Banque de données et banc d'essai en détection de changement

par

Nil Goyette

Mémoire présenté au Département d'informatique
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES

UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, 20 mars 2013

Your file Votre référence
ISBN: 978-0-494-95142-2

Our file Notre référence
ISBN: 978-0-494-95142-2

Le 20 mars 2013

le jury a accepté le mémoire de Monsieur Nil Goyette
dans sa version finale.

Membres du jury

Professeur Pierre-Marc Jodoin
Directeur de recherche
Département d'informatique

Professeur Marc Frappier
Évaluateur
Département d'informatique

Professeur Maxime Descoteaux
Président rapporteur
Département d'informatique

Sommaire

Les caméras de vidéosurveillance sont de plus en plus présentes dans notre société, à un point tel que les séquences vidéo sont souvent enregistrées sans être regardées par des agents de sécurité. Il convient donc de créer des algorithmes qui vont effectuer le même travail d'analyse que des surveillants humains. Bien qu'il y ait des inquiétudes au niveau de la vie privée, on peut envisager maintes applications, toutes au service de la société.

La détection de changement est à la base de bon nombre d'applications en analyse vidéo. Elle consiste à détecter tout changement intéressant dans une séquence capturée par une caméra fixe. Bien que les méthodes gèrent mieux les difficultés inhérentes à ce problème, il n'y a pas encore de solution définitive à la détection de changement. Avec des milliers de méthodes disponibles dans la littérature, il est présentement très difficile, voire impossible, de comparer ces méthodes et d'identifier lesquelles répondent mieux aux différents défis. Les auteurs font face au même problème lorsqu'ils désirent se comparer à l'état de l'art.

Pour faire face à cette situation, nous avons créé un banc d'essai en détection de changement. Ceci inclut la création d'une banque de données d'envergure, d'une méthode d'évaluation quantitative équitable et d'un site web pour consulter le classement et télécharger les résultats de segmentation des compétiteurs. Des outils et de la documentation pour utiliser ces derniers sont aussi offerts, le tout accessible gratuitement et simplement sur Internet. Comme le but est de devenir le standard *de facto*, le projet est suffisamment complet et intéressant pour convaincre la communauté scientifique de l'adopter. Ce faisant, nous avons créé notre propre programme d'annotation libre, rassemblé et programmé une dizaine de méthodes de détection de changement, puis déterminé les meilleures méthodes et les difficultés auxquelles la communauté devrait s'attaquer au cours des prochaines années.

L'atelier organisé à CVPR 2012, le grand nombre de soumissions et le bon achalandage du site sont des indicateurs encourageants quant à la réussite de notre travail. Il est encore trop tôt pour confirmer quoi que ce soit car l'adoption d'un nouveau standard prend du temps, mais notre pronostic est positif.

Mots-clés : détection de changement ; soustraction de fond ; banque de données ; évaluation

Remerciements

Mes pensées vont à tous ceux qui m'ont supporté durant mes recherches, en particulier mon directeur de maîtrise, Pierre-Marc Jodoin, homme de carrière, acériculteur, *câlleur* de danses traditionnelles, excellent pédagogue et infatiguable chercheur. J'aimerais aussi remercier les altruistes et sympathiques membres du RECSUS, collègues et amis toujours prêts à discuter science, philosophie et religion ; des *geeks* avec qui j'ai passé d'innombrables bons moments. Sans oublier Vanessa Arès, ma copine de longue date, artiste, graphiste, écolo et écrivaine, qui accepte stoïquement l'homme solitaire et passionné par l'informatique que je suis.

Maxime Caron et Mathieu Germain méritent aussi ma gratitude. D'un côté, la motivation contagieuse, la bonne humeur et la passion de Maxime, de l'autre, le pragmatisme et le dogmatisme de Mathieu, ainsi que le monde d'absolus et de superlatif dans lequel il vit. Avec une technique bien différente, ils m'ont toujours poussé à donner le meilleur de moi-même et je leur en suis reconnaissant.

Une mention spéciale également pour les créateurs derrière les livres, les animés, les films et les vidéos exceptionnelles sur Internet, particulièrement les séries de Reid Gower, de Phil Hellenes et les *TED Talks*. Ces derniers ont contribué à me faire évoluer et à forger l'homme que je suis. Il est impossible d'être quelqu'un d'autre, mais il est toujours possible de s'améliorer considérablement. *Fake it till you make it.*

Table des matières

Introduction

La vidéosurveillance ne cesse de prendre de l'ampleur depuis son apparition. Malgré l'inquiétude de certains groupes de protection de la vie privée et des libertés civiles, on ne peut ignorer les diverses applications promises par la vidéosurveillance. De nos jours, la quantité de vidéos capturées est tellement énorme que la plupart des vidéos sont enregistrées sans être analysées. Une infime partie de ces séquences est regardée à des fins d'enquêtes après les évènements. À l'aide de la vidéosurveillance, il sera possible dans quelques années d'analyser ces séquences en temps réel. Le grand objectif de la vidéosurveillance est d'effectuer la même analyse qu'un gardien de sécurité, mais sans fatigue et manque de concentration, et sur un grand nombre de caméras.

La détection de changement est une tâche bas niveau fondamentale en vidéosurveillance. À partir d'une séquence vidéo capturée par une caméra fixe, elle fournit un masque binaire indiquant où se trouve le mouvement. Ce masque est ensuite utilisé dans de nombreuses autres applications de surveillance. Elle est directement reliée à de nombreuse tâches dites intelligentes, telles que le suivi d'objets, l'identification et la classification. La détection de changement est souvent considérée comme une étape de post-traitement essentielle. Le perfectionnement des algorithmes de détection de changement laisse envisager une performance accrue pour les méthodes qui en dépendent.

Une recherche rapide sur IEEExplore[1] nous montre un intérêt toujours grandissant pour la détection de changement. Avec 15 articles en 2000, 86 en 2005 et 228 en 2012, il est clair qu'il y a toujours des recherches dans ce domaine. Une recherche plus générale sur la détection de mouvement[2] retourne environ 1200 résultats pour la seule année 2011 !

1. « *background subtraction* » sur `http://ieeexplore.ieee.org`
2. « *motion detection* » sur `http://ieeexplore.ieee.org`

Cette persévérance des chercheurs est justifiée considérant que, à ce jour, il ne semble y avoir aucun algorithme qui réponde adéquatement aux difficultés usuelles d'une vraie scène [1]. On peut alors se demander laquelle de ces méthodes est la meilleure pour un cas d'utilisation particulier, et même de façon générale. Il est présentement très difficile, voire impossible, de répondre à une telle question.

Les auteurs ont l'habitude de comparer leur méthode avec une ou plusieurs autres méthodes, mais la comparaison qui en résulte est trop souvent biaisée. Il n'y a en ce moment aucune banque de données standard largement utilisée dans la communauté de détection de changement. Les comparaisons sont alors effectuées en utilisant toutes sortes de séquences, parfois du cru des auteurs, parfois prises sur une des banques de données existantes. De plus, face à la difficulté de comprendre et de programmer les méthodes récentes, les auteurs choisissent fréquemment des méthodes simples et très connues pour se comparer. Aussi, les auteurs ont rarement la même méthodologie de comparaison car il n'y a encore aucune mesure qui fait l'unanimité dans ce domaine. Tout ceci permet aux scientifiques de se déclarer meilleurs que la concurrence ; une affirmation qui ne peut, de toute évidence, être vraie pour l'ensemble des travaux dans le domaine. En résumé, les auteurs ne se comparent pas sur une base commune ; il est donc impossible de savoir quelle méthode obtient réellement les meilleurs résultats.

Étant donné que la communauté scientifique n'a adopté aucun standard pour l'évaluation des méthodes de détection de mouvement, le présent travail a pour objectif de répondre aux questions suivantes :

1. Comment déterminer quelle méthode de détection de mouvement est la meilleure ?
2. À quel niveau une méthode est meilleure que les autres ?
3. Quelle(s) métrique(s) utiliser pour classer les méthodes entre elles ?
4. Quelles données vidéo utiliser pour valider les méthodes ?
5. Comment définir la réalité de terrain pour ces vidéos ?
6. Comment s'assurer que la procédure d'évaluation n'est pas biaisée en faveur d'une catégorie de méthodes ?
7. Comment convaincre la communauté de traiter nos données ?

En réponse à ces questions, nous avons réalisé les cinq contributions suivantes :

1. Nous avons défini une métrique équitable permettant de classifier les méthodes.
2. Nous avons développé un programme d'annotation libre d'utilisation permettant d'extraire la réalité de terrain d'une vidéo. Ces dernières comprennent 5 étiquettes afin d'assurer une évaluation précise.
3. Nous avons mis sur pied une banque de données comprenant 31 vidéos distribuées dans 6 catégories différentes. Les vidéos de chaque catégorie illustrent une difficulté typique en détection de mouvement. Cette base de données (ainsi que la métrique au point 1) fut élaborée de façon à éviter de favoriser une catégorie de méthodes particulière.
4. Nous avons également mis sur pied un site Internet présentant les résultats pour chaque méthode dans chaque catégorie. Grâce à ce site Internet, toute personne peut télécharger la banque de données et soumettre ses résultats automatiquement.
5. Nous avons rassemblé ou programmé une dizaine de méthodes afin de rendre les résultats disponibles sur le site et d'effectuer quelques tests sur ces derniers, ce qui motive la communauté à comparer leurs méthodes sur notre site.
6. Forts de ces résultats, nous avons pu déterminer les meilleures méthodes à ce jour. Nous avons également pu trouver bon nombre de problèmes non résolus auxquels la communauté scientifique devra s'attaquer au cours des prochaines années.

Hormis ces contributions, ces travaux ont mené à la publication d'un article [1] et à l'organisation d'un atelier à la conférence CVPR 2012, ainsi qu'à la soumission prochaine d'un article journal. À ce jour, le site `www.changedetection.net` répertorie les résultats de 26 méthodes, un sommet dans le domaine.

Dans les prochains chapitres, il sera question de la vidéosurveillance, de la détection de changement et du problème des comparaisons biaisées. Ensuite, nous présenterons un état de l'art sur les banques de données et sur les méthodes d'évaluation. Puis, nous expliquerons ce que nous croyons être la solution aux problèmes de comparaison vécus par la communauté. Notre solution est analogue au célèbre site de vision de Middlebury [3] qui a en partie corrigé les problèmes de comparaison de la communauté de stéréovision. Nous terminerons ensuite par nos contributions et quelques résultats.

3. `http://vision.middlebury.edu/stereo/`

Chapitre 1

Détection de changement

Dans ce chapitre, nous présentons une mise en contexte sur la vidéosurveillance et un survol de l'état de l'art en détection de changement. Ceci inclut une définition du concept de « changement », des difficultés usuelles rencontrées par les chercheurs et d'un état de l'art des méthodes. Nous terminerons par un survol de la situation présente qui est fort problématique pour la recherche, mais que nous proposerons d'améliorer dans les prochains chapitres. Les bases seront donc introduites afin d'avoir une idée claire du domaine, ce qui permettra au lecteur de mieux comprendre le banc d'essai et son importance pour les chercheurs.

1.1 Vidéosurveillance et analyse vidéo

La vidéosurveillance a débuté avec des caméras analogiques et des gardiens de sécurité qui regardaient de nombreux écrans. Le but était de protéger des lieux importants, tels que des services publics stratégiques, des centrales nucléaires, des casinos et autres [2]. Les caméras analogiques étaient dispendieuses et les séquences stockées sur de nombreuses cassettes. De nos jours, les caméras se sont démocratisées et sont présentes dans la plupart des lieux publics. Ceci est grandement dû à la diminution du coût des caméras et du stockage. L'appariton de la technologie numérique et des caméras IP a permis d'enregistrer les séquences sur disques durs ; il est alors possible de les analyser à l'aide d'un ordinateur.

Les techniques d'analyse vidéo ont changé en réponse à de nombreuses études démontrant les limites de la surveillance humaine. Après 20 minutes à regarder des écrans de surveillance vidéo, l'attention de la plupart des individus chute sous un niveau acceptable [2]. Certaines études mentionnent que le rapport entre le nombre d'écrans et le nombre de caméras peut se situer entre 1:4 et 1:78 dans certains réseaux de vidéosurveillance [3]. Or, un surveillant ne peut suivre attentivement 9 à 12 caméras pendant plus de 15 minutes [4]. Dans un réseau de surveillance comprenant 100 caméras et 3 opérateurs, seulement 3% des écrans sont surveillés à un moment donné [3]. On estime à 1 sur 1 000 la probabilité de réagir sur le fait à un évènement capté par un réseau de caméras de surveillance [2]. Bien que les données varient d'une étude à l'autre, les conclusions restent les mêmes : surveiller des caméras pendant plusieurs heures est inefficace. Étant donné ce constat, la vidéosurveillance est surtout utilisée à des fins d'enquête lorsque les gardiens savent ce qu'ils cherchent, par exemple pour retrouver les responsables d'un vol ou de vandalisme. On comprend alors l'importance de remplacer les surveillants par des logiciels d'analyse vidéo. Pour ce faire, les chercheurs ont développé des modèles d'analyse vidéo, qui consistent à reproduire l'analyse qu'effectuerait un humain en regardant les séquences vidéo provenant de caméras de surveillance [2].

Les applications propres à l'analyse vidéo sont nombreuses et diversifiées, surtout en surveillance temps réel. On y retrouve entre autres la détection d'anomalies [5], la recherche d'individus [6], l'analyse de trafic [7] et la détection de vol à l'étalage [8]. L'analyse vidéo fonctionne 24 heures sur 24, sept jours sur sept et elle libère le personnel de sécurité d'une surveillance continue. Le but est généralement d'attirer l'attention des agents de sécurité sur les évènements importants pendant (ou même avant) qu'ils ne se passent ; ce qui leur permet de prévenir les problèmes.

D'autres applications permettent de récolter des statistiques pendant ou après les évènements ; par exemple, pour estimer le nombre de personnes dans une foule [9], ou le débit de piétons qui entrent dans un corridor [10]. Ce type d'analyse peut aider les décideurs à mieux gérer des évènements et des travaux futurs.

L'analyse vidéo permet aussi la recherche rapide d'évènements pertinents dans les séquences vidéo archivées [11]. Ceci peut transformer une recherche de plusieurs heures de vidéo en une recherche de quelques secondes. Ce type de système serait très utile aux

policiers londoniens. Ils ont accès au plus grand réseau de CCTV [1] au monde, mais ils l'utilisent très peu car ils n'ont pas les outils adéquats pour rechercher dans cette immense collection de vidéos [2].

L'analyse vidéo peut aussi réduire la bande passante et l'espace d'archivage nécessaires en ne transmettant ou en n'enregistrant que les données sur les évènements pertinents [12]. Il est déjà possible de compresser les vidéos, mais réduire la quantité de données transmises est plus optimal.

L'intérêt pour la vidéosurveillance est aussi justifié par le fait que c'est présentement un marché lucratif et que les prévisions sont très positives. Avec des revenus mondiaux supérieurs à 7 milliards de dollars en 2007 [2], 10 milliards en 2011 et des prévisions de 25 milliards en 2016 [13], le domaine a tout pour motiver l'industrie et la recherche. Bref, l'analyse vidéo est la solution à un problème réel et important, et elle a un énorme potentiel qui ne cesse de croître avec la recherche.

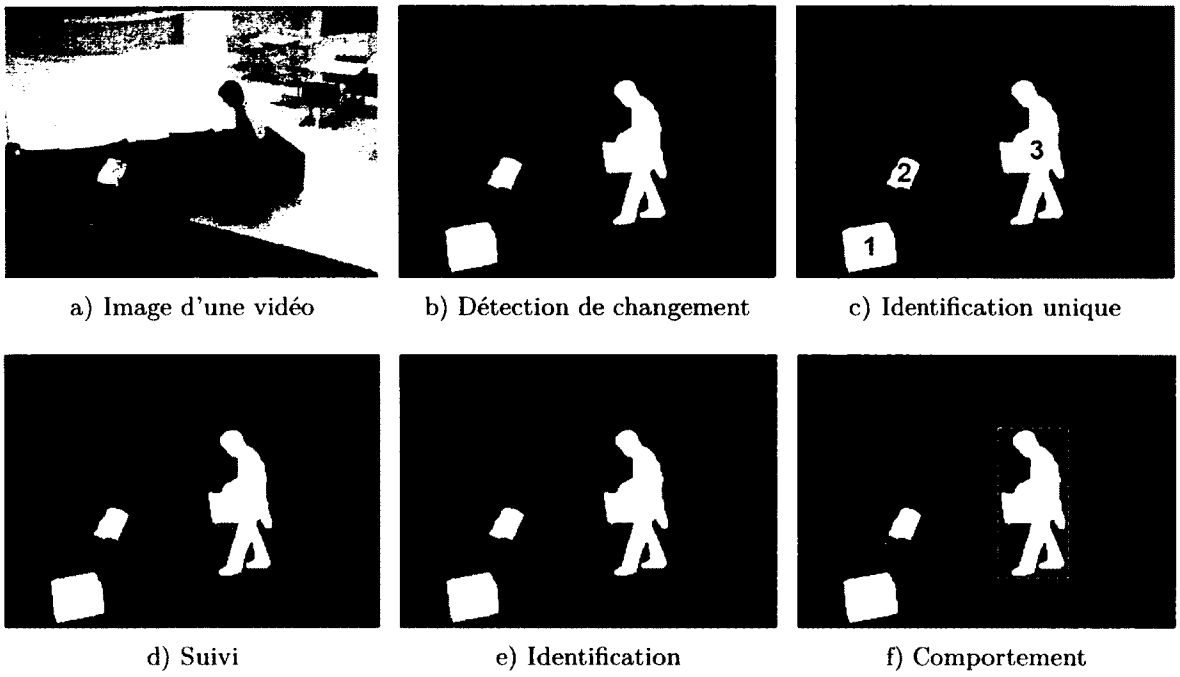
La détection de changement est à la base de bon nombre de systèmes d'analyse vidéo [2]. Les recherches en détection de changement contribuent donc à augmenter leurs performances.

1.2 Étapes de traitement des systèmes d'analyse vidéo

La détection de changement est rarement une fin en soi. Les fonctions d'analyse développées pour les systèmes de vidéosurveillance comportent différents niveaux d'analyse, allant des pixels, aux objets, puis aux comportements [2]. Ces étapes de traitement sont illustrées à la figure 1.1.

Bon nombre de systèmes impliquent 5 niveaux de traitement : 1) détection de changements, 2) segmentation d'objets en mouvement, 3) suivi d'objets, 4) classification et identification d'objets, puis finalement 5) la classification d'activités et de comportements. Ces étapes peuvent alors se diviser en plusieurs autres tâches connexes. Parmi celles-ci, on retrouve les systèmes d'indexation et de recherche vidéo [11], de comptage, d'identification

1. télévision à circuit fermé



a) Image d'une vidéo b) Détection de changement c) Identification unique

d) Suivi e) Identification f) Comportement

FIGURE 1.1 – 5 grandes étapes de l'analyse vidéo. Le résultat de chaque étape utilise le résultat précédent. Les boîtes englobantes, vecteurs et nombres sont présents uniquement à des fins de visualisation ; ces informations ne sont pas dans l'image, elles sont dans un fichier de description. f) Homme s'asseyant en lignes pointillées et objets abandonnés en lignes solides.

de plaques d'immatriculation, de détection d'apparition et de disparition d'objets, d'actions suspectes ou inhabituelles, d'intrusions, de personnes malades, de problèmes sur les voies routières et beaucoup d'autres [2, 4]. Plusieurs systèmes obtiennent déjà d'excellents résultats dans un environnement contrôlé, mais comme le précise le rapport du Centre de Recherche Informatique de Montréal, il s'avère primordial d'améliorer la robustesse et la précision de la plupart des algorithmes de détection, de suivi et de reconnaissance existants, en présence de conditions changeantes de l'environnement [2]. Pour être adoptés par les consommateurs, les systèmes de vidéosurveillance devront bien gérer les différentes conditions météorologiques et s'ajuster aux changements d'illumination de la scène.

