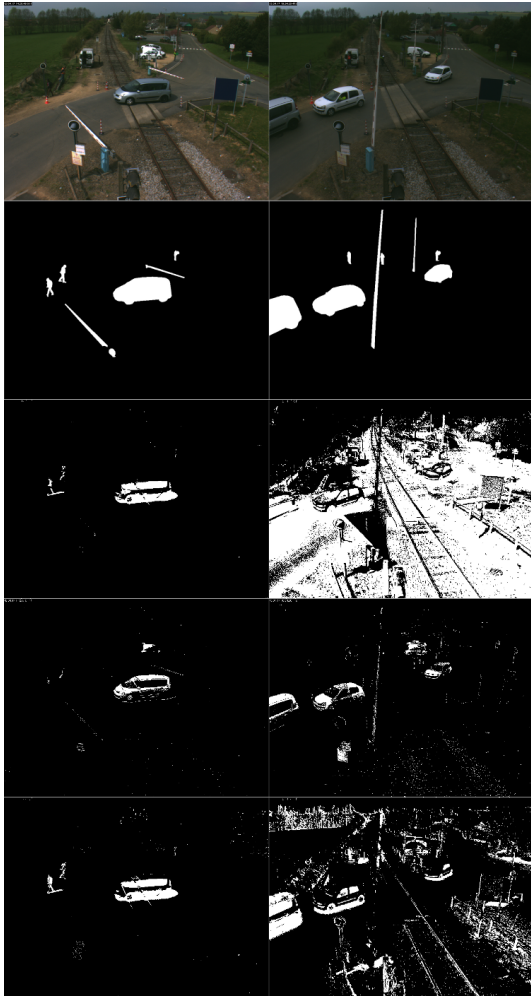


Figure 2 – Illustration des segmentations obtenues avec et sans invariants pour un apprentissage sur la base L1 (12h).
 Colonne 1 : traitement de la base P1. Colonne 2 : traitement de la base P2. De haut en bas : images d'origine, références (vérités terrain), segmentations sans invariant, segmentations avec HSL, segmentations avec RGB-Rank.

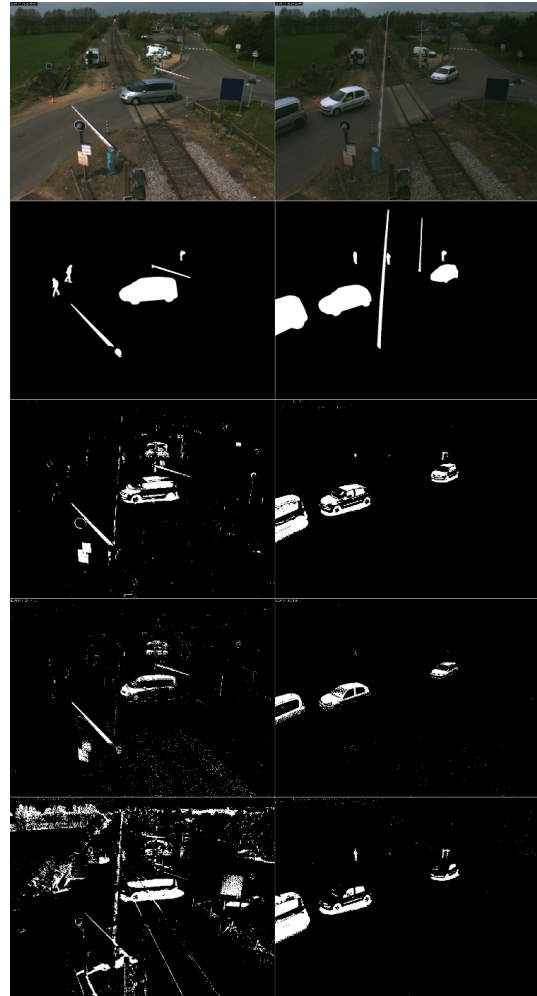


Figure 3 – Illustration des segmentations obtenues avec et sans invariants pour un apprentissage sur la base L2 (16h).
 Colonne 1 : traitement de la base P1. Colonne 2 : traitement de la base P2. De haut en bas : images d'origine, références (vérités terrain), segmentations sans invariant, segmentations avec HSL, segmentations avec RGB-Rank.

appartenant au fond (couleur noire sur les figures 2 et 3) et le codeword correspondant est mis à jour. Sinon, il est étiqueté comme appartenant à un objet mobile (couleur blanche sur les figures 2 et 3).