1.3 Définition du changement

Au sens strict, la moindre variation de signal dans la valeur d'un pixel peut représenter du mouvement. Cependant, une telle définition est trop contraignante pour des raisons qui seront expliquées à la section 1.5. La définition du mouvement est adaptée en fonction des besoins particuliers de la vidéosurveillance et de la sécurité. Il existe toujours des systèmes ayant des besoins particuliers, mais en règle générale, les mouvements d'origine naturelle et répétitifs ne sont d'aucun intérêt et doivent donc être ignorés. Parmi ceux-ci, on retrouve entre autres les nuages, l'eau, le soleil, les arbres, un drapeau dans le vent, tout changement d'illumination (incluant l'ombre), les réflexions, un ventilateur, des rideaux, etc. La plupart des éléments de cette énumération se résument à des mouvements causés par la force du vent et par des changements d'illumination locaux ou globaux. Finalement, on considère en mouvement tout humain, animal ou objet de construction humaine ayant un mouvement non répétitif doit être considéré comme du changement dans une vidéo.

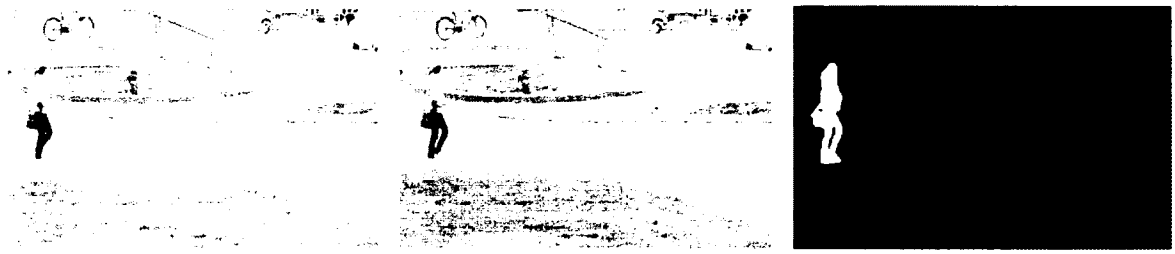
Cette définition très sélective du mouvement entraîne son lot de difficultés. Ces dernières vont varier selon la méthode choisie et selon le type de vidéo. En effet, les difficultés ne sont pas les mêmes dans une séquence vidéo thermique que dans une séquence vidéo couleur. Elles sont aussi différentes dans une séquence 3D, capturée par exemple avec une XBox Kinect. Les données en entrée influencent énormément la qualité des résultats. Dans le cadre de ce travail, nous considérons uniquement des vidéos couleur standards et thermiques, mais il existe aussi des vidéos en noir et blanc, en 3D, des configurations avec plus d'une caméra, des caméras *time-of-flight*, de vision de nuit, etc. Il faut évidemment adapter le choix de la ou des caméras selon l'utilisation.

1.4 Détection de changement

Voyons maintenant le premier élément de ce processus : la détection de changement [2]. Étant donné une séquence vidéo acquise par une caméra fixe, la détection de changement consiste en la création de plans binaires χ_t représentant les changements pertinents dans

2. « Détection de changement » et « détection de mouvement » sont utilisés de façon interchangeable dans la littérature. Dans ce document, le terme « détection de changement » sera utilisé.

la scène : $\chi_t \in [0,1]$ où 0 signifie un pixel lié à un arrière-plan statique et 1, à un objet en mouvement. Ce traitement est visible à la figure 1.2. Une séquence vidéo est généralement constituée d'une série d'images prises à intervalles fixes Δt, contenant un ou des objets en mouvement devant une image de fond fixe. Mathématiquement, $I_t \in [(0,0,0),(255,255,255)]$ est une image au temps t constituée de pixels couleur RGB.



I_t I_{t+1} χ_{t+1}

FIGURE 1.2 – χ_{t+1} illustre l'emplacement du mouvement dans la scène au temps $t+1$, donc dans l'image I_{t+1}.

La méthode la plus simple et largement implémentée pour effectuer ce traitement est souvent appelée « soustraction de fond ». Tel que présentée dans la figure 1.3, elle consiste à construire un modèle de l'arrière-plan B, puis à signaler sur un masque binaire χ_t toute différence significative entre l'image courante I_t et le modèle B.

B I_t χ_t

FIGURE 1.3 – Le masque binaire χ_t est créé en seuillant la différence entre B_t et I_t, donc $\chi_t = \|B_t - I_t\| > \tau$ où τ est un seuil choisi par l'utilisateur.

Dans ce modèle, chaque pixel de l'arrière-plan est représenté par trois valeurs réelles entre 0 et 255, de sorte que le modèle peut être visualisé comme une image. La construction du modèle de l'arrière-plan B peut alors être gardée au plus simple en utilisant une image sans objet en mouvement.

Une fois le modèle B construit, on compare ce modèle avec l'image en cours I_t, ce qui produit un masque binaire χ pour chaque image I_t. Plus précisément, chaque pixel s de l'image I_t est comparé au modèle de l'arrière-plan :

$$\chi_{t,s} = \begin{cases} 1 & \text{si } d(I_{t,s}, B_s) > \tau \\ 0 & \text{sinon} \end{cases} \tag{1.1}$$

où d est une mesure de distance entre un pixel du modèle et de l'image en cours. En niveaux de gris et en couleur, une simple distance euclidienne peut être utilisée. Le seuil τ dépend normalement de l'application en cours. Si l'application de surveillance requiert un nombre faible de faux positifs (voir section 2.2), il est préférable d'augmenter le seuil.

Cette soustraction de fond décrite dans les derniers paragraphes se base sur 4 hypothèses :

- le modèle de l'arrière-plan doit est connu et constant ;
- la caméra est parfaitement stable ;
- les objets en mouvement sont d'une couleur différente de l'arrière-plan ;
- tout changement brusque dans la couleur d'un pixel est causé par un objet en mouvement.

Malheureusement, ces critères ne sont jamais totalement respectés dans un environnement réel. Étant donné cette situation, il est normal que les résultats ne soient pas optimaux. Lorsque ces hypothèses ne sont pas respectées, la segmentation sera de mauvaise qualité ; les objets seront incomplets et il y aura plusieurs artéfacts fantômes.

Étant donné ce non-respect des hypothèses, certaines modifications ont été apportées à la soustraction de fond de base afin qu'elle soit plus robuste. Une de ces modifications consiste à mettre à jour le modèle de l'arrière-plan. Cela permet de prendre en compte les changements d'illumination et autres modifications apportées à la scène. La mise à jour est moins importante dans un environnement intérieur contrôlé car l'arrière-plan ne devrait pas changer, mais elle devient vite importante dans un environnement moins contrôlé où des objets sont enlevés et ajoutés et où l'arrière-plan est modifié par des phénomènes naturels comme le soleil et le vent.

La mise à jour est également utile pour incorporer dans le modèle les objets stables depuis plusieurs minutes. Si un objet cesse de bouger pendant plus de 4 minutes, il est normalement absorbé par l'arrière-plan, donc oublié jusqu'à ce qu'il recommence à bouger. Sans cet oubli, une scène de stationnement, par exemple, serait indéfiniment en mouvement dès que de nouveaux véhicules prendrait place. Étant donné cette situation, la mise à jour du modèle de l'arrière-plan devient fort utile même dans une scène intérieure.

Dans le cas de la méthode de soustraction de fond de base, la mise à jour se fait en incorporant une faible partie de l'intensité de l'image I_t au modèle :

$$B_{s,t+1} = (1 - \alpha)B_{s,t} + \alpha \cdot I_{s,t} \tag{1.2}$$

où α est une constante entre 0 et 1. Avec un $\alpha = 0.01$, par exemple, uniquement 1% de l'image en cours est incorporé dans le modèle de l'arrière-plan. Ainsi, le contenu de cette scène serait totalement oublié, ou plutôt remplacé, après 100 images. Comme il est souhaitable qu'un objet en mouvement soit incorporé dans le modèle après environ 4 minutes, $\alpha = \frac{1}{4 \cdot 60 \cdot 30} = 0.000139$ serait un choix adéquat si la vidéo a été capturée à 30 images par seconde. Avec un $\alpha = 1.0$, cette soustraction de fond devient une simple différence entre l'image en cours et la dernière image : $\chi_t = \|I_{t-1} - I_t\| > \tau$, ce qui donne généralement la bordure des objets en mouvement.

Notons que la mise à jour du modèle de l'arrière-plan peut s'avérer néfaste dans certains cas. Lorsqu'un objet en mouvement cesse de bouger, il est lentement appris par le modèle. Par contre, lorsqu'il recommence à bouger, le modèle peut avoir oublié le vrai arrière-plan et le détecte comme un objet en mouvement. C'est le problème du mouvement intermittent des objets présentés dans la section 1.5.1. Les figures 1.4 et 1.5 sont de bons exemples représentant cette difficulté. Une méthode aussi simple donne malheureusement de mauvais résultats dans un environnement réel.

Les autres modifications à apporter pour rendre la soustraction de fond plus robuste consistent à changer le modèle de l'arrière-plan B et la fonction de distance d. Nous verrons plusieurs exemples dans la section 1.6 où un état de l'art des méthodes de détection de changement sera présenté. Les difficultés usuelles sont définies avant cette section afin de permettre au lecteur de mieux comprendre les décisions des chercheurs.

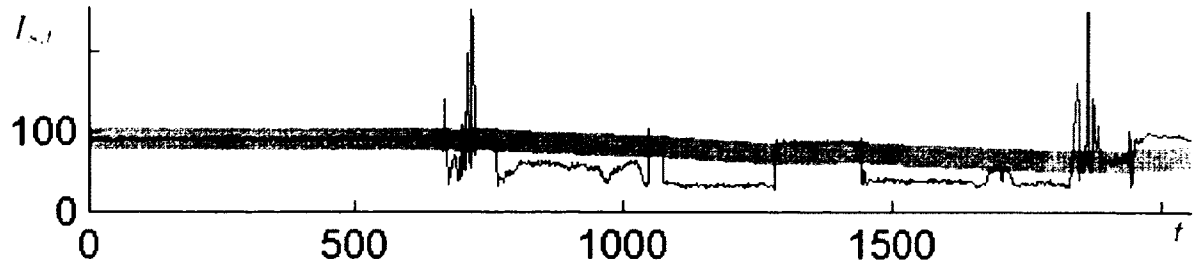


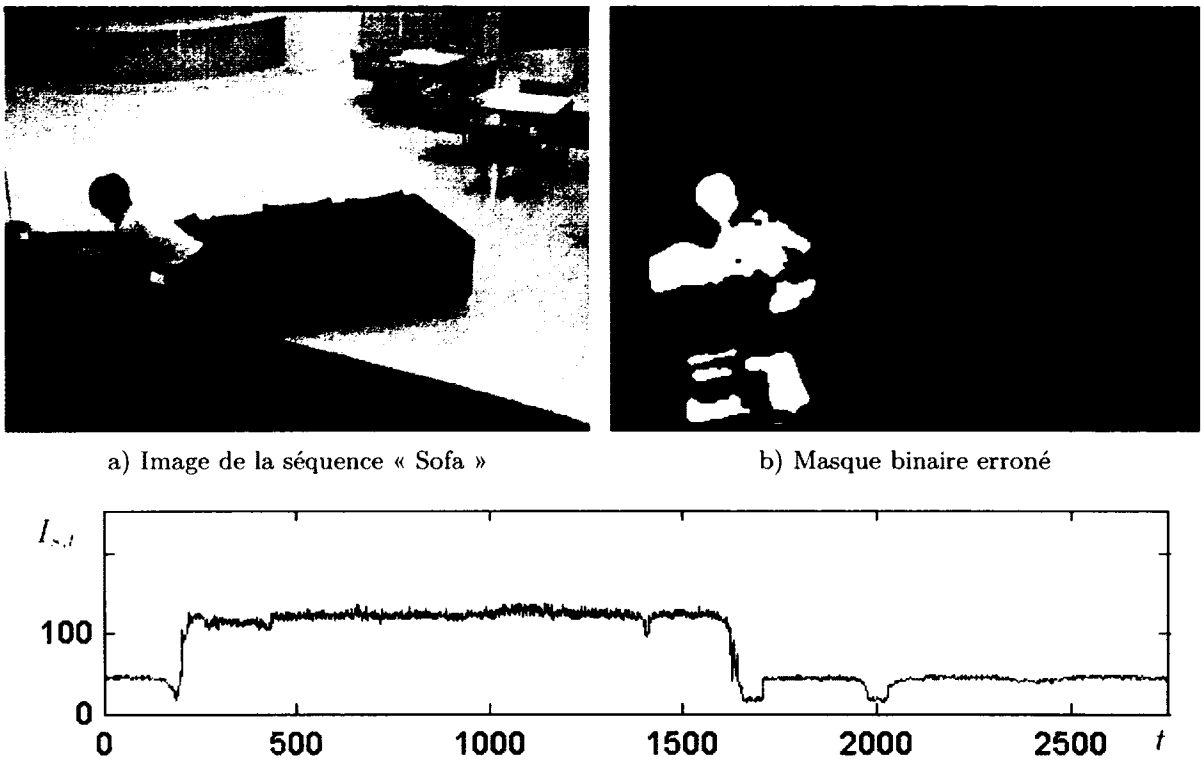
FIGURE 1.4 – En bleu, intensité (en niveaux de gris) d'un pixel de la séquence « Office » (figure 1.3) et, en vert, les intensités considérées comme faisant partie de l'arrière-plan étant donné une mise à jour $\alpha = 0.001$ et un seuil $\tau = 15$. Le modèle et les intensités acceptées par ce dernier sont dans la zone verte. On peut voir dans les plateaux vers les images 1350 et 2000 que l'adaptation du modèle aux nouvelles intensités cause des erreurs de segmentation.

1.5 Les difficultés standards

Nous avons identifié huit difficultés majeures et récurrentes en détection de changement. La définition particulière du changement est en partie responsable de celles-ci. Le paradigme en soustraction de fond est aussi responsable de certaines difficultés. Cette section énumère les problèmes usuels auxquels les méthodes font face.

1.5.1 Mouvement intermittent des objets

Lorsqu'un objet en mouvement s'arrête pendant quelques minutes, il est appris par le modèle et devient donc partie intégrante de l'arrière-plan. Par exemple, une voiture stationnée pour une longue durée, des piétons attendant longtemps sur le trottoir ou un objet quelconque déposé ; ils seraient tous appris par le modèle de l'arrière-plan. Si cet objet recommence à bouger, du mouvement est détecté à l'endroit où il était car l'intensité des pixels change abruptement. On appelle le « fantôme » ce type de faux positifs. On peut le constater dans la figure 1.5. Ce problème est une conséquence directe de la mise à jour de l'arrière-plan. Notons que cette dernière est très utile, mais elle peut avoir des inconvénients ; les fantômes en sont un bon exemple.



a) Image de la séquence « Sofa »

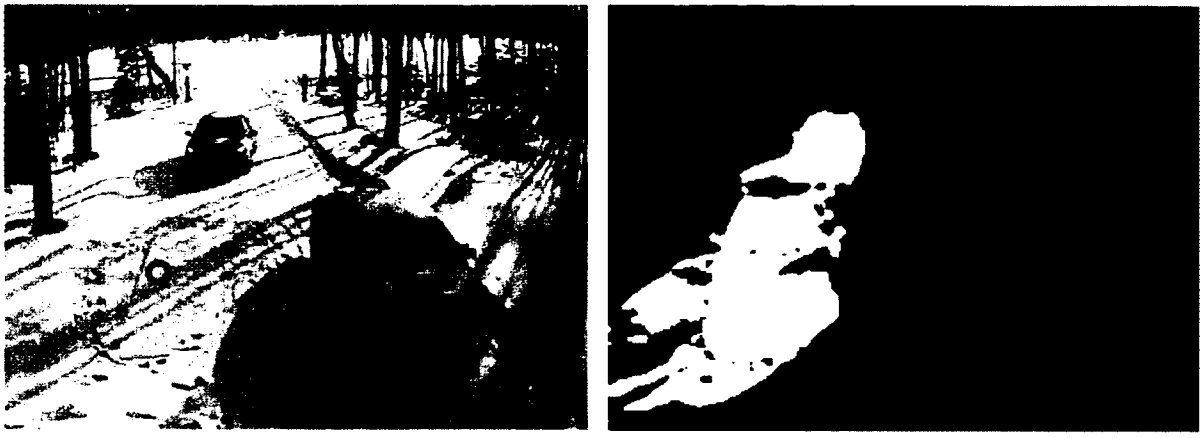
b) Masque binaire erroné

c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.5 – Une boîte est placée au sol vers l'image 200 pour être déplacée sur le divan vers l'image 1700, ce qui laisse croire à la plupart des méthodes qu'il y a deux objets en mouvement : la boîte et son fantôme.

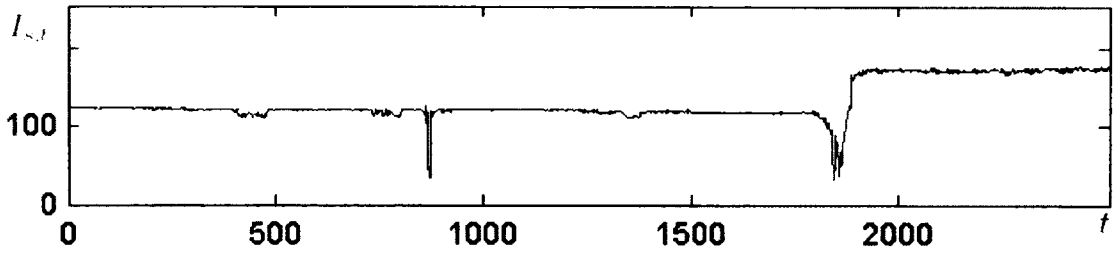
1.5.2 Arrière-plan inconnu

Dans certains lieux achalandés, il est fréquent que l'arrière-plan ne soit pas connu, par exemple sur une autoroute ou dans un aéroport. Ceci laisse peu de chances aux algorithmes de soustraction de fond d'apprendre l'arrière-plan. Ce problème n'est pas uniquement causé par une scène trop mouvementée, il peut aussi être causé par un objet qui n'a jamais bougé ou qui a cessé de bouger depuis très longtemps. La figure 1.6 illustre le problème. Étant donné qu'un grand nombre de méthodes donnent beaucoup de poids à la première image de la séquence vidéo pour apprendre le modèle, compenser un mauvais apprentissage peut nécessiter des centaines, voire des milliers d'images.



a) Image de la séquence « WinterDriveway »

b) Masque binaire erroné

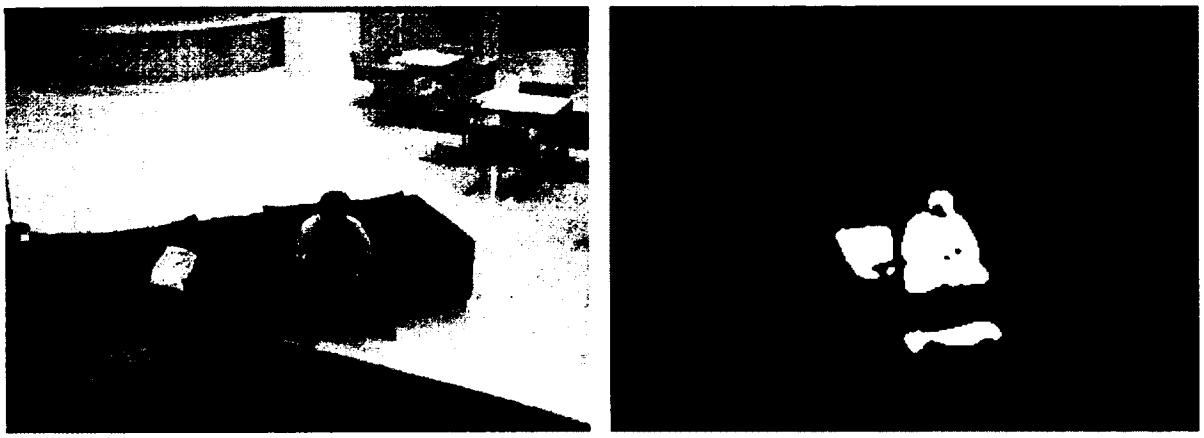


c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.6 – Lorsque l'automobile de gauche (et son ombre) démarre vers l'image 1850, elle laisse sa trace car le modèle n'a jamais eu l'occasion d'apprendre à quoi ressemble l'arrière-plan derrière ce véhicule. Il détecte alors le fantôme du réel objet en mouvement.

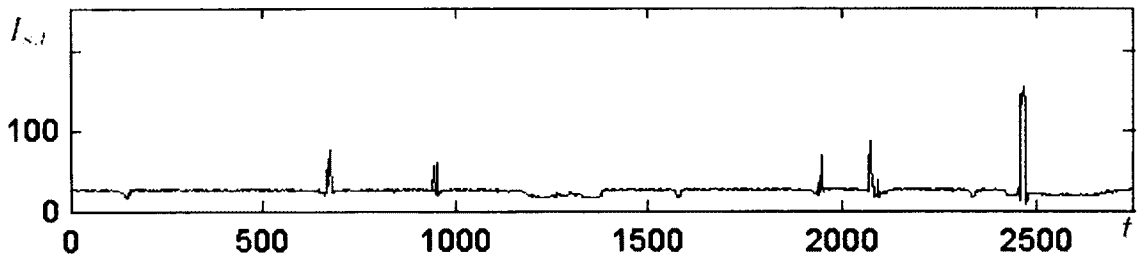
1.5.3 Camouflage

Problème très fréquent mais malheureusement ignoré par la plupart des méthodes, le camouflage survient lorsque la totalité ou une partie de l'objet en mouvement est de la même couleur que le modèle de l'arrière-plan. Étant de la même intensité ou presque, le modèle croit à tort que l'objet est l'arrière-plan. Dans la figure 1.7 par exemple, l'homme est en deux parties car la couleur de ses pantalons ressemble trop à celle du divan. Les méthodes de détection de changement uniquement basées sur les pixels sont particulièrement sensibles à ce phénomène. N'ayant aucune connaissance quant à la forme normale des objets, elles ne peuvent savoir qu'il est anormal qu'un objet soit coupé en deux, par exemple.



a) Image de la séquence « Sofa »

b) Masque binaire erroné

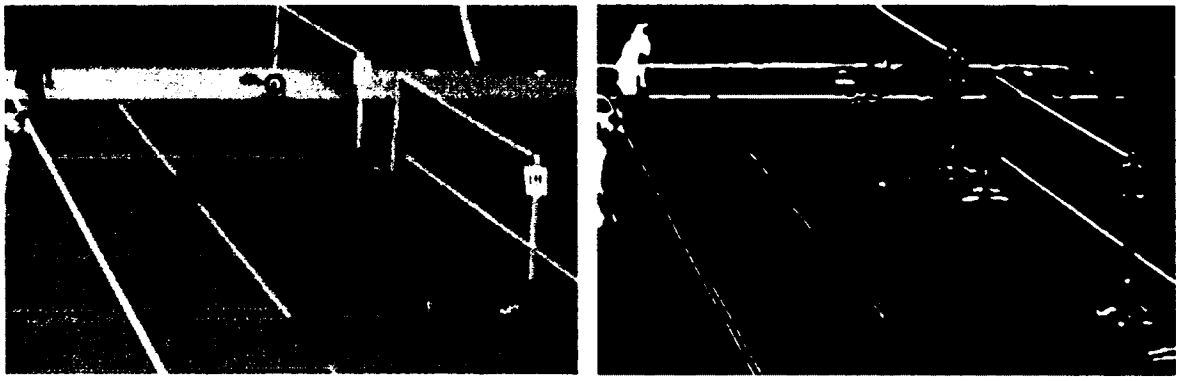


c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.7 – Constatant peu de différences sur l'intensité des pixels entre le divan et les jambes de l'homme, la méthode ne détecte pas de mouvement. La fluctuation est pourtant visible vers l'image 1200.

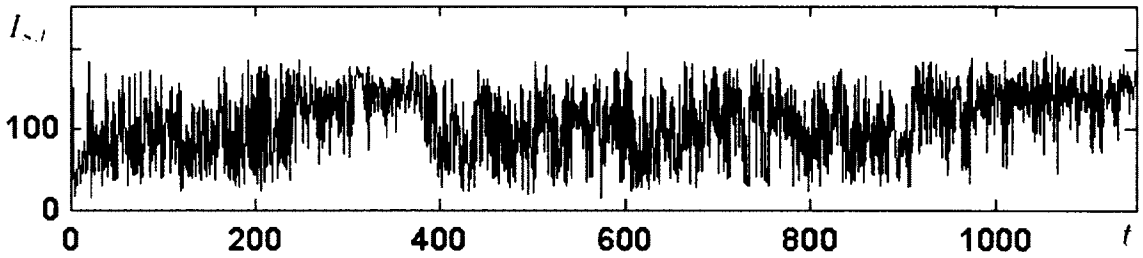
1.5.4 Mouvements de la caméra

Comme la plupart des méthodes détectent du mouvement lorsqu'il y a une variation importante sur l'intensité ou la couleur des pixels, il est important que la caméra soit parfaitement fixe. Si ce n'est pas le cas, les pixels sont soudainement déplacés dans une direction aléatoire, ce qui implique que du mouvement sera détecté partout où il y a des zones texturées et des bords. Le vent, la proximité d'un ventilateur, la position de la caméra (bateau, gratte-ciel) et des vibrations de tout genre peuvent faire bouger la caméra. Tel qu'illustré à la figure 1.8, les zones uniformes ne seront pas affectées car un déplacement aléatoire dans une direction aléatoire donne une intensité égale au reste de la zone.



a) Image de la séquence « Badminton »

b) Masque binaire erroné



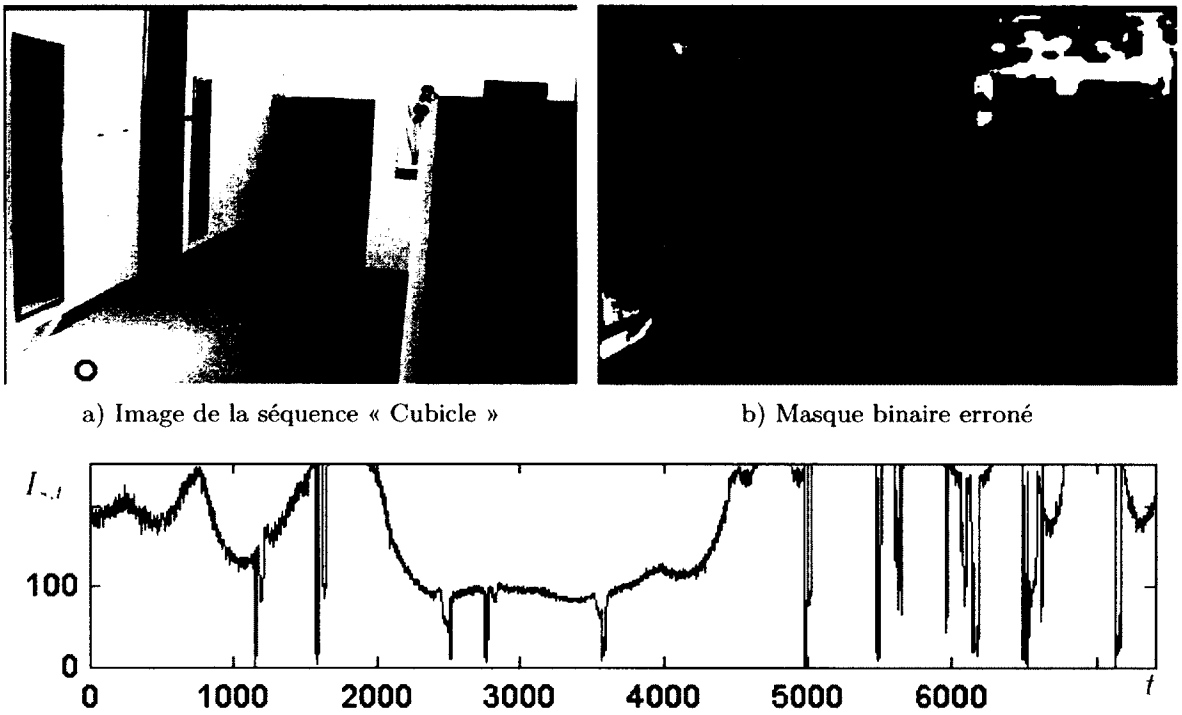
c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.8 – Une partie de badminton capturée avec un déplacement variant entre faible et moyen. Les réels objets en mouvement sont visibles, mais il y a beaucoup de faux positifs sur les lignes et les filets. De couleur uniforme, les zones vertes au sol ne sont pas affectées. Bien qu'il n'y ait aucun objet en mouvement dans le cercle rouge en a), le graphique en c) laisse croire qu'il y a continuellement du mouvement.

1.5.5 Changement d'illumination

Il y a plusieurs types de changement d'illumination. Lorsqu'une méthode traite une vidéo extérieure durant plusieurs heures, il y a un changement d'illumination graduel dû à la rotation de la terre. Cette illumination cause rarement des problèmes car le modèle des méthodes est généralement mis à jour plus rapidement que le changement d'illumination. Les illuminations soudaines, qu'elles soient globales ou locales, sont problématiques car le modèle n'a pas nécessairement le temps de s'adapter à cette nouvelle intensité. Il peut alors détecter de grandes zones comme étant en mouvement bien que rien n'ait bougé.

L'ouverture et la fermeture des lumières dans une pièce et le mouvement des nuages (voir figure 1.9) sont des causes fréquentes de changement d'illumination soudain.



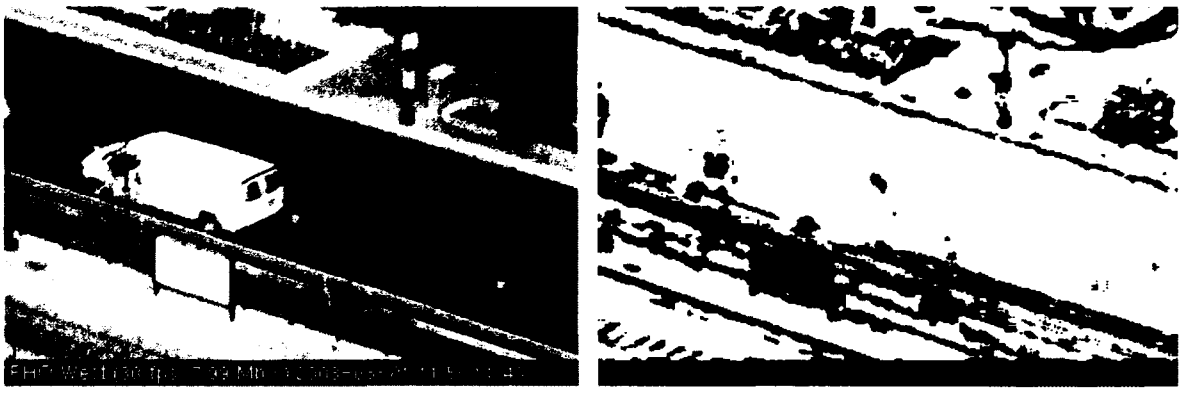
a) Image de la séquence « Cubicle »

b) Masque binaire erroné

c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.9 – L'illumination soudaine causée par le soleil modifie l'intensité de la majorité des pixels. Normalement, la modification est insuffisante pour détecter du mouvenent, mais dans le cas du carré sur le plancher et du mur, elle est amplement suffisante.

Une autre cause de changement d'illumination provient des caméras elles-mêmes. En effet, plusieurs d'entre elles possèdent un module d'ajustement automatique des blancs ; ceci crée un effet de changement d'illumination soudain. Lorsqu'un objet de grande taille passe dans la scène, la caméra ajuste les couleurs, ce qui est un problème en soustraction de fond car l'intensité de tous les pixels change en très peu de temps. On peut le constater avec la figure 1.10 où du mouvement est détecté presque uniformément alors qu'une camionnette traverse la scène. Les systèmes d'ajustement automatique de l'ouverture de la lentille peuvent aussi causer des variations importantes sur l'intensité et la couleur des pixels.



a) Image de la séquence « Boulevard »

b) Masque binaire erroné

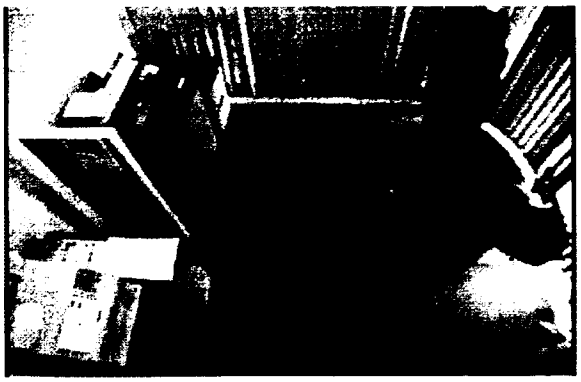
c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.10 – Une camionnette blanche traverse la scène vers l'image 850, causant un ajustement des blancs par la caméra, ce qui cause à son tour une détection erronée globalement sur la scène.

1.5.6 Ombres

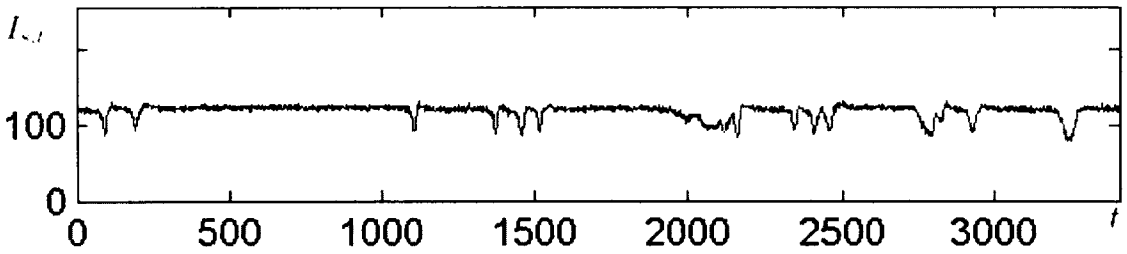
Un des critères pour que la soustraction de fond fonctionne parfaitement est que la scène soit bien éclairée et qu'il n'y ait pas d'ombre. La présence d'ombres déforme fortement l'objet détecté. Ce n'est rien de grave si le but est simplement de détecter du mouvement, mais c'est problématique dans un système plus complexe où la forme et la taille de l'objet sont importantes. Dans le masque binaire de la figure 1.11, on constate qu'il est beaucoup plus difficile de reconnaître un humain car sa forme ne respecte plus ce qu'on s'attend être la forme d'un humain. De plus, la zone en mouvement sur la photocopieuse pourrait être interprétée par erreur comme une troisième personne dans la pièce.

Il y a une différence entre les ombres douces et les ombres dures. Bien qu'il n'y ait



a) Image de la séquence « CopyMachine »

b) Masque binaire erroné



c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.11 – Deux personnes dans une salle de photocopie éclairée majoritairement par la droite. La plupart des méthodes se font piéger en détectant les ombres comme du mouvement. Malgré l'apparence du graphique en c), il n'y a aucun mouvement réel dans le cercle rouge en a).

aucun seuil clair pour déterminer si une ombre est douce ou dure, on peut généralement deviner son type en regardant le contraste avec les autres éléments de la scène. Si la surface ombragée est d'une intensité assez différente du reste de la surface, ce sera considéré comme de l'ombre dure. Une scène éclairée par plusieurs sources lumineuses a généralement moins d'ombres dures. Il existe quelques algorithmes pour détecter les ombres [14], mais ils ne fonctionnent généralement que sur les ombres douces.

Les ombres sont parfois projetées par des objets en mouvement, parfois par les arbres et les bâtiments. Dans le premier cas, les pixels deviennent nécessairement plus sombres que leur valeur usuelle dans le modèle. Dans l'autre cas, l'intensité des pixels peut autant

diminuer qu'augmenter car certaines zones ombragées peuvent ne plus l'être et certaines zones claires peuvent soudainement être ombragées. Cette nuance entre les deux causes d'ombre peut sembler mineure, mais des algorithmes pourraient utiliser ces connaissances afin d'améliorer leur robustesse face aux problèmes d'ombre.

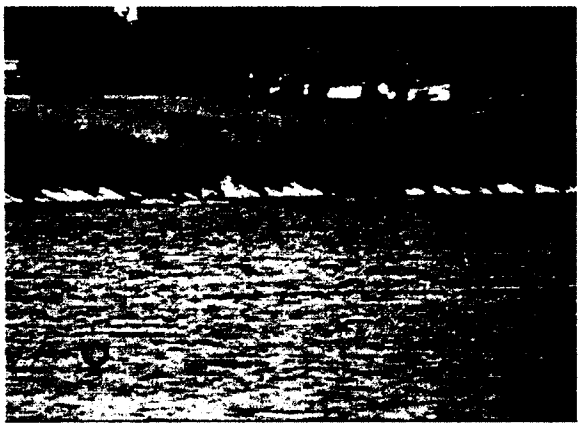
1.5.7 Arrière-plan dynamique

Problème longtemps étudié en soustraction de fond, l'arrière-plan dynamique représente, comme son nom l'indique, un mouvement local et relativement périodique visible dans l'arrière-plan. Ce problème est souvent causé par le vent dans les arbres ou dans un drapeau, le mouvement de l'eau, la rotation d'un ventilateur et les nuages. Ce problème est plus fréquent dans les scènes extérieures, mais on le retrouve aussi dans certaines scènes intérieures, par exemple avec du vent dans des rideaux. La figure 1.12 donne un bon exemple d'arrière-plan dynamique.

Tel qu'expliqué dans la section 1.3 sur la définition du mouvement, ce type de mouvement n'est pas intéressant et doit être ignoré. Ceci force les méthodes à modéliser un arrière-plan qui change sans cesse, malgré les hypothèses de base qui demandent un arrière-plan totalement fixe. Pour ajouter à la difficulté, la fréquence et l'intensité des mouvement peut varier amplement entre les vidéos. Certains mouvements ont une haute fréquence (vague, ventilateur, drapeau), d'autres ont une base fréquence (panneau publicitaire changeant toutes les 30 secondes). Idem pour l'amplitude qui peut être faible (ondulations sur un étang, vagues dans une rivière calme) ou élevée (vagues à Hawaii). Quels que soient leur fréquence et leur intensité, ces mouvements devraient toujours être classés dans l'arrière-plan car ils ne sont pas pertinents.

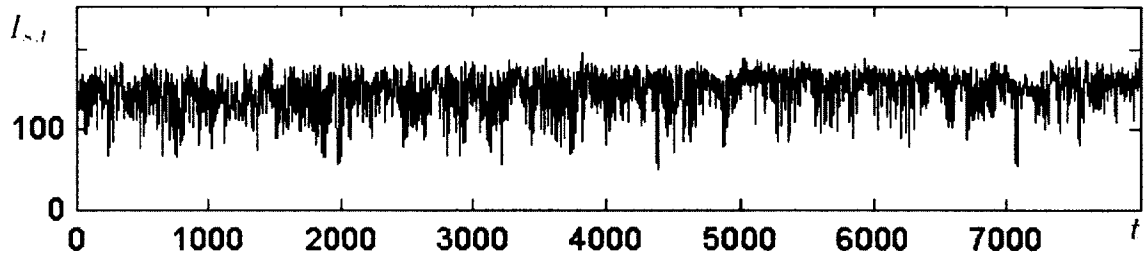
1.5.8 Bruit

Une vidéo capturée dans un environnement peu éclairé va souvent avoir des artéfacts de bruit causés par un ratio signal sur bruit faible. Le mauvais temps (pluie, neige, brouillard, tempête de sable) peut aussi être considéré comme une source de bruit car il introduit des modifications d'apparence aléatoire dans la vidéo. Quel que soit le type de bruit, une



a) Image de la séquence « Boats »

b) Masque binaire erroné



c) Intensité du pixel (en niveaux de gris) au centre du cercle rouge de l'image a)

FIGURE 1.12 – Le mouvement de l'eau et des arbres dans cette scène fait croire à plusieurs méthodes qu'il y a des mouvements importants. Les résultats de cette méthode sont plutôt bons, mais elle se fait tout de même piéger par le mouvement de l'eau. Même s'il n'y a aucun objet en mouvement dans le cercle rouge en a), le graphique en c) laisse croire qu'il y en a continuellement.

modification importante sur l'intensité des pixels va certainement donner des résultats erronés. Cela dit, ce problème peut facilement être réglé avec des filtres morphologiques, mais un bruit trop important va tout de même introduire de fausses valeurs dans les données utilisées par les méthodes.