3 Invariants couleur

Le concept d'invariance couleur assume le fait que nous pouvons percevoir la couleur d'un objet, quelle que soit la source de lumière qui l'illumine. Cette technique nécessite l'estimation de la source de la source de lumière. Il existe plusieurs méthodes permettant d'effectuer cette opération, que nous classons dans trois catégories suivantes :

1. Utilisation des caractéristiques bas-niveau : ces méthodes sont basées sur des statistiques bas-niveau ou un modèle de réflexion dichromatique basé sur la physique ;

2. Utilisation d'une phase d'apprentissage : l'illuminant est déterminé à l'aide d'un modèle calculé sur des données d'apprentissage ;
3. Une combinaison de ces méthodes.

Naturellement, toutes les méthodes d'invariance couleur ne produisent pas la même image. Elles nécessitent souvent de créer un nouveau modèle adapté aux caractéristiques dont elles font usage. Néanmoins, aucun algorithme ne peut être considéré comme universel. Certains travaux consistent en une combinaison de plusieurs stratégies : Gijssen et al. [11] a montré que plusieurs algorithmes d'invariance couleur fonctionnaient mieux sur différents types d'images ; Bianco et al. [12] a proposé un algorithme permettant de choisir le meilleur invariant couleur pour une image donnée grâce à la méthode CART et une forêt de décision. Nos travaux visent quant à eux à observer les ef-

Tableau 1 – *F-Measure background*

Base	Affine Norm	c1c2c3	Chromaticity	Compr. Norm	Greyworld	HSL	L*a*b*	l1l2l3	m1m2m3	none	RGB-rank	YCbCr	YIQ
L1_P1	93,01%	94,12%	94,12%	94,34%	94,34%	94,19%	93,35%	94,02%	92,78%	94,34%	94,30%	48,53%	91,59%
L1_P2	81,63%	96,43%	96,43%	58,05%	58,66%	96,35%	95,69%	83,31%	91,52%	61,04%	92,50%	25,00%	95,45%
Mean L1	87,3%	95,3%	95,3%	76,2%	76,5%	95,3%	94,5%	88,7%	92,1%	77,7%	93,4%	36,8%	93,5%
Rank L1_Px	9	1	2	12	11	3	4	8	7	10	6	13	5
Rank L1_P2	9	1	2	12	11	3	4	8	7	10	6	13	5
L2_P1	78,73%	94,13%	94,12%	93,49%	93,59%	93,87%	89,48%	88,88%	91,62%	93,58%	90,30%	3,90%	19,85%
L2_P2	95,51%	96,43%	96,43%	96,69%	96,68%	96,67%	95,60%	96,53%	95,00%	96,68%	96,55%	49,39%	88,25%
Mean L2	87,1%	95,3%	95,3%	95,1%	95,1%	95,3%	92,5%	92,7%	93,3%	95,1%	93,4%	26,6%	54,0%
Rank L2_Px	11	1	2	6	4	3	10	9	8	5	7	13	12
Rank L2_P1	11	1	2	6	4	3	9	10	7	5	8	13	12