1.5.9 Autres difficultés

Il existe de nombreuses autres difficultés, mais elles sont moins fréquentes que celles présentées dans les dernières sections. Les effets de réflexion dans les vitres, sur l'eau et dans les séquences thermiques, la perte d'images due aux troubles de communication, le manque de constraste (vision de nuit, brouillard, etc.), les artéfacts de compression JPEG, l'auto-focus, les caméras PTZ [3], l'effet de moment des gratte-ciel dans les séquences UAV [4] causent tous des problèmes au niveau de la détection de changement.

1.6 Méthodes de segmentation

Les méthodes de détection de changement sont diversifiées. Cette section présente un état de l'art de ces dernières et explique dans certains cas les problèmes qui s'y rattachent. Bien que les méthodes se regroupent généralement par famille, une première séparation sera faite entre les méthodes paramétriques, semi-paramétriques et non paramétriques, et ensuite par famille.

1.6.1 Modèle de l'arrière-plan

Nous avons vu en détails dans la section 1.4 la méthode de base en soustraction de fond. Nous avons aussi vu qu'il est possible d'améliorer la robustesse de cette méthode en modifiant les différents éléments de l'équation 1.1. Ce qui différencie les méthodes est principalement leur façon de modéliser l'arrière-plan B. Elles se distinguent également par le type de caractéristiques utilisées et leur facon de détecter du mouvement (fonction de distance d et seuil τ). Les prochains paragraphes porteront sur ces deux derniers points et sur les 4 grandes familles de méthodes, soient paramétrique, semi-paramétrique, non paramétrique et autres.

3. *Point, Tilt, Zoom*, Caméra manipulable à distance
4. *Unmanned Aerial Vehicle*, Drone

Caractéristiques

Les méthodes de détection de changement ne se basent pas toutes sur les mêmes informations pour estimer un masque binaire. Avant de comprendre leur fonctionnement, il est important de connaître les caractéristiques de départ.

Les premières séquences vidéos disponibles étaient enregistrées en niveaux de gris alors, en règle générale, les vieilles méthodes se basent sur une seule valeur par pixel. L'image devient alors une matrice 2D contenant des valeurs entre 0 (noir) et 255 (blanc). Mathématiquement, $I_{t,s} \in [0, 255]$ est une image au temps t constituée de pixels monochromes. L'évolution rapide de la technologie favorisa l'avènement des caméras couleur. Les images deviennent alors des matrices 2D avec 3 canaux par élément. Mathématiquement, $I_{t,s} \in [(0, 0, 0), (255, 255, 255)]$ est une image au temps t constituée de pixels couleur RGB. Ces séquences couleur offrent maintenant 2 dimensions de plus, ce qui permet généralement d'améliorer les résultats de la segmentation.

Certains auteurs utilisent les bordures des objets [15, 16] ou de la scène [17] pour améliorer la robustesse de leurs méthodes. On peut voir les résultats d'un algorithme de détection de contours à la figure 1.13. Il est aisé et rapide d'obtenir les bordures des objets avec un détecteur de contours ou avec la différence entre deux images consécutives [18].

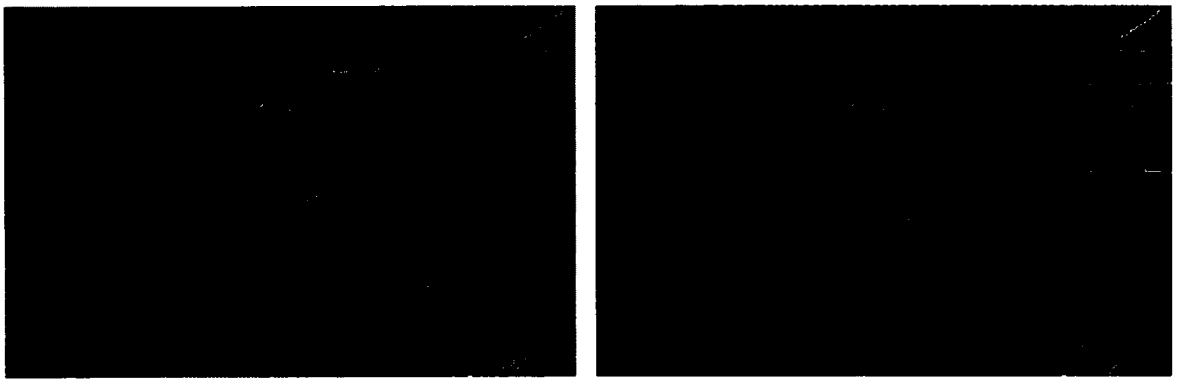


FIGURE 1.13 – Détection de contour à l'aide de l'algorithme de Canny selon deux seuils. Certaines méthodes de détection de changement se basent sur cette information. Image de la séquence « Office ».

Le flux optique peut être utilisé pour produire un masque binaire. Le but est de calculer le déplacement de différentes zones entre I_t et I_{t+1}. Les travaux du psychologue J.J. Gibson sur la perception de la profondeur ont grandement influencé les recherches modernes sur la perception et ont permis de mieux comprendre les fondements du flux optique. Comme on peut le voir à la figure 1.14, cette méthode est souvent illustrée par un champ de vecteurs. Ces derniers sont une projection 2D des réels vecteurs 3D générés à partir du flux entre les deux images. Si $I(x, y, t)$ est la fonction d'intensité de l'image pour le pixel (x, y) au temps t, alors : $I(x, y, t) \approx I(x + \delta x, y + \delta y, t + 1)$. Pour chaque pixel, le vecteur de déplacement entre I_t et I_{t+1} est $w = (\delta x, \delta y, 1)^T$. Les méthodes pour calculer ces vecteurs sont nombreuses et dépassent la portée de ce document. Précisons simplement que le flux optique est un paradigme totalement différent de la soustraction de fond et qu'il apporte son lot d'avantages et d'inconvénients.

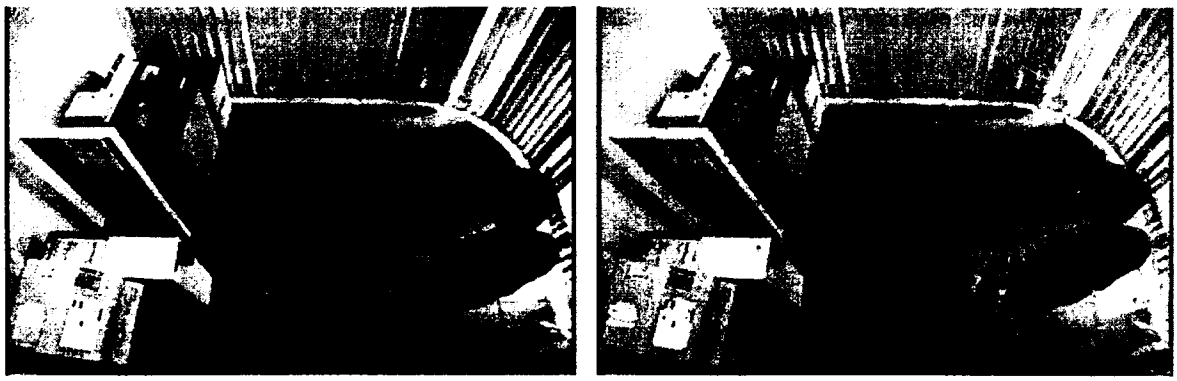


FIGURE 1.14 – Résultat d'un calcul de flux optique selon l'algorithme itératif de Lucas-Kanade. Les points verts représentent des coins saillants dans l'image de gauche et les lignes vertes, le déplacement de ces points dans l'image de droite. Images de la séquence « CopyMachine ».

L'analyse en composantes principales (ACP, *PCA* en anglais) est une technique statistique pour réduire la dimensionalité d'un ensemble d'observations. Elle transforme les observations en vecteurs non corrélés orthogonaux. Il est ensuite possible de retrouver les données de départ à partir de ces vecteurs. Un arrière-plan peut donc être modélisé par un ensemble d'ACP et comparé aux nouvelles images [19, 20].

Méthodes paramétriques

Les méthodes paramétriques supposent que la couleur ou l'intensité de chaque pixel suit une loi particulière (le plus souvent gaussienne) dont les paramètres peuvent être déterminées par entraînement. Notons que certains auteurs doutent de la pertinence statistique d'un modèle gaussien car les images naturelles montrent parfois des statistiques non gaussiennes [21].

La méthode la plus simple pour modéliser un arrière-plan est d'utiliser une image de la scène sans objet en mouvement. La distance entre les images et ce modèle peut ensuite être calculée et seuillée afin d'obtenir un masque binaire. En général, ce type de modèle présuppose que la couleur (ou le niveau de gris) de chaque pixel suit une distribution de type porte centrée sur une valeur donnée. Capturer une image sans mouvement peut être complexe, mais le modèle peut aussi être estimé à l'aide de filtres médians temporels fixes [22, 16, 23] ou adaptatifs [24, 25] ou par des histogrammes [26]. Il est aussi possible d'utiliser l'image précédente $t-1$ comme modèle B [27], mais ceci permet uniquement de voir des petites parties des objets en mouvement. La fonction de distance est choisie parmi une des quatre normes standards, soient :

$$\begin{aligned} d_0 &= |I_{t,s} - B_{t,s}| \\ d_1 &= |I^R_{t,s} - B^R_{t,s}| + |I^V_{t,s} - B^V_{t,s}| + |I^B_{t,s} - B^B_{t,s}| \\ d_2 &= (I^R_{t,s} - B^R_{t,s})^2 + (I^V_{t,s} - B^V_{t,s})^2 + (I^B_{t,s} - B^B_{t,s})^2 \\ d_\infty &= \max\{|I^R_{t,s} - B^R_{t,s}|, |I^V_{t,s} - B^V_{t,s}|, |I^B_{t,s} - B^B_{t,s}|\} \end{aligned}$$

où R, V et B sont les canaux rouge, vert et bleu, et d_0 fonctionne seulement sur les images monochromes. La figure 1.4 présente un exemple de segmentation avec une vidéo en niveaux de gris. Dans le cas où le modèle contient des objets en mouvement, des objets fantômes seront visibles durant quelques minutes et seront ensuite absorbés par le modèle (voir section 1.6.3).

Modéliser l'arrière-plan avec une image unique comme dans la méthode de base demande un arrière-plan particulièrement fixe sans bruit ni artéfact. Comme la plupart des vidéos ne respectent pas ces critères, de nombreux auteurs modélisent les pixels comme une densité de probabilité [14, 28]. La segmentation équivaut alors à calculer la probabilité d'un pixel

d'appartenir à l'arrière-plan ; un pixel avec une basse probabilité devrait appartenir à un objet en mouvement. Wren *et al.* [29] modélisent chaque pixel comme une distribution gaussienne $\mathcal{N}(\mu_{t,s}, \Sigma_{t,s})$, où $\mu_{t,s}$ et $\Sigma_{t,s}$ sont la couleur moyenne et la matrice de covariance du pixel s au temps t. On peut ensuite calculer la distance avec la vraisemblance logarithmique :

$$d_G = \frac{1}{2}\log[(2\pi)^3|\Sigma_{t,s}|] + \frac{1}{2}(I_{t,s} - \mu_{t,s})\Sigma_{t,s}^{-1}(I_{t,s} - \mu_{t,s})^T$$

ou la distance de Mahalonabis :

$$d_M = |I_{t,s} - \mu_{t,s}|\Sigma_{t,s}^{-1}|I_{t,s} - \mu_{t,s}|^T$$

Cette famille de méthodes est beaucoup plus flexible car chaque pixel a un seuil unique qui s'adapte au niveau de bruit et de perturbations qui lui est associé.

Méthode semi-paramétriques

Les méthodes semi-paramétriques émettent également l'hypothèse que la couleur d'un pixel d'arrière-plan suit une densité de probabilités dont les paramètres peuvent être appris, mais certains éléments sont ajoutés aux algorithmes afin qu'ils soient plus robustes face à certaines difficultés.

Pour prendre en compte les arrière-plans dynamiques souvent rencontrés à l'extérieur, tels que les feuilles d'arbres et les vagues sur l'eau, certains auteurs proposent de modéliser chaque pixel par un mélange de gaussiennes [30, 31, 32]. Selon le modèle de Stauffer et Grimson [33], la probabilité d'occurence d'une couleur pour un pixel s est donnée par :

$$P(I_{t,s}) = \sum_{i=1}^{K} \omega_{i,t,s}\mathcal{N}(\mu_{i,t,s}, \Sigma_{i,t,s})$$

où k est le nombre de gaussiennes, $\mathcal{N}(\mu_{i,t,s}, \Sigma_{i,t,s})$ est le i-ème modèle gaussien et $\omega_{i,t,s}$ son poids. La moyenne, l'écart type et le poids de la gaussienne la plus près de $I_{t,s}$ sont mis à jour. Lorsqu'aucune gaussienne ne correspond à cette couleur, la gaussienne avec le poids le plus faible est remplacée par une nouvelle ayant une moyenne centrée à $I_{t,s}$, un large écart type et un poids faible. Un pixel est classifié en mouvement s'il n'est pas à l'intérieur de 2.5 écart type des gaussiennes ayant un poids suffisant.

Une autre approche pour prendre en compte les arrière-plans dynamiques est la méthode *Codebook* de Kim *et al.* [34]. Durant la phase d'entraînement, une ou plusieurs valeurs (*code words*) sont assignées à chaque pixel afin de représenter l'ensemble des valeurs que peut prendre un pixel. Ainsi, un pixel dans une zone stable pourrait être décrit avec seulement une valeur. Un pixel dans une zone agitée, par exemple un arbre dans le vent, pourrait être décrit par trois valeurs : le vert des feuilles, le bleu du ciel et le brun de l'écorce. Un pixel est classifié statique s'il respecte l'équation de distortion des couleurs :

$$\left[I_{t,s}^{R^2} + I_{t,s}^{V^2} + I_{t,s}^{B^2} - \frac{\mu_{t,s}^{R} \cdot I_{t,s}^{R} + \mu_{t,s}^{V} \cdot I_{t,s}^{V} + \mu_{t,s}^{B} \cdot I_{t,s}^{B}}{\mu_{t,s}^{R^2} + \mu_{t,s}^{V^2} + \mu_{t,s}^{B^2}} < \tau \right]$$

et l'équation de distortion d'illumination :

$$\alpha_{t,s} \leq I_{t,s}^{R^2} + I_{t,s}^{V^2} + I_{t,s}^{B^2} \leq \beta_{t,s}$$

où $\mu_{t,s}^{R}$, $\mu_{t,s}^{V}$, $\mu_{t,s}^{B}$, $\alpha_{t,s}$ et $\beta_{t,s}$ sont des paramètres de la i-ème valeur d'un pixel s, et τ est un seuil prédéfini par l'utilisateur.

Méthodes non paramétriques

Il y a deux familles de méthodes non-paramétriques :

1. celles qui ne présupposent pas que la couleur des pixels d'arrière-plan suit une distribution de probabilités et
2. celles qui supposent que la distribution de probabilités des pixels d'arrière-plan dépend des données observées et non de paramètres pouvant être appris.

Les méthodes non-paramétriques n'émettent donc aucune hypothèse quant à la forme des données.

L'estimation par noyau, ou KDE (*Kernel Density Estimator*), est une méthode non-paramétrique d'estimation de la densité de probabilités d'une variable aléatoire. Elle a été appliquée à la détection de changement entre autres par Elgammal *et al.* [35].

$$P(I_{t,s}) = \frac{1}{N} \sum_{i=t-N}^{t-1} K(I_{t,s} - I_{i,s})$$

où K est le noyau (généralement une gaussienne) et N le nombre d'images utilisées pour estimer $P(.)$. Pour une séquence vidéo couleur RGB, un produit de noyaux peut être utilisé :

$$P(I_{t,s}) = \frac{1}{N} \sum_{i=t-N}^{t-1} \prod_{j=R,V,B} K\left(\frac{I^j_{t,s} - I^j_{i,s}}{\sigma_j}\right)$$

Un pixel est classifié statique s'il est improbable qu'il fasse partie de cette distribution, donc lorsque $P(I_{t,s})$ est plus petit qu'un seuil prédéfini.

Zivkovic et van der Heijden [32] publient une méthode adaptative d'estimation par noyau où la taille K du noyau varie selon la densité. Le volume du ballon sera alors plus grand dans les zones de faible densité et plus petit dans les zones de forte densité. Cette technique, couramment appelée le « *Balloon estimator* », est une généralisation de l'algorithme des k plus proches voisins où un objet obtient la même classification que celle de ses voisins.

Certains auteurs ont tenté d'estimer les paramètres de l'arrière-plan avec une modulation delta-sigma ($\Sigma\Delta$), une technique très fréquente en conversion analogue vers numérique [36, 37]. L'algorithme $\Sigma\Delta$, populaire dans les systèmes embarqués [38], consiste à approximer récursivement l'arrière-plan avec de simples incrémentations et décrémentations au niveau des pixels lorsque la différence entre le modèle et l'image en cours est trop grande.

Certains auteurs utilisent l'approche par consensus pour déterminer si un pixel appartient à l'arrière-plan [39, 40]. Ils gardent les N dernières valeurs observées pour chaque pixel et classifient un pixel statique lorsque sa valeur concorde avec la majorité des valeurs enregistrées.

Barnich et van Droogenbroeck [41] préfèrent voir la détection de changement comme un problème de classification ; ils croient qu'un pixel devrait être classifié selon son voisinage et non selon un modèle. Ils modélisent chaque pixel comme un ensemble d'échantillons pris aléatoirement dans le voisinage. Un pixel doit alors ressembler à quelques échantillons pour être classifié statique. Une telle définition implique que l'ensemble d'échantillons doit contenir uniquement des pixels de l'arrière-plan ; les pixels des objets en mouvement ne sont donc jamais ajoutés au modèle. Les échantillons sont remplacés selon une loi uniforme, de sorte qu'ils ont davantage de chances de l'être s'ils sont vieux. Cette méthode est l'une des rares offrant une robustesse en regard des artéfacts de fantômes.

Yin *et al.* [42] proposent une méthode à deux niveaux basée sur un classificateur. Les images sont divisées en bloc, puis un classificateur (SVN ou GC-Boost) détermine si ce bloc fait partie de l'arrière-plan. Le résultat est ensuite utilisé afin de garder le modèle de l'arrière-plan à jour. Maddalena et Petrosino utilisent aussi un classificateur avec leurs algorithmes SOBS [43, 44], où l'arrière-plan apprend le modèle des mouvements à l'aide de réseaux de neurones artificiels.

Zhu *et al.* [17] proposent une méthode basée sur des descripteur de type SIFT (*Scale-Invariant Feature Transform* en anglais). L'arrière-plan est modélisé selon une partie des coins visibles dans les images. Ces coins sont ensuite retrouvés par l'algorithme de flux optique de Lucas-Kanade. Lorsqu'un nouveau coin apparaît dans une zone, son descripteur est comparé aux autres présents dans cette région et il est classifié en mouvement lorsqu'il est assez différent. Une méthode de ce type est robuste face aux arrière-plans dynamiques et aux changements d'illumination.

Une autre approche pour s'adapter localement aux perturbations est la méthode W^4 d'Haritoaglu *et al.* [45]. Suite à la phase d'entraînement, chaque pixel s est représenté par un minimum m_s, un maximum M_s et un maximum de différence entre les images consécutives D_s. Les pixels sont classifiés statiques lorsque les critères suivants sont respectés :

$$|M_s - I_{t,s}| < \tau d_\mu \qquad \text{ou} \qquad |m_s - I_{t,s}| < \tau d_\mu$$

où τ est un seuil défini par l'utilisateur et d_μ est la médiane des différences entre les images. Ceci permet de prendre en compte une plus grande variation pour les pixels qui le requièrent. Par contre, une telle définition implique qu'il ne doit y avoir aucun objet en mouvement durant la phase d'entraînement.

Autres

Une approche non basée sur les pixels proposée par Oliver *et al.* [19] modélise l'arrière-plan en utilisant un espace propre. Une matrice de covariance Σ est calculée à l'aide d'une représentation par colonne des images d'une séquence. Une analyse par composante principale est ensuite appliquée pour obtenir ϕ_M, une matrice composée des M vecteurs propres ayant les plus grandes valeurs propres. Le modèle de l'arrière-plan est alors calculé :

$B_t = \phi_M(I_t - \mu)$, où μ est la moyenne de l'intensité des pixels. Il peut ensuite être reconstruit ainsi : $I_t^{'} = \phi_M^T B_t + \mu$, puis comparé aux nouvelles images à l'aide d'une distance euclidienne et d'un seuil. Ce type de modèle permet d'avoir une définition plus globale de l'arrière-plan car il prend en compte les pixels voisins.

Seki *et al.* [46] utilisent l'analyse en composantes principales (ACP) pour modéliser l'arrière-plan. Ils séparent les images en blocs de NxN pixels et utilisent d'anciens échantillons pour entraîner une ACP pour chaque bloc. Un bloc est classifié statique lorsque le nouveau bloc observé est assez près de la reconstruction utilisant les coefficients de projection de l'ACP. Un modèle de ce type est plus robuste en regard des variations de luminosité et aux mouvements parasites.

Une approche similaire, l'analyse par composantes indépendantes (ACI, *ICA*), est utilisée par Tsai et Lai [47]. $\underline{x} = (x_1, x_2, ..., x_p)$ est un vecteur aléatoire, l'ACI revient à identifier le modèle $\underline{x} = A\underline{s} + \sigma$ où les composantes s_i du vecteur $\underline{s} = (s_1, s_2, ..., s_p)$ sont mutuellement indépendantes, la matrice A est de taille pxn et σ est le bruit. Des images de la phase d'apprentissage sont utilisées pour construire le modèle ACI, puis le vecteur $\underline{x}$ est comparé aux nouvelles images observées. Ce modèle est robuste face aux changements d'illumination.

Il est possible d'utiliser les filtres de prédiction de type Wiener ou Kalman pour prédire la couleur que prendra chaque pixel. Un article de Ridder *et al.* [48] en 1995 propose de modéliser chaque pixel avec un filtre de Kalman dans le but de prendre en compte les grandes variations de certains pixels, causées par les arrière-plans dynamiques et par le bruit. Un pixel est classifié en mouvement lorsque la prédiction est trop loin de sa valeur réelle. Zhong et Sclaroff [49] proposent aussi d'utiliser un filtre de Kalman, mais en incorporant la corrélation entre les pixels dans leurs calculs.

Certains auteurs ont tenté d'exploiter le domaine spectral pour créer des méthodes de détection de changement. La transformée en cosinus discrète (TCD, *DCT* en anglais) est préférée à la transformée de Fourier car elle est purement réelle, elle requiert la moitié de mémoire et regroupe mieux l'énergie. Porikli et Wren [50] gardent un historique de N images pour modéliser chaque pixel avec une TCD. Le modèle de l'arrière-plan est construit avec N images au début de la vidéo. Une TCD est recalculée à chaque temps t et les coefficients sont comparés à ceux du modèle. Une technique de ce type est réputée

être robuste en regard des arrière-plans dynamiques. Pour profiter de l'encodage en cosinus discret offert par les standards jpeg et mpeg, certains chercheurs utilisent le domaine fréquentiel. Ceci leur permet d'éviter la transformée inverse vers le domaine spatial, une opération assez coûteuse. Wang *et al.* [51], entre autres, revisitent trois méthodes simples de soustraction de fond dans le domaine fréquentiel. Une première partie de leur algorithme classifie les blocs en mouvement lorsque la distance entre leurs coefficients et ceux de l'image en cours est assez grande. Lorsque c'est le cas, la transformée inverse est calculée uniquement sur ces blocs afin de calculer la segmentation au niveau des pixels.

1.6.2 Modèle des objets

Dans la majorité des cas, les objets ne sont tout simplement pas modélisés. La plupart des méthodes décrites dans la littérature travaillent sur chaque pixel indépendamment. Ces méthodes dépendent alors du post-traitement pour donner une consistance aux objets dans leurs masques binaires [41].

Certains systèmes complets de vidéosurveillance améliorent les résultats de la segmentation en utilisant les résultats du système de suivi d'objets [52]. Sachant qu'un objet devrait s'y trouver, il est possible de diminuer les probabilités ou le seuil τ afin d'augmenter les chances qu'un pixel soit statique dans cette zone.

Les auteurs de [53, 54, 55] modélisent à la fois les couleurs de l'arrière-plan et celles des objets en mouvement. Un mélange de gaussiennes est utilisé pour chaque pixel de l'arrière-plan, mais uniquement un modèle est partagé entre tous les objets car ils couvrent plusieurs pixels. Withagen *et al.* [53] et Landabaso *et al.* [55] modélisent les objets avec une distribution uniforme alors que Lindstrom *et al.* [54] modélisent les objets avec une mélange de gaussiennes. Une fois le modèle construit, ils proposent de comparer les probabilités qu'un pixel appartienne à l'arrière-plan via un test d'hypothèse par ratio de vraisemblance.

1.6.3 Mise à jour du modèle

Tel que spécifié dans la section 1.4, les objets stables depuis quelques minutes doivent être appris par le modèle de l'arrière-plan. Pour cette raison, même sans modification

parasitaire d'intensité, le modèle de l'arrière-plan devrait tout de même être mis à jour.

Les techniques pour mettre à jour le modèle de l'arrière-plan diffèrent selon les méthodes de détection de changement. De par leur nature, certaines méthodes effectuent la mise à jour automatiquement ; c'est le cas, par exemple, du filtre de Kalman [48] et de KDE [35]. D'autres doivent ajouter une étape à l'algorithme. Nous avons vu dans l'équation 1.2 que la mise à jour de la méthode de soustraction de fond de base se fait ainsi :

$$B_{s,t+1} = (1 - \alpha)B_{s,t} + \alpha \cdot I_{s,t}$$

où α est une constante entre 0 et 1. Le concept est généralement le même : une petite partie de la nouvelle image I_t est incorporée au modèle B_t. Dans certains cas, la mise à jour est conditionnelle aux résultats de la segmentation. Dans d'autres, elle peut être effectuée plus rapidement lors d'une illumination globale, par exemple.

1.6.4 Intégration spatiale

Certains auteurs utilisent le voisinage des pixels pour améliorer la segmentation [41]. D'autres utilisent les résultats des masques binaires précédents pour parfaire l'image en cours [56]. Intégrer la dimension spatiale et temporelle peut effectivement améliorer les résultats [57, 58, 59].

Modèles Markoviens

On suppose généralement que les objets en mouvement auront une forme compacte et des bordures lisses et que les petits objets isolés sont attribuables à des erreurs de segmentation. Une technique typique pour améliorer la segmentation est d'imposer un lissage au masque binaire. Ceci fait disparaître la plupart des petites régions erronées et donne de la consistance aux objets. Il est possible d'utiliser le modèle et la fonction de distance d'une méthode pour améliorer la segmentation après qu'elle ait été calculée. Les méthodes de cette section tentent généralement de maximiser la cohérence de la segmentation en utilisant le voisinage des pixels et les résultats des dernières images.

Migdal et Grimson [56] utilisent trois champs aléatoires de Markov (*MRF* en anglais) pour améliorer la segmentation. Le premier prend en compte la relation entre les pixels. Le deuxième, la relation temporelle en regardant les segmentations précédentes. Le dernier prend en charge la relation temporelle et les segmentations futures ; il ne peut donc pas être utilisé en temps réel. La segmentation est obtenue en estimant le *maximum a posteriori* (MAP) des MRF à l'aide de l'échantillonneur de Gibbs.

Cheng *et al.* [60] s'assurent d'une cohérence spatiale en utilisant l'algorithme de *Graph cut* [61]. Chaque pixel est connecté à la source et au drain avec un poids (énergie) dépendant de la fonction de distance utilisée lors de la segmentation. Plus il est probable qu'un pixel soit statique, plus son lien vers la source sera faible et plus son lien vers le drain sera fort. Des liens sont aussi créés entre les pixels voisins. Leur poids est déterminé par la distance entre l'intensité des deux pixels. Plus un lien est faible, plus il est coûteux de classifier différemment deux pixels voisins. *Graph cut* calcule ensuite la coupe minimale pour traverser le graphe ainsi créé.

Post-traitement

Il est possible d'améliorer la segmentation sans aucune information sur la méthode de détection de changement utilisée. Les filtres et les opérateurs morphologiques sont des outils peu coûteux en temps de calcul qui peuvent améliorer considérablement les résultats de la segmentation.

Parks et Fels [57] testent la fermeture (dilatation suivie d'une érosion) et la suppression des petits objets en utilisant divers paramètres. Benezeth *et al.* [58] testent le filtre médian 5x5, différentes combinaisons d'opérateurs morphologiques et un *a priori* markovien. Ils concluent que les trois techniques produisent des résultats équivalents, mais que les filtres et les opérateurs morphologiques sont à privilégier pour des raisons de temps de calcul. Brutzer *et al.* [59] effectuent sensiblement les mêmes tests et concluent que l'utilisation des algorithmes de post-traitement réduit l'écart entre les méthodes de détection de changement et qu'elles sont toutes améliorées par le post-traitement.

1.7 Problématique

La recherche en détection de changement étant toujours active, il est important que les chercheurs aient des outils adéquats pour valider leurs travaux. Un problème survient lorsqu'il est temps de comparer leur méthode avec des méthodes concurrentes. Il n'y a présentement pas de banque de données acceptée par la communauté scientifique. N'ayant pas de données, les auteurs choisissent les séquences qu'ils désirent dans les banques de données existantes (voir chapitre 2) ou ils utilisent des séquences qui leur sont propres. Ce n'est toutefois pas tant la production des séquences que l'annotation (la création des réalités de terrain) de ces dernières qui demande plusieurs jours de travail. C'est ce travail que personne ne veut faire car il demande beaucoup de ressources et de temps. Ces difficultés font que presque personne ne veut créer de séquences et que lorsque certains le font, c'est uniquement pour une partie des séquences ou pour une minorité des images. De plus, lors de la création de ces nouvelles séquences, les chercheurs ont tendance à choisir des séquences dont les difficultés à relever les avantagent par rapport aux autres méthodes. C'est une réaction logique de la part des chercheurs, mais qui mène rapidement vers de mauvaises comparaisons.

Une fois les séquences choisies, les chercheurs sélectionnent une ou plusieurs méthodes avec lesquelles se comparer. Voulant obtenir les résultats sur ces séquences avec ces méthodes, ils doivent trouver l'exécutable ou le code source de ces méthodes. Il y a présentement très peu de méthodes qui offrent ceci ; les chercheurs doivent alors programmer eux-mêmes ces méthodes. Cette tâche étant relativement difficile, les mêmes méthodes simples, vieilles et très connues sont fréquemment choisies. C'est un réflexe très normal de se comparer à des méthodes simples car elles sont faciles à implémenter, mais elles sont aussi moins précises, ce qui avantage encore une fois les chercheurs lorsqu'ils se comparent. Une fois les méthodes programmées, il est fréquent que les chercheurs ne partagent pas leur code, ce qui donne plusieurs versions d'une même méthode, mais toutes inaccessibles. De plus, il est risqué de programmer soi-même une méthode car l'auteur de celle-ci peut accuser le programmeur d'avoir mal compris sa méthode ou d'avoir introduit des bogues dans son implémentation. Bref, les bonnes comparaisons avec des méthodes complexes et récentes sont rares et les vidéos choisies avantagent fréquemment les auteurs.

Une fois les méthodes choisies, les auteurs doivent sélectionner une méthode d'évaluation, soit une ou plusieurs métriques pour évaluer quantitativement les méthodes entre elles. Encore une fois, la communauté de détection de changement n'a pas adopté de métrique standard pour les comparaisons. Les auteurs peuvent alors choisir n'importe quelle méthode d'évaluation, dont une qui les avantage.

Conséquemment, les méthodes sont curieusement toutes meilleures les unes que les autres. En se comparant à de vieilles méthodes, les chercheurs concluent avec raison que leur solution est plus précise. Mais il est bien sûr impossible que toutes ces méthodes soient les meilleures. Il serait plus pertinent que ces nouvelles méthodes se comparent entre elles et non avec les méthodes des années 90, ou que les séquences choisies soient standardisées afin que les chercheurs se comparent sur une même base.

La situation présente pourrait être corrigée si la plupart des chercheurs offraient au moins un exécutable de leur méthode. Il faudrait toutefois que la commnunauté s'entende sur le format utilisé et sur le nommage des fichiers, ce qui est loin d'être le cas. Fournir le code serait alors la solution évidente pour régler ce problème, mais ce n'est pas non plus une solution rêvée car le code est rarement présentable. La propriété intellectuelle et les brevets sont une autre raison majeure qui explique le manque de partage dans cette communauté. Cela dit, quelles que soient les raisons, il est clair que les chercheurs sont réticents à partager leur code et il serait malheureusement difficile de les convaincre de changer leurs habitudes.

1.8 Améliorations possibles

Il y a quelques années, la communauté de stéréovision faisait face à un problème similaire à celui expliqué dans la section 1.7. En 2002, Daniel Scharstein et Richard Szeliski publient un article [62] qui présente le site de vision de Middlebury[5]. Ce site devient rapidement le standard *de facto* en stéréovision car il apporte la solution à des problèmes réels : la difficulté de créer ses propres données ou de les trouver, la difficulté de se comparer avec les autres méthodes et la difficulté de trouver des métriques de comparaison. Ce site

5. `http://vision.middlebury.edu/stereo/`

web est simple ; il permet de télécharger un ensemble de données, donne les outils pour travailler avec celles-ci et permet de téléverser ses résultats afin qu'ils soient comparés automatiquement et que le score soit affiché au grand public.

Malheureusement, ce site n'a pas que des points positifs. L'intention de départ était bonne, mais l'exécution pourrait être améliorée. Bien qu'il y ait trois jeux de données, les chercheurs utilisent encore seulement le premier qui était plutôt modeste. En effet, sur les 9 couples d'images, seulement deux ont une réalité de terrain, soit « Sawtooth » et « Tsukuba ». Malheureusement, « Tsukuba » est faite à la main et est donc hautement imprécise. Les résultats compilés sur le site sont bien sûr calculés avec les deux seules réalités de terrain disponibles, ce qui motive malheureusement certains auteurs à sur-spécialiser leurs méthodes pour bien réussir sur ces images. Bref, le principal test de ce site ne représente pas la réalité et, victime de sa propre popularité, le site n'a pas réussi à convaincre les chercheurs d'utiliser les autres ensembles de données.

Les points positifs apportés par ce site sont tout de même majeurs et bien réels. Nous croyons d'ailleurs que c'est ce qu'il manque à la communauté de détection de changement. Nous avons l'intention de rendre disponible un ensemble de vidéos avec leurs réalités de terrain afin de permettre aux chercheurs de se comparer sur une base commune, procurer des outils pour travailler avec ces données et, finalement, fournir une page d'informations qui présente les scores des différentes méthodes selon une comparaison quantitative. Tout ceci dans le but que les chercheurs aient des outils pour comparer leurs méthodes et que la communauté scientifique et commerciale puisse savoir quelles méthodes excellent dans quelles catégories.

Chapitre 2

Travaux antérieurs

Ce chapitre porte sur les banques de données ainsi que sur les métriques d'évaluation des méthodes de détection de changement. Dans chaque section, un survol des travaux antérieurs est présenté. Dans un premier temps, quelques explications sur l'annotation sont fournies, puis nous voyons quelques projets sur les banques de données. Le chapitre se termine avec une présentation de la matrice de confusion et des mesures en classification binaire, puis un survol des techniques d'évaluation des méthodes de détection de changement.

2.1 Banques de données

Plusieurs banques de données en vidéosurveillance ont vu le jour durant les dernières décennies. Certaines étaient très modestes et d'autres étaient le fruit de projets d'envergure. Aucune ne peut être considérée comme un standard *de facto* en détection de changement, mais certaines comme « Wallflower » et « PETS » ont tout de même acquis une certaine notoriété. Cette section présente les banques de données les plus fréquemment utilisées en détection de changement.

2.1.1 Annotation

L'évaluation quantitative des méthodes exige l'annotation de séquences vidéo, donc de créer les réalités de terrain pour un certain nombre d'images. On suppose que ces dernières sont correctes, mais il faut prendre en considération certains points. Pour des séquences réelles, le processus d'annotation ne peut pas être complètement automatisé et doit donc être fait à la main. Il est possible d'automatiser certaines parties de l'annotation, mais uniquement celles qui ne peuvent nuire à la qualité des résultats. Pour le reste, l'annotation est un travail répétitif sujet aux erreurs subjectives et aux erreurs d'inattention [63, 64]. La bordure des objets, la qualité des images et les artéfacts de compression sont des facteurs d'incertitude pour la personne qui annote. Idéalement, l'annotation serait faite par plusieurs personnes et les résultats moyennés afin de minimiser les erreurs. Ceci n'est toutefois pas réaliste pour des raisons de temps, d'argent et de motivation. Même sans ces contraintes, il est pratiquement impossible de créer des réalités de terrain parfaites [64].

L'annotation demande beaucoup de temps et d'effort [63]. Il est d'usage de trouver des raccourcis pour éviter d'annoter toutes les images d'une séquence. Certaines banques de données offrent simplement une série de vidéos sans annotation [65] ; les chercheurs doivent alors segmenter eux-mêmes les séquences pour se comparer avec les autres méthodes [66]. D'autres banques de données offrent l'annotation pour un sous-ensemble des images, par exemple 10 images réparties également vers la fin d'une séquence [28].

Les défis de l'annotation ont contribués à l'apparition de trois types de vidéos, soient synthétiques, semi-synthétiques et réelles. Une séquence synthétique est complètement rendue à l'aide d'un logiciel de synthèse d'image ; rien n'y est réel. Une séquence semi-synthétique utilise généralement un arrière-plan réel, capturé avec une caméra fixe ordinaire, où des objets en mouvement seront par la suite ajoutés par ordinateur. Ces deux types sont très avantageux au niveau de l'annotation des séquences car elle peut facilement être automatisée. Étant déjà virtuels, les objets en mouvement sont utilisés pour produire aisément les réalités de terrain. Les séquences réelles, quant à elles, ne sont modifiées d'aucune façon. L'annotation doit alors se faire manuellement car les modèles n'existent pas et aucun algorithme n'est encore capable de segmenter parfaitement des séquences réelles.

Face à ces avantages au niveau de l'annotation, on pourrait s'attendre à ce que l'utilisation de séquences synthétiques et semi-synthétiques soit très répandue, mais ce n'est pas le cas. Les séquences synthétiques ont fréquemment une apparence peu réaliste. Les séquences semi-synthétiques ont aussi un problème de réalisme. Lorsque des objets en mouvement sont ajoutés à une scène, ils ont rarement les bonnes propriétés de profondeur et de luminosité. De plus, les pixels bordant les objets peuvent être de la mauvaise couleur. Les objets en mouvement n'ont aucune interaction avec la scène et l'illumination n'a pas d'influence sur l'ombre qu'ils projettent et sur l'intensité de leurs pixels. De plus, l'objet synthétique n'est pas réfléchi dans la scène. Comme on peut le voir dans la figure 2.1, tout ceci cause de graves problèmes au niveau du réalisme de la scène, ce qui peut biaiser les résultats en modifiant les difficultés standards. Notons qu'une séquence synthétique ayant une apparence parfaitement réelle ne causerait probablement aucun problème, mais les laboratoires de recherche ont rarement les ressources nécessaires pour créer ce type de séquence.

a) « video6 » b) « video7 »

FIGURE 2.1 – La plupart des séquences synthétiques en vidéosurveillance ont une apparence peu réaliste. Images empruntées à la banque de données VSSN06 [65]. a) Les modèles sont illuminés uniformément et n'ont pas la même résolution. Une bordure blanche est visible dans les cheveux de la jeune fille. b) Mêmes problèmes d'illumination et absense d'ombre sous le véhicule.