Tableau 2 – *F-Measure foreground*

Base	Affine Norm	c1c2c3	Chromaticity	Compr. Norm	Greyworld	HSL	L*a*b*	l1l2l3	m1m2m3	none	RGB-rank	YCbCr	YIQ
L1_P1	14,64%	0,85%	0,28%	31,12%	31,29%	23,51%	2,87%	28,07%	9,11%	31,80%	28,71%	4,63%	10,79%
L1_P2	7,69%	2,33%	1,44%	3,49%	3,50%	25,80%	4,74%	8,36%	11,82%	3,60%	14,06%	5,14%	24,56%
Mean L1	11,2%	1,6%	0,9%	17,3%	17,4%	24,7%	3,8%	18,2%	10,5%	17,7%	21,4%	4,9%	17,7%
Rank L1_Px	8	12	13	7	6	1	11	3	9	4	2	10	5
Rank L1_P2	6	12	13	11	10	1	8	5	4	9	3	7	2
L2_P1	6,18%	4,15%	1,32%	35,02%	36,27%	34,70%	9,78%	9,35%	9,84%	35,66%	13,36%	3,42%	2,20%
L2_P2	18,41%	1,79%	1,15%	29,11%	28,99%	21,47%	9,55%	25,41%	14,88%	28,53%	26,28%	5,99%	6,93%
Mean L2	12,3%	3,0%	1,2%	32,1%	32,6%	28,1%	9,7%	17,4%	12,4%	32,1%	19,8%	4,7%	4,6%
Rank L2_Px	8	12	13	3	1	4	9	6	7	2	5	10	11
Rank L2_P1	9	10	13	3	1	4	7	8	6	2	5	11	12

fets de plusieurs invariants et espaces couleur sur les segmentations obtenues par l’algorithme du Codebook afin d’améliorer les performances de celui-ci sur des bases d’images réelles acquises dans des conditions d’illumination très hétérogènes.

13 invariants et espaces couleur ont été testés : Chromaticity space (aussi nommé Normalized RGB), Greyworld [13], Comprehensive normalization [14], Affine normalization [15], les espaces couleur c1c2c3, m1m2m3 et l1l2l3 [16], RGB-Rang [17], les espaces couleur YIQ [18], YCbCr [19, 20], YCh1Ch2 [21] et CIE L*a*b* [22].

4 Résultats et observations

4.1 Protocole expérimental

Comme indiqué en Section 1, nous proposons d’améliorer les résultats de segmentations obtenues par l’algorithme Codebook grâce à l’utilisation d’un invariant colorimétrique adapté. Les séquences d’images sur lesquelles nous travaillons ont été acquises dans le cadre du projet PANsafer et surveillent un passage à niveau. Les images de taille 640x480 ont été acquises à l’aide d’une caméra jAI CV-M9CL dotée d’un capteur 3-CCD.

Ces séquences d’images sont issues de scénarios enregistrés afin de détecter des situations dangereuses induites par des comportements ou des blocages involontaires sur les passages à niveau. Ces scénarios consistent par exemple en une stationnarité de véhicule suite à une remontée de file aboutissant au blocage d’un ou plusieurs véhicules sur la voie ferrée, ou encore d’un passage en chicane après abaissement des barrières.

Ces séquences acquises à différents moments de la journée mettent en évidence des différences de couleur et des variations de luminosité. Toutefois, ces acquisitions n’ont pas permis d’observer des différences très marquées et l’illumination globale des séquences reste généralement sta-

ble. Néanmoins, nous avons retenu deux séquences d’images, l’une affiche une intensité lumineuse importante (acquise vers 12h) tandis que l’autre est de plus faible intensité (acquise vers 16h). Cette configuration nous permet de justifier l’importance de l’utilisation d’invariants colorimétriques / espaces couleur pour une meilleure détection des objets mobiles.

Les deux bases d’images d’apprentissage de l’algorithme issues de l’une ou l’autre base d’images sont dénommées L1 (12h) et L2 (16h) ; les bases utilisées pour la détection fond/forme sont dénommées P1 (12h) et P2 (16h). Un test impliquant un apprentissage sur la base L1 et un traitement de la base P2 est donc nommé L1_P2.

Les paramètres utilisés pour l’algorithme Codebook sont les suivants : 75 images d’apprentissage par base, $\alpha = 0.4$, $\beta = 1.2$, $\epsilon_1 = 0.2$, $\epsilon_2 = 100$. Ces valeurs sont les valeurs relevées empiriquement par les auteurs de l’algorithme Codebook.

La stratégie de détection d’objets mobiles proposée couplant l’utilisation d’un invariant colorimétrique / espace couleur et l’algorithme du Codebook se déroule en trois étapes (comme illustré sur la Figure 1) :

1. Appliquer un opérateur d’invariance colorimétrique / espace couleur sur les images de la base d’apprentissage ;
2. Construire le modèle du Codebook sur cette base d’apprentissage ;
3. Traiter chaque base de test avec le même opérateur d’invariance colorimétrique / espace couleur que celui utilisé pour la base d’apprentissage.

Rappelons que le modèle d’arrière-plan obtenu à l’étape 2 n’est à aucun moment mis à jour. L’effet attendu des invariants colorimétriques est la réduction des différences de couleur et d’intensité visibles entre les images d’apprentissage et les images de test.

4.2 Evaluation des résultats

Afin d'évaluer la performance de l'approche proposée par rapport à une segmentation de référence, nous avons évalué les segmentations obtenues à l'aide d'un critère de F-Mesure appliqué sur le fond (background) et sur les objets mobiles (foreground).

Un pixel est considéré comme Vrai Positif (TP) lorsqu'il est correctement classifié comme appartenant à un objet mobile, Vrai Négatif (TN) lorsqu'il est correctement classifié comme fond, Faux Positif (FP) lorsque le pixel est incorrectement détecté comme objet mobile et Faux Négatif (FN) lorsqu'il est détecté comme fond alors qu'il devrait être détecté comme objet mobile. La valeur de F-Mesure FM_c , classiquement utilisée comme mesure de justesse des segmentations, est calculée séparément pour le fond (bg) et pour les objets mobiles (fg) de la manière suivante (Eq. 2) :

$$\begin{aligned} Recall_{fg} &= \frac{TP}{TP + FN}, \\ Recall_{bg} &= \frac{TN}{TN + FP}, \\ Precision_{fg} &= \frac{TP}{TP + FP}, \\ Precision_{bg} &= \frac{TN}{TN + FN}, \\ FM_c &= \frac{2 \cdot (Recall_c \cdot Precision_c)}{Recall_c + Precision_c} \end{aligned} \quad (2)$$

Avec $c \in \{bg, fg\}$ selon que la valeur s'applique au fond (bg) ou aux objets mobiles (fg). La F-Mesure est la moyenne harmonique du Rappel et de la Précision. Plus la F-Mesure est grande, meilleure est la segmentation. Cette valeur est déterminée séparément pour le fond et les objets mobiles pour observer les effets des invariants colorimétriques sur ces deux classes. En effet, il est plus probable d'observer un taux de faux positifs élevé dans l'arrière-plan des images lorsque l'algorithme traite les images d'un horaire donné puisque le modèle d'apprentissage ne correspond pas à cet horaire. Ainsi, l'utilisation d'invariants colorimétriques devrait permettre d'améliorer davantage la détection dans ces zones.

Evidemment, de bons résultats sont attendus dans le cas où la base d'apprentissage est issue du même horaire que la base utilisée pour la détection des objets mobiles (L_x_P_x). Pour éviter de trop influencer les résultats, ces valeurs ne sont pas prises en compte dans le calcul de la F-Mesure moyenne prend seulement en compte les résultats pour L_x_P_y (avec $x \neq y$).