Bien qu'aucun chercheur ne l'ait fait, il serait possible de capturer des séquences ne comportant aucun objet en mouvement. Il pourrait y avoir beaucoup de pièges dans la

scène, mais l'absence d'objet en mouvement permettrait de créer des réalités de terrain complètement statiques, donc de ne rien créer. Ce type de séquences aurait son utilité pour vérifier la robustesse des méthodes de détection de changement, mais il aurait tendance à biaiser les résultats dans un banc d'essai. En effet, l'absence de mouvement dans les séquences avantagerait des méthodes qui sous-estiment systématiquement le nombre de pixels statiques.

Le travail d'annotation est différent selon l'intention des auteurs. En détection de changement, il est fait au niveau des pixels, ce qui donne généralement un masque binaire (voir figures 1.2 et 1.3). Par contre, pour valider des méthodes de suivi d'objets (*tracking*), de détection et d'autres domaines reliés, il est fait au niveau des objets avec des boîtes englobantes. De plus amples détails seront donnés dans les prochaines sections.

2.1.2 Segmentation basée sur les pixels

Beaucoup d'articles utilisent des vidéos à usage unique pour comparer leurs résultats. Dans certains cas, ils utilisent tout de même une banque de données existante, en partie ou en totalité, ou même un mélange de séquences de plusieurs banques de données. Tel qu'expliqué dans la section 1.7, ceci a des conséquences fâcheuses sur la qualité et la validité des comparaisons. Voici une liste de quelques articles et sites web parmi les plus cités ayant fourni une banque de données à la communauté scientifique.

En 1999, Toyoma *et al.* [67] publient une méthode de détection de changement nommée « Wallflower ». Ils comparent cette dernière en utilisant une banque de données de leur création. Très connue et encore utilisée à ce jour, elle contient 7 vidéos (voir figure 2.2), chacune correspondant à une difficulté spécifique de la détection de changement. Certaines difficultés ne sont pas considérées, comme le mouvement de la caméra, les ombres et le bruit. Les séquences sont toutes de taille 160x120 et deux sont de courte durée, soient 289 et 355 images. Les auteurs ont annoté à la main uniquement une image par vidéo, ce qui fait 7 images annotées sur un total de 15804 images. Cette banque de données et son annotation sont encore disponibles[1] à ce jour, gratuitement et aisément.

1. `http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm`

FIGURE 2.2 – 7 images des séquences de la banque de données Wallflower [67].

En 2004, Cheung *et al.* [28] évaluent la performance de 6 méthodes de soustraction de fond sur des séquences de circulation automobile en milieu urbain. Leur banque de données contient 4 vidéos (voir figure 2.3) choisies sur le site de l'*Institut für Algorithmen und Kognitive Systeme*[2]. Ce site contient 14 séquences fournies par différents auteurs, donc de résolutions différentes, presque toutes en niveaux de gris. Ces 4 séquences ont été choisies afin de mettre l'accent sur les difficultés causées par les conditions météorologiques. Ces vidéos représentent du trafic routier par temps clair, brumeux, neigeux et très ensoleillé. Trois montrent le même coin de rue, mais une est capturée d'un point de vue légèrement différent. Bien que les méthodes testées soient capables d'utiliser les informations de couleur, la moitié des séquences utilisées sont en niveaux de gris. Une petite partie de ces séquences a été annotée, soit 10 images à intervalle régulier pour chaque séquence. Les séquences sont encores disponibles à ce jour, mais les 40 images annotées ne le sont plus.

FIGURE 2.3 – 4 images des séquences de la banque de données Cheung *et al.* [28].

2. `http://i21www.ira.uka.de/image_sequences/`

L'article publié par Li *et al.* [68] en 2004 présente une méthode de détection de mouvement bayésienne validée sur 10 séquences (voir figure 2.4). Cette banque de données contient autant de scènes extérieures qu'intérieures et fait face à la plupart des difficultés usuelles. Les séquences sont de petite taille, soit 160x120 ou légèrement plus. Certaines scènes contiennent beaucoup d'objets en mouvement dans de courts laps de temps. Toutes les séquences sont annotées à raison de 10 images par séquence. Cette banque de données et son annotation sont disponibles[3] gratuitement et simplement, mais la séquence « SW » (SideWalk) et son annotation ne le sont pas.

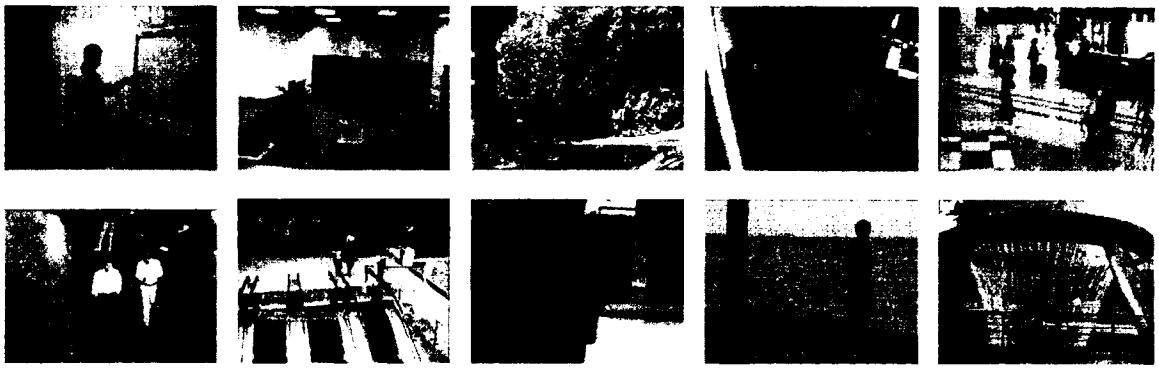


FIGURE 2.4 – 9 images des séquences de la banque de données Li *et al.* [68] ; légèrement rognées et redimensionnées à des fins de visualisation.

En 2005, Karaman *et al.* [69] évaluent la performance de 9 méthodes de soustraction de fond. Ils ont une banque de données de 5 vidéos (voir figure 2.5) ; soit deux de leur propre création et trois empruntées au projet artistique européen « art.live ».[4] Ces vidéos ne sont pas directement reliées aux difficultés standards de la détection de changement (voir section 1.5) ; l'accent a plutôt été mis sur des artéfacts de compression et sur différents degrés d'illumination. L'annotation est faite à la main, en partie par des étudiants à l'École Polytechnique Fédérale de Lausanne et en partie par les auteurs. Il n'est pas précisé si la totalité ou seulement une partie de la vidéo a été annotée. Cette banque de données et son annotation ne sont plus disponibles.

3. `http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`
4. `http://www.tele.ucl.ac.be/PROJECTS/art.live/home.html`

FIGURE 2.5 – 5 images des séquences de la banque de données Karaman *et al.* [69].

Une banque de données de 9 séquences (voir figure 2.6) est créée en 2006 pour les besoins d'une compétition organisée par la conférence VSSN [65]. Les séquences mettent l'accent sur l'arrière-plan dynamique, les changements d'illumination et les ombres. Une des scènes a été capturée avec 2 caméras afin de permettre aux chercheurs d'utiliser l'information dupliquée. L'annotation est fournie uniquement pour les quatre séquences semi-synthétiques ; les séquences réelles n'ont pas été annotées. Les quatre séquences annotées et sept des neuf séquences de cette banque de données sont disponibles [5] aisément et gratuitement ; le reste n'est pas disponible sur le site officiel.

FIGURE 2.6 – 7 images des séquences de la banque de données VSSN06 [65]. Les deux dernières images représentent la même scène capturée par deux caméras.

Un article de Yin *et al.* [70] publié en 2007 inclut une banque de données de 28 vidéos. Toutes les séquences ont une résolution de 320x240 et sont de courte durée, entre 1 seconde

5. `http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/`

et 53 secondes. Comme on peut le voir dans la figure 2.7, les vidéos représentent toutes une personne (deux dans un cas) assises, discutant avec un destinataire inconnu à l'aide d'une webcaméra. Les seuls objets détectés sont ces personnes, même lorsque d'autres personnes sont présentes dans l'arrière-plan. Bien que ces séquences ne soient pas toutes adaptées à la détection de changement, certaines sont utilisées dans le domaine. L'annotation est effectuée à toutes les 5 ou 10 images. Bon nombre de ces séquences ont aussi été capturées en stéréovision. Ces séquences monoculaires et binoculaires et leurs annotations sont accessibles[6] facilement et sans contrainte.

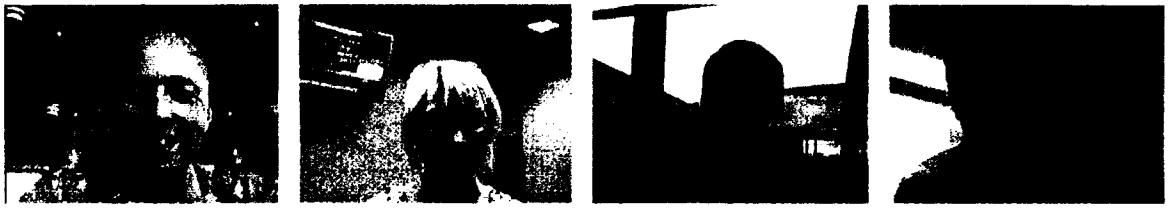


FIGURE 2.7 – 4 images des séquences de la banque de données Yin *et al.* [70].

En 2008, Parks *et al.* [57] évaluent la performance de 7 algorithmes avec un pot-pourri de 13 séquences empruntées à 5 sources différentes. La plupart des difficultés usuelles sont visées, sauf le mouvement de la caméra et le mouvement intermittent des objets. Encore une fois, les séquences et annotations sont impossibles ou difficiles à trouver, et elle ne sont pas visibles dans l'article correspondant.

La même année, Tiburzi *et al.* [71] créent une banque de données semi-synthétique contenant 15 vidéos (voir figure 2.8). Les objets en mouvement sont enregistrés dans un studio afin d'obtenir leur segmentation automatiquement. Ils sont ensuite combinés entre eux avec différents arrière-plans afin de bien représenter certaines des difficultés standards. Cette banque de données est encore disponible[7], mais son utilisation demande certaines actions contraignantes : imprimer un accord de licence, le remplir et le faxer, puis envoyer un courriel contenant l'adresse IP et l'équipement qui sera utilisé pour télécharger la banque de données.

6. `http://research.microsoft.com/en-us/projects/i2i/data.aspx`
7. `http://www-vpu.ii.uam.es/CVSG/`

FIGURE 2.8 – 10 images de séquences différentes de la banque de données cVSG de Tiburzi *et al.* [71]. Les 5 autres séquences sont un mélange de ces acteurs et de ces scènes.

En 2011, Brutzer *et al.* [59] créent une banque de données complètement synthétique et profitent de l'occasion pour évaluer 9 algorithmes de soustraction de fond. Ils offrent 9 séquences de taille 800x600, toutes rendues avec *Mental Ray*, un programme de lancer de rayons, afin d'atteindre un certain photoréalisme. Ces séquences représentent toutes le même coin de rue sous le même angle ; les différences sont au niveau de l'illumination, de la compression et des objets en mouvement. La figure 2.9 donne un aperçu du réalisme de la qualité des séquences. Les séquences correspondent aux difficultés usuelles de la soustraction de fond. L'annotation a été faite automatiquement et en entier car les séquences sont synthétiques. Cette banque de données est disponible[8] simplement et sans contrainte.

Ce survol de la littérature nous informe que l'annotation des séquences réelles est tenue au minimum. Les vidéos avec le plus de réalités de terrain ont une dizaine d'images annotées ou légèrement plus. L'existence d'un programme d'aide à l'annotation aurait sans doute amélioré la situation, mais il n'en existe aucun. De plus, certaines des banques de données décrites précédemment présentent une méthode de comparaison, mais aucune n'offre de système en ligne permettant de comparer les résultats des différentes méthodes. Le site de vision de Middlebury l'a démontré, ce type de système sauve un temps énorme aux chercheurs et contribue directement à accroître la qualité des comparaisons.

8. `http://www.vis.uni-stuttgart.de/index.php?id=sabs`

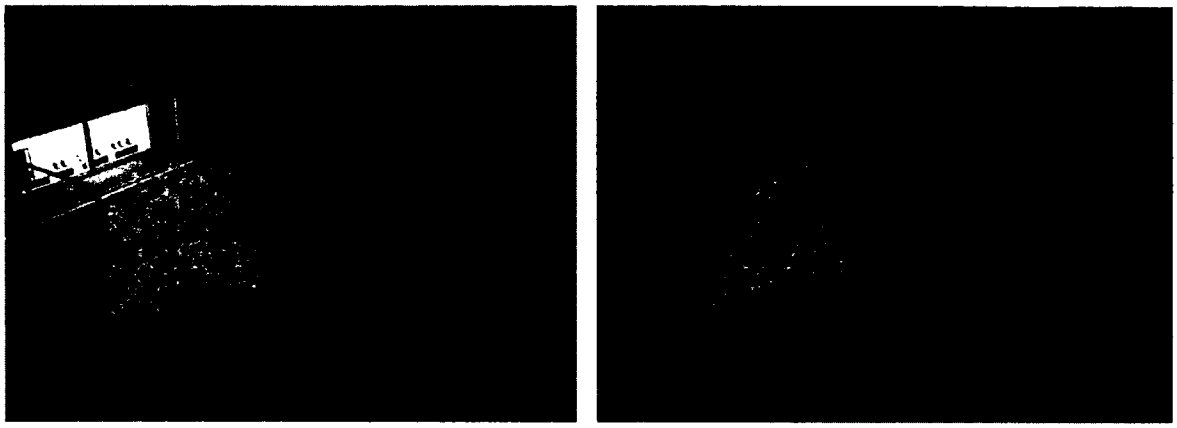


FIGURE 2.9 – Deux images consécutives de la séquence « LightSwitch » de la banque de données Brutzer *et al.* [59] où la lumière d'une boutique se ferme soudainement afin de tester la robustesse des méthodes face aux changements d'illumination. Luminosité augmentée à des fin de visualisation.

2.1.3 Segmentation basée sur les objets

Tel que mentionné précédemment, une segmentation basée sur les pixels est inadéquate pour valider les résultats du suivi d'objets et de la reconnaissance de formes et d'actions. Dans ces cas, le but n'est plus de savoir avec précision dans quels pixels sont les objets en mouvement, mais plutôt dans quelles zones se trouve l'action. Une simple boîte englobante (voir 2.10) est utilisée lors de l'annotation car elle est suffisamment précise. Pour les besoins des algorithmes de suivi et de reconnaissance, les boîtes englobantes sont numérotées et peuvent être annotées avec une description de l'action en cours. La segmentation au niveau des pixels est ignorée dans ces domaines, mais les vidéos sont parfois utilisées par des auteurs pour leurs comparaisons.

L'utilisation de séquences synthétiques ou semi-synthétiques est moins fréquente dans ces domaines car l'annotation avec des boîtes englobantes peut se faire plus aisément. De plus, l'existence de certains programmes libre d'utilisation pour annoter des séquences simplifie grandement la création de réalités de terrain. Parmi ceux-ci, on retrouve CAVIAR[9],

9. Anciennement disponible à `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`.

FIGURE 2.10 – Représentation des boîtes englobantes autour des objets en mouvement. Boîtes ajoutées manuellement ; les informations sont dans un fichier xml. Images et informations empruntées à la banque de données « BEHAVE ».

AVITRACK [10], ODViS [11] et ViPER [12]. Les trois premiers ne sont plus disponibles sur Internet, mais ils est probablement possible de les obtenir en demandant une copie aux auteurs des articles correspondants.

PETS [13], ETISEO [14], BEHAVE [15], IBM [16] et i-LIDS [17] sont 5 exemples de banques de données utilisant des boîtes englobantes. Il est possible d'en trouver beaucoup d'autres en consultant la page d'informations du projet européen CANTATA [18]. PETS et ETISEO contiennent ensemble plus de 200 séquences. En combinant les particularités de ces 5 banques de données, nous avons un ensemble de défis très diversifiés, tant au niveau de la classification (suivi, comptage, reconnaissance d'expressions faciales, direction des regards,

10. Anciennement disponible à `http://www.avitrack.net` Utilisé pour l'annotation de la banque de données CAVIAR.

11. Anciennement disponible à `www.metaverselab.org/software/odvis`

12. `http://viper-toolkit.sourceforge.net/` Utilisé dans l'annotation de quelques banques de données, dont Visor et BEHAVE.

13. `http://www.cvg.rdg.ac.uk/slides/pets.html`

14. `http://www-sop.inria.fr/orion/ETISEO/`

15. `http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/`

16. `http://www.research.ibm.com/peoplevision/performanceevaluation.html`

17. `http://www.homeoffice.gov.uk/science-research/hosdb/i-lids/`

18. `http://www.hitech-projects.com/euprojects/cantata/datasets_cantata/dataset.html`

etc.), des caméras (type, qualité et nombre) que du type de scènes (densité d'objets en mouvement, période de la journée, conditions météorologiques, lieux, etc.). Ces banques de données étaient, pour la plupart, utilisées pour une compétition lors de conférences scientifiques ; les résultats ne sont donc malheureusement pas mis à jour, mais les vidéos sont toujours disponibles à des fins de tests et de comparaisons. Les chercheurs sont donc avantagés quant au choix des vidéos, mais pas au niveau des comparaisons.

Il serait possible d'utiliser l'annotation de ces banques de données comme base pour automatiser la création de réalités de terrain en détection de mouvement, donc basées sur les pixels. Ceci simplifierait le travail, dans l'optique où tout pixel qui n'est pas dans une boîte englobante est nécessairement statique. Malgré cet avantage indéniable, les pixels dans les boîtes englobantes doivent tout de même être classifiés. Ceci peut être automatisé en partie, mais l'intervention manuelle d'un humain est nécessaire pour réparer les erreurs qui seront certainement présentes. De plus, les vidéos choisies pour évaluer les méthodes de suivi visent normalement des défis différents de ceux visés par les méthodes de détection de mouvement ; on peut alors se questionner sur la pertinence d'utiliser ces vidéos.

Le projet ViSOR [19] [72] est un effort d'envergure en vidéosurveillance. Accessible publiquement sur Internet, on y trouve des outils pour annoter des séquences (ViPER) et pour évaluer les performances des méthodes (en construction). On peut aussi trouver de nombreuses séquences déjà annotées, des articles scientifiques sur la vidéosurveillance, une onthologie pour la création des fichiers xml et un forum pour la communauté de vidéosurveillance. ViSOR avait tous les outils pour être le portail de la vidéosurveillance, mais sans mise à jour importante depuis 2009, l'effort semble être tombé dans l'oubli. Un site de ce type dépend de la communauté pour survivre car le contenu est fourni en grande partie par ces derniers.

19. `http://www.openvisor.org/`

2.2 Évaluation

Une méthode d'évaluation permet de comparer quantitativement deux méthodes de détection de changement entre elles ou entre une méthode et la réalité de terrain afin d'établir un classement entre les méthodes. Uniquement les comparaisons avec la réalité de terrain sont étudiées dans le cadre de ce travail. Il existe trois sources de méthodes d'évaluation en détection de changement : les chercheurs qui publient une nouvelle méthode de segmentation, ceux qui présentent un banc d'essai et ceux qui publient directement une méthode d'évaluation. Dans les deux premiers cas, le besoin de se comparer quantitativement les force à trouver une méthode d'évaluation. Encore une fois, le sujet peut être divisé entre les méthodes d'évaluation sur les pixels et celles avec les boîtes englobantes. De par leur nature, ces deux types de segmentation ont des méthodes d'évaluation différentes.