Les tableaux 1 et 2 illustrent respectivement la F-Mesure calculée pour le background et le foreground sur la totalité des images des deux bases de test. Nous y présentons les résultats de segmentations obtenues avec et sans invariant colorimétrique. Pour chacun de ces deux tableaux et pour chaque horaire d'apprentissage, nous avons effectué deux classements des résultats : le premier prend en compte la

moyenne des F-Mesures calculées sur les deux bases de test (L1_P_x), le second ne prend en compte que les résultats obtenus dans le cas d'horaires d'apprentissage et de test différents (L1_P2 et L2_P1).

La Figure 2 illustre les résultats de segmentations obtenues lorsque l'apprentissage est effectué sur les images de la base L1 (12h). Nous notons que les deux invariants qui donnent les meilleurs résultats sont HSL et RGB-Rank.

Dans le premier cas (1^{ère} colonne), nous observons des résultats quasi identiques voire meilleurs lorsque nous n'utilisons pas d'invariant colorimétrique / espace couleur (même si leur utilisation ne dégrade pas beaucoup la qualité de la segmentation). Dans ce cas précis, les F-Mesures de l'arrière-plan se situent autour de 94% (sans invariant, avec HSL et avec RGB-Rank). Les F-Mesures pour les objets mobiles se situent à 31% sans invariant, 28% avec RGB-Rank et 23% avec HSL.

En revanche, on constate que dès lors que l'apprentissage et la base traitée ne sont plus les mêmes (L1_P2, 2^{ème} colonne), les segmentations obtenues sans invariant comportent un très fort taux de faux positifs. A l'inverse, l'utilisation d'invariants couleur permet d'obtenir de meilleures segmentations. Précisément, les F-Mesures du fond se situent à plus de 90% pour HSL et RGB-Rank alors que celle sans invariant est de 61%. Quant aux F-Mesures des objets mobiles, elles sont de 25% pour HSL, 14% pour RGB-Rank et seulement 3% sans invariant.

La Figure 3 présente les mêmes traitements avec un apprentissage effectué sur les images de la base L2 (16h, plus sombre). Cet exemple est un peu plus nuancé. De même qu'en Figure 2, la colonne 2 correspond au traitement où la base d'apprentissage est la même que celle qui permet la détection des objets mobiles. La colonne 1 correspond au traitement L2_P1. Ici, on remarque que la segmentation obtenue sans invariant colorimétrique est largement acceptable. Sa F-Mesure est d'environ 93% pour l'arrière-plan, environ 35% pour les objets mobiles. La raison à cela se trouve dans le modèle couleur utilisé par l'algorithme du codebook (distorsion de couleur/distorsion d'intensité) : l'apprentissage effectué sur des images plus sombres crée une distorsion de couleur moins importante lorsque l'algorithme traite des images plus claires. Or, la validation des contraintes de distorsions décrites dans la Section 2 nécessite pour l'une (couleur) qu'elle soit inférieure à un seuil donné et pour l'autre (intensité) que sa valeur se situe dans un intervalle déterminé pendant l'apprentissage.

Même si nous constatons une certaine proximité des résultats avec et sans l'utilisation d'invariant colorimétrique selon la base testée, il n'en est pas moins vrai que l'usage d'un invariant colorimétrique / espace couleur adapté est généralement bénéfique dans le cadre de la détection d'objets mobiles dans des séquences d'images.

5 Conclusion

Dans ce papier, nous avons apporté la confirmation de l'intérêt non négligeable qui doit être porté aux invari-

ants colorimétriques dans le cadre de la détection d'objets mobiles. Les résultats que nous avons pu obtenir montrent que l'invariance couleur apporte une amélioration dans tous les cas, mais que le choix d'un invariant colorimétrique au détriment d'un autre est dépendant de plusieurs paramètres, notamment des conditions d'observation (lumière forte/faible.) Nous formulons donc l'hypothèse que la définition d'une méthode générique n'est pas triviale et qu'une fonction décisionnelle doit permettre de déterminer l'invariant colorimétrique à utiliser selon les conditions d'acquisition. Il est en effet possible d'obtenir de meilleurs résultats dans la majorité des cas avec l'utilisation d'un invariant couleur.