2.2.1 Évaluation basée sur les pixels

L'évaluation d'une segmentation basée sur un masque binaire consiste à calculer la similarité entre ce masque et la réalité de terrain. Bien que la tâche semble simple, calculer la similarité entre deux images binaires s'avère être un problème délicat. Comme la plupart des méthodes d'évaluation en détection de changement utilisent les informations contenues dans une matrice de confusion, nous verrons en premier lieu les mesures standards en classification binaire, puis nous terminerons avec des méthodes d'évaluation de quelques auteurs.

Mesures en classification binaire

Une comparaison pixel à pixel entre deux images binaires ouvre 4 possibilités, soient 0-0, 0-1, 1-0 et 1-1, où 0 correspond à l'étiquette « statique » et 1 à l'étiquette « mouvement ». Il est d'usage d'utiliser une matrice de confusion pour représenter ces valeurs [73]. Les cases contiennent le nombre de pixels correspondants à la description.

Comme le montre le tableau 2.1, une matrice de confusion binaire contient 4 données de base qui, dans l'optique de la détection de changement, équivalent à :

TABLE 2.1 – Matrice de confusion

	Classes prédites	
Classes réelles	**Vrai Positif**	**Faux Négatif**
	Faux Positif	Vrai Négatif

VP - Vrai positif nombre de pixels en mouvement, bien détectés ;

FP - Faux positif nombre de pixels en mouvement, mal détectés ;

FN - Faux négatif nombre de pixels statiques, mal détectés ;

VN - Vrai négatif nombre de pixels statiques, bien détectés,

où l'on peut obtenir le nombre total de pixels en mouvement (#mouvement) avec $VP+FN$, le nombre total de pixels statiques (#statique) avec $FP+VN$ et le nombre total de pixels (#total) avec $VP+FN+FP+VN$.

Bien que ces quatre chiffres représentent déjà une évaluation quantitative, il est d'usage de calculer une série d'autres mesures qui ont un sens plus évocateur et aisément compréhensible. Surtout dans ce cas-ci où nous comptons des pixels ; les quatre chiffres seront très grands et n'auront donc pas un sens évident à première vue. De plus, on peut se demander comment classer des méthodes sur la base de ces quatre chiffres ; ils doivent au minimum être combinés afin d'obtenir un classement unique. Les prochains paragraphes et équations portent sur des mesures de performance fréquentes en classification binaire.

La **précision** est la fraction des pixels en mouvement bien classés sur le nombre total de pixels classifiés en mouvement. Elle peut être vue comme la probabilité qu'un pixel pris au hasard dans ceux en mouvement soit réellement en mouvement. Une haute précision implique que très peu de pixels statiques ont été classifiés en mouvement.

$$Pr = \frac{VP}{VP+FP}$$

Le **rappel**, ou **sensibilité**, est la fraction des pixels en mouvement bien classés sur le nombre de pixels en mouvement. Il peut être vu comme la probabilité qu'un pixel en

mouvement soit classé comme tel. Un haut rappel implique que la plupart des pixels classés en mouvement étaient réellement en mouvement.

$$Ra = \frac{VP}{VP + FN} = \frac{VP}{\#\text{mouvement}}$$

Utiliser seulement la précision ou le rappel peut introduire un biais dans les résultats. La précision avantage les méthodes qui sous-estiment le nombre de pixels en mouvement car elle ignore les faux positifs. Le rappel avantage les méthodes qui surestiment le nombre de pixels en mouvement en ignorant les faux négatifs. Comme nous le verrons avec la f-mesure et les courbes PR, utiliser ensemble ces deux métriques donne de bons outils d'évaluation.

La **f-mesure** est la moyenne harmonique de la précision et du rappel. Elle requiert autant une bonne précision qu'un bon rappel. Une haute f-mesure implique que peu de pixels ont été mal classifiés.

$$fm = 2\frac{Pr \cdot Re}{Pr + Re}$$

La **spécificité**, ou **taux de vrais négatifs**, est la fraction des pixels statiques bien classifiés sur le nombre de pixels statiques total. Il peut être vu comme la probabilité qu'un pixel en mouvement ait été classifié statique. Une haute spécificité implique que peu de pixels statiques ont été classifiés en mouvement.

$$Sp = \frac{VN}{VN + FP} = \frac{VN}{\#\text{statique}}$$

Le **taux de faux positifs**, aussi appelé le taux de fausses alarmes, est la fraction des pixels en mouvement mal classifiés sur le nombre total de pixels classifiés en mouvement. Il peut être vue comme la probabilité de rejeter un pixel en mouvement.

$$T_{FP} = \frac{FP}{FP + VN} = \frac{FP}{\#\text{statique}}$$

Le **taux de faux négatifs** est la fraction des pixels statiques mal classifiés sur le nombre total de pixels classifiés en mouvement. Il peut être vu comme la probabilité de rejeter un pixel statique.

$$T_{FN} = \frac{FN}{VP + FN} = \frac{FN}{\#\text{mouvement}}$$

Le **pourcentage de bonnes classifications**, PBC, est la probabilité d'une bonne classification sur un pixel pris au hasard. Bien que cette mesure semble adéquate, elle peut donner des résultats trompeurs lorsqu'il y a très peu de positifs dans les données. C'est le cas de bon nombre de vidéos de surveillance dont moins de 5% du contenu est en mouvement. Une méthode de détection de changement ayant tendance à sous-estimer les objets en mouvement (faux négatifs élevés) aurait alors un bon score pour cette mesure.

$$PBC = 100\frac{VP + VN}{VP + FN + FP + VN} = 100\frac{VP + VN}{\#\text{total}}$$

Inversement, le **pourcentage de mauvaises classifications**, PMC, est la probabilité d'une mauvaise classification sur un pixel pris au hasard. Il est obtenu avec $100 - PBC$ ou avec l'équation suivante :

$$PMC = 100\frac{FN + FP}{VP + FN + FP + VN} = 100\frac{FN + FP}{\#\text{total}}$$

Comme elles sont construites à partir d'une matrice de confusion binaire, certaines de ces mesures sont opposées à une autre mesure. Si une mesure augmente, l'autre diminuera et inversement.

$$T_{FP} = 1 - \text{spécificité}$$

$$T_{FN} = 1 - \text{sensibilité}$$

$$PBC = 100\% - PMC$$

Ainsi, une méthode qui sous-estime le nombre de pixels en mouvement aura un meilleur score sur la précision, la spécificité et le T_{FN} alors qu'une méthode qui surestime aura un meilleur score sur le rappel et le T_{FP}. Le choix des métriques prend alors toute son importance.

Le rappel et le T_{FP} ne sont pas des opposés au même titre que les autres valeurs, mais il y a tout de même un compromis entre ceux deux mesures. Elles peuvent être utilisées dans la construction d'une courbe ROC. Pour ce faire, il suffit de tester une méthode avec un seuil très faible, puis de l'augmenter jusqu'à ce qu'il soit très élevé. Lorsqu'il est très faible, un grand nombre de pixels sont détectés en mouvement, le rappel et le T_{FP} se rapprochent alors de 0. Lorsqu'il est très élevé, les méthodes ne détectent presque plus rien alors les deux mesures se rapprochent de 1. Ce calcul effectué pour chaque seuil donne

un graphique qui ressemble à celui de la figure 2.11. Bien qu'il y ait un compromis entre ces deux métriques, une bon algorithme de détection de mouvement devrait atteindre un rappel aussi haut que possible en gardant un taux de faux positifs minimal.

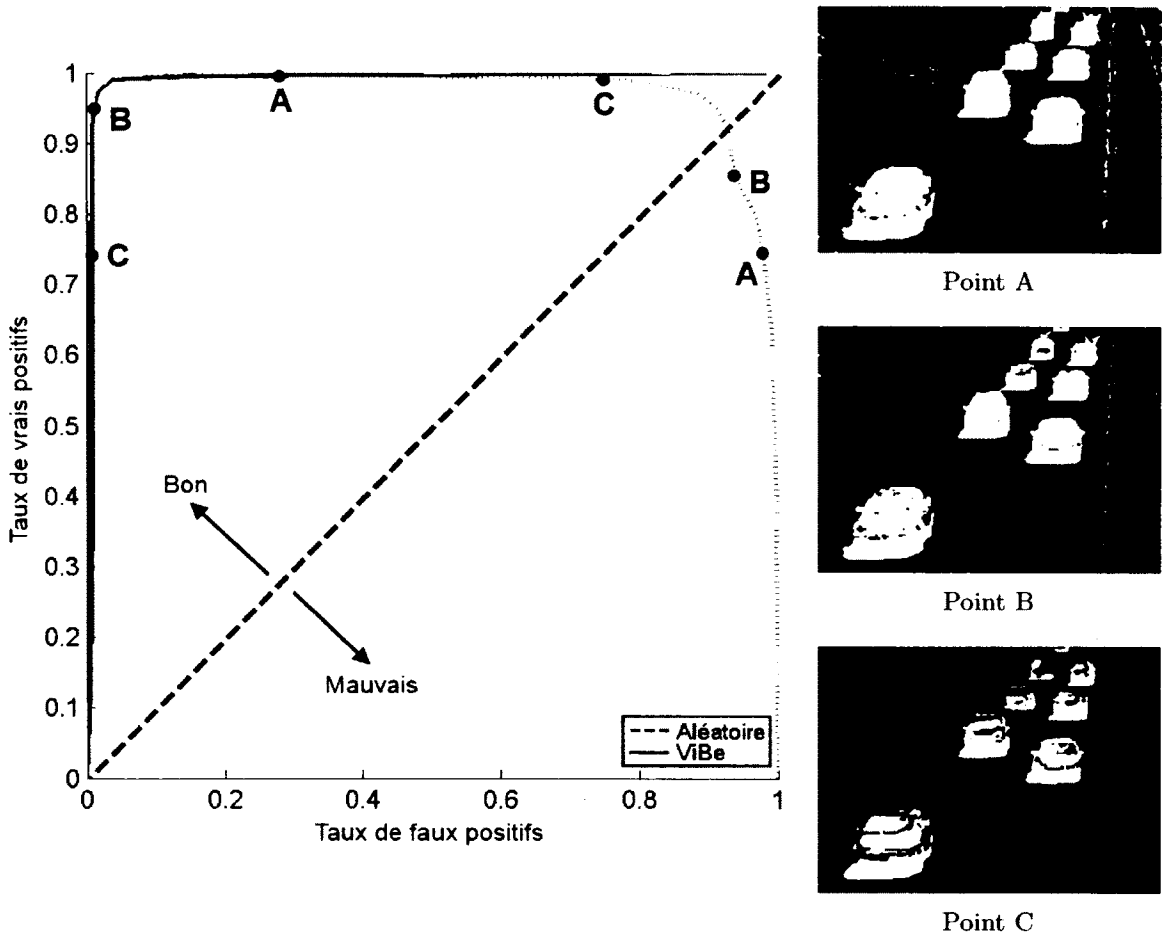


FIGURE 2.11 – Courbe ROC pour la méthode ViBe [41] calculée uniquement sur l'image 795 de la séquence « Highway ». Trois seuils ont été choisis en A, B et C afin d'expliquer l'effet du seuil sur la qualité des masques binaires. La ligne constituée de tirets minces représente la courbe PR selon les mêmes résultats. Avec cette ligne, l'axe des ordonnées devient la précision et l'axe des abscisses, le rappel. A) $\tau = 6$: $Ra = 98.3\%$, $T_{FP} = 26.5\%$ B) $\tau = 10$: $Ra = 95.4\%$, $T_{FP} = 1.3\%$ C) $\tau = 30$: $Ra = 73\%$, $T_{FP} = 0.8\%$

Un article publié en 2006 de Davis et Goadrich [74] montre qu'il est préférable d'utiliser

une courbe PR (Précision-Rappel) au lieu d'une courbe ROC lorsque les classes sont asymétriques. La courbe PR est construite aussi en testant plusieurs seuils, mais elle se définit par le rapport entre la précision et le rappel (voir figure 2.11). Davis et Goadrich ont prouvé que la courbe PR est toujours au moins aussi descriptive que la courbe ROC. Son avantage par rapport à la courbe ROC est qu'il est souvent plus aisé de déterminer si une méthode est meilleure. Malgré le compromis entre la précision et le rappel, une bonne méthode de détection de changement devrait atteindre un rappel aussi haut que possible sans sacrifier de la précision. Par définition, une bonne méthode devrait avoir très peu de faux négatifs et de faux positifs, donc une bonne précision et un bon rappel [58].

Aucune de ces mesures ne fait l'unanimité. Prise individuellement, elles peuvent avantager certaines décisions. Par exemple, une méthode de détection de changement qui classifie tous les pixels à 0 aurait un score parfait pour la spécificité et le taux de faux positifs. De plus, elle serait légèrement avantagée car, dans la plupart des vidéos, il y a beaucoup plus de pixels statiques que de pixels en mouvement. Inversement, une méthode qui répond que tous les pixels sont en mouvement aurait un score parfait sur le rappel et sur le taux de faux négatifs.

Méthodes d'évaluation

Les créateurs de la banque de données « Wallflower », Toyama *et al.* [67] additionnent le total de faux positifs et de vrais négatifs pour chaque méthode comparée. La méthode ayant le moins de pixels mal classifiés (voir figure 2.12) est considérée meilleure.

FIGURE 2.12 – Capture d'écran de l'article de Toyama *et al.* [67] où les méthodes de détection de changement sont jugées selon leur nombre d'erreurs, soit $FP + FN$.

Les auteurs précisent que leur classement final ne devrait pas être considéré comme une référence car ces résultats sont dépendants de l'application en cours et des séquences choisies. Malgré cette mise en garde, la même méthodologie et la même banque de données sont utilisées par Bouwmans *et al.* en 2008 [75] et en 2011 [76] pour comparer les performances de différentes méthodes utilisant une mixture de gaussiennes.

En 2005, Brown *et al.* [77] évaluent la performance de quelques méthodes de détection de changement et de suivi. Ils utilisent directement les valeurs de la matrice de confusion, soient VP, FP, FN et VN, mais ils calculent aussi la taille des taches résultant des faux positifs et des faux négatifs.

Rosin et Ioannidis [78] publient en 2003 une méthode d'évaluation qui utilise trois mesures : le pourcentage de bonnes classifications, le **coefficient de Jaccard**, C_J, et le **coefficient de Yule**, C_Y :

$$C_Y = |\frac{VP}{VP+FP} + \frac{VN}{VN+FN} - 1|$$

$$C_J = \frac{VP}{VP+FP+FN}$$

Le coefficient de Yule ne peut pas être appliqué lorsque la classification des pixels statiques ou en mouvement est parfaite car il y aurait une division par zéro. Le pourcentage de bonnes classifications n'est pas utilisé seul car, comme le précise les auteurs, il y a très peu de mouvement dans les séquences. L'utilisation des coefficients de Yule et Jaccard permet de réduire l'effet du grand nombre de vrais négatifs. Lorsqu'il n'y a pas de mouvement dans les séquences, les images sont analysées séparément afin de mieux comprendre les effets du bruit et des artéfacts de compression.

La même année, Chalidabhongse *et al.* [79] publient une méthode de comparaison basée sur ce qu'ils nomment le « taux de perturbation ». Le principal avantage de cette dernière est de ne pas nécessiter de réalité de terrain. Ils choisissent N images de leurs séquences où il n'y a aucun mouvement réel, puis ils ajoutent des vecteurs de perturbation à des endroits aléatoires dans les images. Ces vecteurs (R', G', B') sont obtenus en générant des points uniformément distribués sur une sphère colorimétrique de rayon Δ ; les couleurs perturbées sont alors de contraste Δ. Les méthodes de détection de changement doivent

alors détecter ces vecteurs comme des zones en mouvement. Selon les auteurs, la meilleure méthode est celle qui obtient le plus haut PBC lors des perturbations.

En 2008, Panahi *et al.* [80] utilisent les faux positifs, les faux négatifs et les pourcentages de bonnes et mauvaises classifications normalisés par la taille des objets en mouvement dans les réalités de terrain. Le T_{FP} et le T_{FN} sont aussi utilisés, mais pour être considéré en erreur, un pixel et au moins la moitié de ses voisins doivent être mal classifiés.

Karaman *et al.* [69], créateurs de la banque de données éponyme, utilisent la précision, le rappel et la f-mesure. Ils n'effectuent pas de calcul additionnel avec ces trois mesures ; ils préfèrent créer trois classements selon ces métriques. Ils confirment leur évaluation objective avec une évaluation subjective qui montre visuellement (voir figure 2.13) les bonnes et mauvaies classifications. Ils calculent aussi le temps d'exécution et la consommation mémoire des méthodes testées. Les mêmes mesures sont utilisées lors de la compétition VSSN06 [20] et dans l'article de Herero *et al.* [81] en 2009.

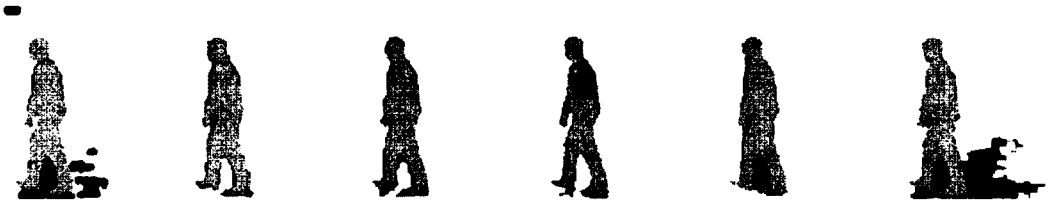


FIGURE 2.13 – Capture d'écran de l'article de Karaman *et al.* [69] Évaluation qualitative de quelques méthodes de soustraction de fond. Vrais positifs en vert, faux négatifs en rouge et faux positifs en bleu.

Probablement influencés par l'article de Davis et Goadrich [74], de nombreux auteurs [28, 57, 66, 58, 59] ont utilisé la courbe PR. En plus d'afficher leurs résultats sous cette forme, Brutzer *et al.* [59] fournissent aussi la F-mesure, probablement parce qu'il est difficile de juger des méthodes uniquement en regardant des courbes.

L'évaluation des méthodes de détection de changement ne porte pas que sur la qualité de leur segmentation. Bien que ce soit souvent le sujet principal, une partie des articles est

20. `http://www.multimedia-computing.de/mediawiki/images/6/65/VSSN-Algo.pdf`

parfois consacrée à d'autres critères. Des études ont été effectuées sur la consommation mémoire, le temps d'exécution [69, 58] et sur l'effet des algorithmes de post-traitement sur la qualité des résultats [57, 58, 59].

2.2.2 Évaluation basée sur les objets

Comme l'annotation est effectuée avec des boîtes englobantes, le but n'est maintenant plus de calculer la similarité entre deux images, mais de calculer la similitude entre la taille et la position de deux rectangles. Encore une fois, aucune mesure ne fait l'unanimité, mais certaines sont plus fréquentes dans la littérature.

Certains auteurs [82, 64, 83] ont tenté d'évaluer des méthodes de détection de changement en utilisant des banques de données annotées avec des boîtes englobantes. Ces dernières forcent une évaluation différente car l'annotation au niveau des pixels n'est pas disponible. Une telle évaluation force une définition différente pour VP, FP, FN et VN ; un pixel détecté en mouvement sera bien classifié s'il est dans une boîte englobante, même s'il ne fait pas réellement partie d'un objet en mouvement.

Pour ce qui est des comparaisons entre les méthodes de suivi d'objets, les mesures de classification restent les mêmes dans le cas d'évaluations basées sur des pixels, mais VP, FP, FN et VN ont un sens différent. Un vrai positif aura plutôt le sens d'une intersection entre l'objet détecté et la réalité de terrain [84, 77], d'une distance euclidienne suffisamment courte entre les deux centroïdes [84, 85] ou du quotient entre le nombre de pixels dans l'intersection et le nombre de pixels dans l'union des deux boîtes englobantes [86]. Il faut toutefois être prudent avec l'identification des boîtes englobantes ; les objets en mouvement se rencontrent parfois et il ne faut pas récompenser une méthode pour un suivi accidentel [77, 85, 86].