Nos futurs travaux porteront sur la définition d'un nouvel espace colorimétrique de façon à rendre à les méthodes de BGS insensibles aux légers changements de lumière et plus robustes aux forts changements opérant tout au long de la journée (du lever au coucher du soleil).

Références

- [1] D.-N. Truong Cong, L. Khoudour, C. Achard, C. Meurie, et O. Lezoray. People re-identification by spectral classification of silhouettes. *Signal Processing*, 90(8) :2362 – 2374, 2010. Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.
- [2] Houssam Salmane, Yassine Ruichek, et Louahdi Khoudour. Using hidden markov model and dempster-shafer theory for evaluating and detecting dangerous situations in level crossing environments. Dans *Mexican International Conference on Artificial Intelligence (MICAI'2012)*, LNCS, San Luis Potosi, Mexico, 2012.
- [3] Houssam Salmane, Yassine Ruichek, et Louahdi Khoudour. Gaussian propagation model based dense optical flow for objects tracking. Dans *International Conference on Image Analysis and Recognition (ICIAR'2012)*, LNCS, pages 234–244, Aveiro, Portugal, 2012.
- [4] Thierry Bouwmans, Fida El Baf, Bertrand Vachon, et al. Background modeling using mixture of gaussians for foreground detection-a survey. 2008.
- [5] Thierry Bouwmans et Fida El Baf. Statistical background modeling for foreground detection : A survey. *Handbook of Pattern Recognition and Computer*, 2010.
- [6] Alan M. Mcivor. Background Subtraction Techniques. 2000.
- [7] C. Wren, A. Azarbayejani, T. Darrell, et A. Pentland. Pfunder : real-time tracking of the human body. pages 51 –56, oct 1996.
- [8] Thanarat Horprasert, David Harwood, et Larry S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. *IEEE Frame-Rate Applications Workshop Kerkyra, Greece*, pages 1–19, 1999.
- [9] C. Stauffer et W.E.L. Grimson. Adaptive background mixture models for real-time tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2 :2246, 1999.
- [10] K. Kim, T. H. Chalidabhongse, D. Hanuood, et L. Davis. Background modeling and subtraction by codebook construction. 2004.
- [11] A. Gijssenij et T. Gevers. Color constancy using natural image statistics and scene semantics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4) :687 –698, avril 2010.
- [12] S. Bianco, G. Ciocca, C. Cusano, et R. Schettini. Automatic color constancy algorithm selection and combination. *Pattern Recognition*, 43(3) :695 – 705, 2010.
- [13] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1) :1 – 26, 1980.
- [14] Graham D. Finlayson, Bernt Schiele, et James L. Crowley. Comprehensive colour image normalization. 1998.
- [15] Stephan Obdrzalek, Jiri Matas, et Ondrej Chum. On the interaction between object recognition and colour constancy. *Proc. International Workshop on Color and Photometric Methods in Computer Vision*, 2003.
- [16] Theo Gevers et Arnold W.M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32(3) :453 – 464, 1999.
- [17] Graham D. Finlayson, Steven D. Hordley, Gerald Schaefer, et Gui Yun Tian. Illuminant and device invariant colour using histogram equalisation. *Pattern Recognition*, 38(2) :179–190, 2005.
- [18] Walter H. Buchsbaum. *Color TV Servicing, third edition*. Englewood Cliffs, NJ : Prentice Hall, ISBN 0-13-152397-X, 1975.
- [19] Recommendation itu-r bt.709-5, 1990-2002.
- [20] Recommendation itu-r bt.601-5, 1982-1995.
- [21] T Carron. *Segmentation d'images couleur dans la base Teinte Luminance Saturation : approche numérique et symbolique*. Thèse de doctorat, Université de Stanford, 1995.
- [22] K. McLaren. *The development of the CIE 1976 (L*a*b*) uniform colour-space and colour-difference formula*. *Journal of the Society of Dyers and Colourists* 92., 1976.