Nous constatons à la lecture de ce chapitre qu'il n'y a aucune banque de données non synthétique et complètement annnotée à la disposition des chercheurs et que, dans certains cas, les séquences ne sont plus accessibles. De plus, il n'y a aucun consensus quant à la

banque de données et aux métriques à utiliser. Cette situation est à la source des problèmes de comparaison évoqués dans le premier chapitre. Nous verrons dans les prochaines pages ce que nous avons mis en œuvre afin de corriger ces lacunes.

Chapitre 3

Contributions

Face aux problèmes présentés dans la section 1.7 et inspirés du modèle de Middlebury, nous avons créé une banque de données dédiée à la validation des méthodes de détection de changement. Dans un premier temps, nous présentons notre banque de données en énumérant ses séquences, en détaillant ses particularités et en présentant les outils avec lesquels elle a été créée. Ensuite, nous expliquons notre méthodologie pour comparer les méthodes de détection de changement, puis nous présentons le site ChangeDetection.net (CDnet) où nous offrons gratuitement nos contributions à la communauté scientifique. Finalement, nous présentons quelques résultats obtenus en analysant les soumissions des scientifiques à travers le monde.

3.1 Banque de données

La banque de données CDnet contient 31 séquences vidéo réelles séparées en 6 catégories, totalisant 88 039 images, dont 65 064 (73.9%) sont annotées. Les vidéos ont été sélectionnées pour couvrir les défis importants en détection de changement et elles sont représentatives des données capturées de nos jours en vidéosurveillance. De plus, toutes les séquences ont au moins un objet en mouvement. Les séquences ont été acquises par 6 types de caméras différentes, ayant toutes des ajustements différents et un niveau de compression différent. La moitié des séquences ont une résolution de 320x240 et le reste est de taille supérieure,

jusqu'à un maximum de 720x576. Certaines vidéos capturées avec des caméras IP de basse qualité souffrent de distortion radiale. L'effet est visible sur les séquences « Parking » et « WinterDriveway » dans la figure 3.4. Il est aussi particulièrement apparent sur la séquence « StreetLight ». La position et l'orientation de la caméra, tout comme la taille des objets en mouvement, varient beaucoup d'une séquence à l'autre.

Cette diversité de caméras, de qualité et de taille des séquences peut être utile dans un banc d'essai car on ne peut prévoir si certains algorithmes excelleront sur une configuration particulière. Il est crucial d'éviter une sur-spécialisation des méthodes soumises sur le site. L'utilisation de multiples caméras, le nombre de vidéos et leurs difficultés inhérentes, la présence de mouvement parasite et la catégorie « Thermique » contribueront à la qualité de l'évaluation en diminuant nos chances d'être biaisés. Nous ne souhaitons pas une méthode qui excelle sur une séquence ou une catégorie particulière, nous recherchons un bon compromis : des méthodes offrant de bonnes performances sur un maximum de cas.

Ces particularités en font la plus grande banque de données en détection de changement au monde. Pour ce qui est des séquences réelles, nos plus proches compétiteurs ont moins de séquences et beaucoup moins d'images annotées. Les seuls compétiteurs qui ont davantage de séquences et d'images annotées ont des séquences semi-synthétiques ou synthétiques. Les problèmes occasionnés par ces dernières sont expliqués dans la section 2.1.1.

Malgré un pourcentage d'annotation inférieur à 100%, nous déclarons néanmoins que notre banque de données est complètement annotée. La raison étant que la plupart des algorithmes de détection de changement ont besoin d'une phase d'apprentissage au début de la vidéo pour apprendre l'arrière-plan. Chaque séquence est accompagnée d'un fichier nommé « temporalROI.txt » contenant deux chiffres : l'index de la première image annotée dans la séquence et l'index de la dernière image de la séquence. Toutes les images avant la première image annotée font partie de la phase d'apprentissage. Les autres images sont réellement utilisées lors des comparaisons. Donc, hormis les images d'apprentissage, toutes les images sont annotées. Quoi qu'il en soit, une vidéo de surveillance dure normalement plusieurs heures. Avoir une méthode de détection de changement ayant besoin de quelques secondes pour se stabiliser n'est donc pas un problème dans la plupart des cas. Toujours dans le but d'éviter les biais, nous avons choisi les tailles des phases d'apprentissage afin qu'elles soient différentes d'une vidéo à l'autre.

Cette banque de données est accessible gratuitement et simplement sur notre site Internet. La seule contrainte sur son utilisation consiste à nommer notre site Internet `http://changedetection.net/` et à référencer notre article [1] lorsque notre banque de données est utilisée dans le cadre d'une recherche académique ou commerciale.

3.1.1 Catégories et séquences

La banque de données CDnet est divisée est 6 catégories : basique, arrière-plan dynamique, mouvements de la caméra, mouvement intermittent des objets, ombres et thermique. Certaines de ces catégories sont directement associées aux difficultés présentées dans la section 1.5. Il est difficile de viser une seule difficulté lors de la capture d'une séquence. Lorsqu'elles visent réellement une difficulté unique, leur mise en scène a fréquemment l'air très planifiée et fausse. Sachant que nous voulons des vidéos d'apparence réaliste, nous évitons ce type de scènes. Les séquences sont donc influencées par d'autres défis que celui annoncé par leur catégorie, mais ce dernier est tout de même dominant.

On remarquera à la lecture des prochaines sections qu'il n'y a aucune catégorie qui traite du problème de camouflage (objet ayant la même couleur que l'arrière-plan). Malgré son importance en détection de changement, le camouflage n'a pas sa propre catégorie pour deux raisons. Premièrement, presque toutes les séquences réelles contiennent au moins un peu de camouflage car c'est un défi très présent en détection de changement. C'est d'ailleurs un problème uniformément répandu dans toutes les catégories. Par conséquent, une méthode robuste en regard du camouflage verra ses statistiques améliorées globalement. Deuxièmement, il est difficile de créer une catégorie contenant exclusivement du camouflage tout en évitant que les autres catégories en contiennent. Une catégorie de ce type aurait une mise en scène particulièrement fausse et peu convaincante. Comme il n'a pas sa propre catégorie, il est plus complexe de juger la robustesse des méthodes face au camouflage, mais nous verrons dans la section 3.4 qu'il est possible de le faire.

Basique

Comme le montre la figure 3.1, la catégorie « Basique » contient 4 séquences ; 2 capturées à l'intérieur et 2 à l'extérieur. Ces vidéos ne sont pas simplistes ou triviales à traiter, mais sont plutôt un mélange des défis représentés par les prochaines catégories. On y trouve à des divers degrés des ombres, des piétons qui bougent très peu pendant un certain temps, des arrière-plans dynamiques et d'autres défis standards. Ces vidéos sont prévues pour être assez faciles à traiter, sans toutefois être triviales.

Highway	Office	Pedestrians	PETS2006
1700 images | 320x240	2050 images | 360x240	1099 images | 360x240	1200 images | 720x576

FIGURE 3.1 – Les 4 séquences de la catégorie « Base ». Ces images et toutes celles qui suivent dans cette section sont rognées et redimensionnées à des fins de visualisation.

La séquence « PETS2006 » a été empruntée à la banque de données de la compétition de vidéosurveillance PETS. Nous n'avons pas capturé cette séquence, mais nous avons créé sa réalité de terrain et nous les offrons gratuitement à la communauté.

Arrière-plan dynamique

On peut voir dans la figure 3.2 les 6 séquences de cette catégorie. Ces dernières ont toutes été capturées à l'extérieur où il n'est pas rare de voir des mouvements parasites dans l'arrière-plan. Les objets en mouvement ont un déplacement continu afin d'éviter les problèmes de fantôme. De plus, l'angle, la position et la période de la journée ont été sélectionnées afin de minimiser les ombres. Ainsi, ces vidéos visent directement à tester la robustesse des méthodes de détection de changement face à l'arrière-plan dynamique. La

difficulté est représentée par le mouvement de l'eau dans des cours d'eau et des fontaines, puis par le mouvement des arbres balancés par le vent.

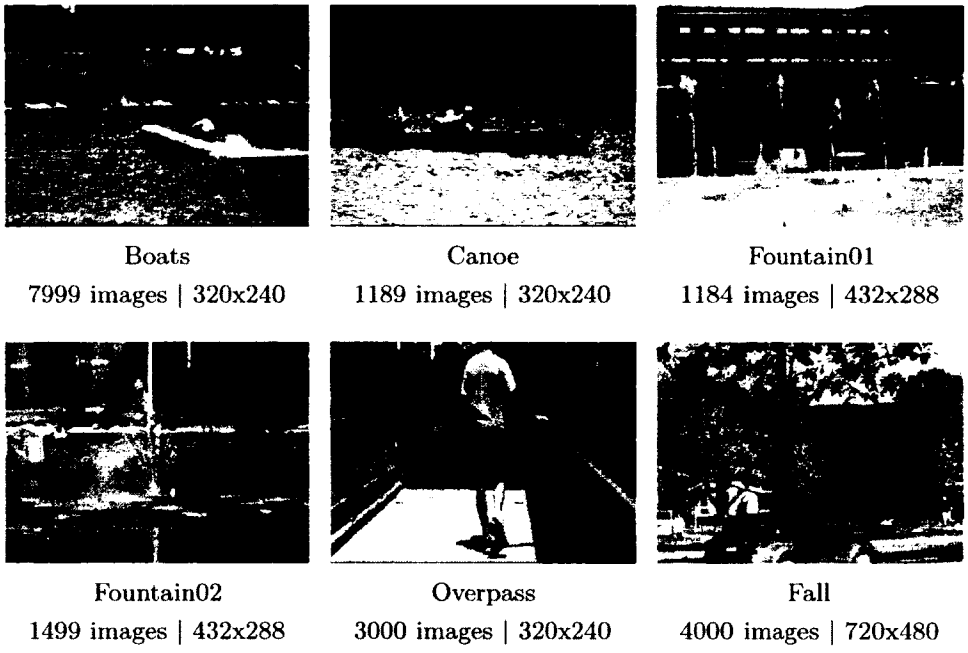
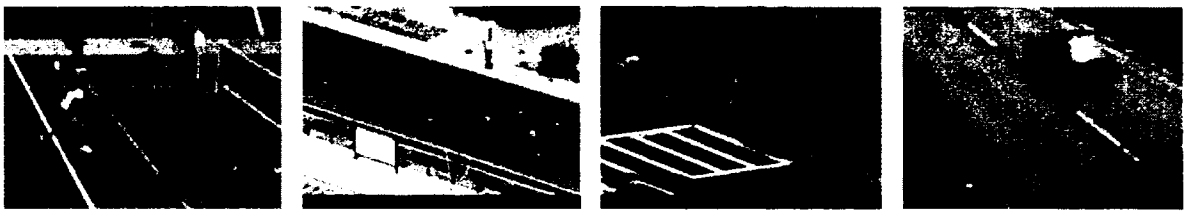


FIGURE 3.2 – Les 6 séquences de la catégorie « Arrière-plan dynamique ».

Mouvements de la caméra

Les caméras sont parfois instables ; une faible vibration peut occasionner un léger déplacement dans une direction aléatoire. Une vibration importante près de la caméra peut causer un déplacement tout aussi important. Certaines méthodes de détection de changement, de par le design de leur modèle, sont robustes envers de tels déplacements ; d'autres non. Cette catégorie comporte des déplacements d'un niveau entre faible et fort. Son but est de tester la robustesse des méthodes de détection de changement face aux mouvements de la caméra. Tel que montré par la figure 3.3, elle contient 4 séquences, dont une capturée à l'intérieur. Trois scènes montrent des voitures sur la route et deux se concentrent sur des personnes. Les vibrations sont régulières dans toutes les vidéos de cette

catégories, sauf dans la séquence « Traffic » où l'amplitude du mouvement de la caméra varie dans le temps.



Badminton	Boulevard	Sidewalk	Traffic
1150 images | 720x480	2500 images | 352x240	1200 images | 352x240	1570 images | 320x240

FIGURE 3.3 – Les 4 séquences de la catégorie « Mouvements de la caméra ».

Les séquences dans les autres catégories sont capturées avec une caméra totalement fixe ou, du moins, avec une caméra dont les vibrations sont imperceptibles. Cette difficulté est donc limitée à cette catégorie.

Mouvement intermittent des objets

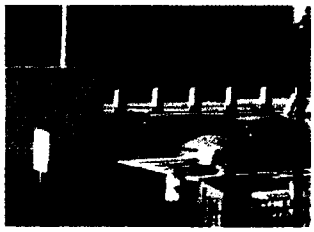
La définition du changement force la communauté à prendre quelques décisions difficiles. Tel qu'expliqué dans la section 1.5.1, le mouvement intermittent des objets peut causer l'apparition de fantômes dans les masques binaires. Visibles à la figure 3.4, les 6 séquences de cette catégorie permettent de tester la robustesse des méthodes de détection de changement face au mouvement intermittent des objets. Certaines montrent des objets bougeant pour la première fois, d'autres des objets abandonnées dont certains recommencent à bouger après un certain temps. Uniquement la séquence « Sofa » a été capturée à l'intérieur.

AbandonedBox
4500 images | 432x288

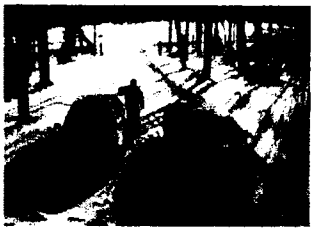
Parking
2500 images | 320x240

StreetLight
3200 images | 320x240

Sofa
2750 images | 320x240



Tramstop
3200 images | 432x288



WinterDriveway
2500 images | 320x240

FIGURE 3.4 – Les 6 séquences de la catégorie « Mouvement intermittent des objets ».

Ombres

Cette catégorie contient 2 séquences capturées à l'intérieur et 4 à l'extérieur. Comme le montre la figure 3.5, elle comporte des ombres douces et dures. Il y a aussi des ombres dans les autres catégories, par exemple dans la séquence « WinterDriveway », mais leur présence est ici mise de l'avant afin de tester majoritairement la robustesse des méthodes de détection de changement face aux ombres et aux changements d'illumination. Les ombres sont parfois projetées par des objets en mouvement, parfois par les arbres et les bâtiments. De plus, leur taille varie beaucoup entre les séquences ; elle peut être un fin segment dans une séquence, puis elle peut prendre une grande partie de la scène dans une autre, comme c'est le cas dans la séquence « Bungalows ».

La séquence « Cubicle » présente des ombres douces et dures, mais elle comporte aussi des changements d'illumination importants au niveau du mur et du tapis. En effet, l'intensité de l'éclairage naturel change fortement à certains moments de la vidéo.

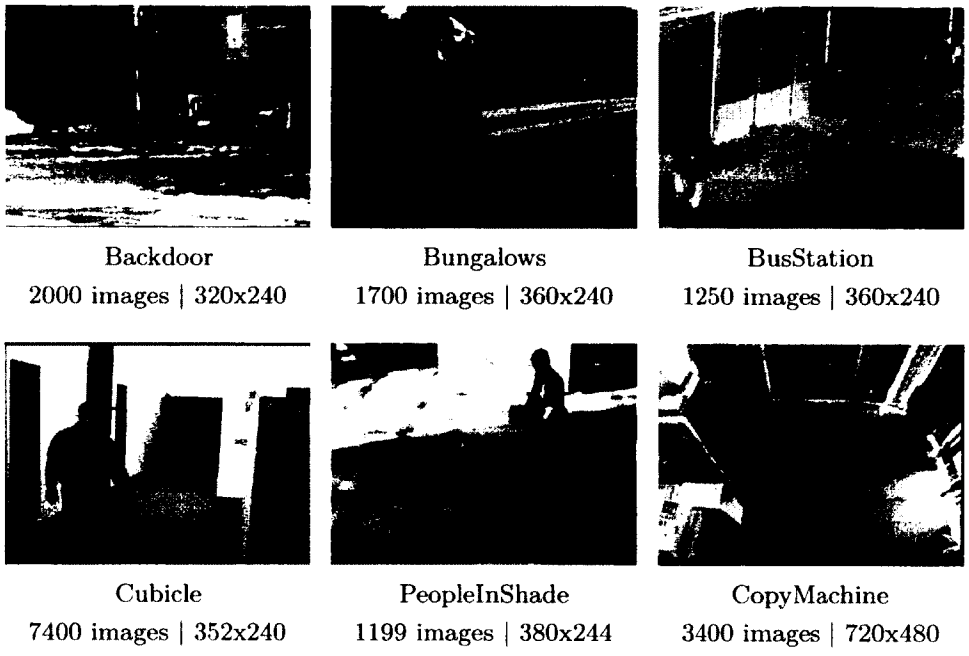


FIGURE 3.5 – Les 6 séquences de la catégorie « Ombre ».

Thermique

On trouve 5 vidéos dans cette catégorie, présentées dans la figure 3.6. Trois ont été filmées à l'intérieur et 2 à l'extérieur. Elles ont toutes été capturées par une caméra infrarouge. De ce fait, elles comportent des difficultées différentes de celles exprimées dans la section 1.5. Les difficultés d'analyse sont maintenant associées aux propriétés physiques de la chaleur et non pas à l'intensité et à la couleur. Parmi celles-ci, on retrouve :

empreintes thermiques : faux positifs causés par la zone brillante (chaude) laissée par des objets chauds (humains et animaux sur un divan par exemple) ;

reflets : faux négatifs causés par la réflexion des objets en mouvement (visible dans la séquence « Corridor » de la figure 3.6) ;

camouflage : faux négatifs causés par un objet en mouvement ayant la même température que l'arrière-plan.

Le camouflage et la réflexion existent aussi dans les vidéos capturées avec une caméra ordinaire, mais sont de nature différente.

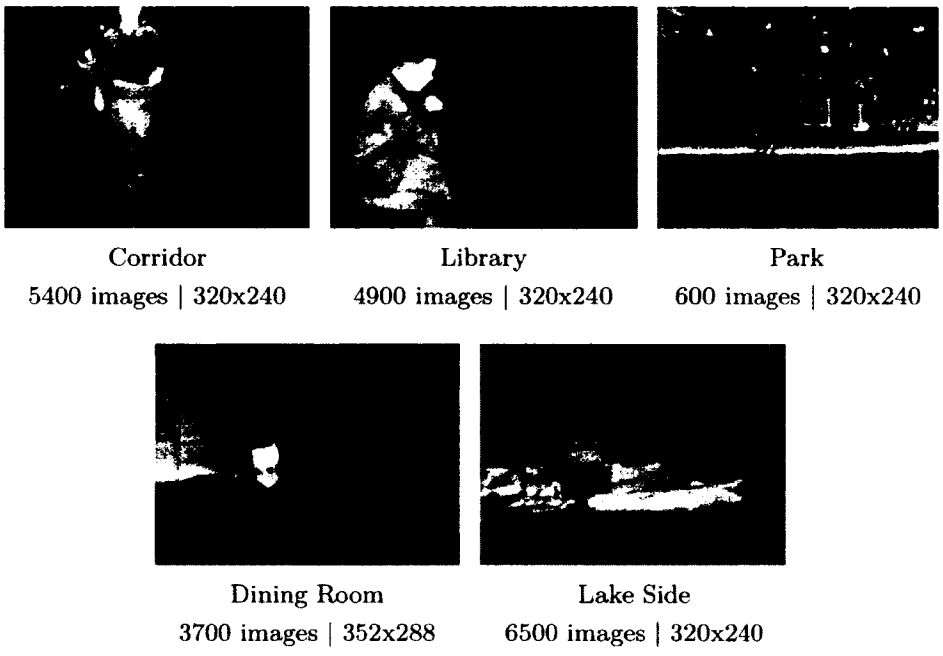


FIGURE 3.6 – Les 5 séquences de la catégorie « Thermique ».

3.1.2 Étiquettes

La configuration minimale à adopter pour comparer des méthodes de détection de changement requiert uniquement deux étiquettes, soit 0 « statique » et 1 « en mouvement ». Une telle configuration représente toutefois mal la réalité. Nous avons donc jugé nécessaire d'utiliser 5 étiquettes lors de la création des réalités de terrain, tel qu'illustré à la figure 3.7. Les méthodes soumises sur le site doivent toutefois en utiliser seulement deux : « statique » et « en mouvement ». Ce sont les deux seules étiquettes utilisées lors des comparaisons et les autres étiquettes seraient inutiles ou nuisibles pour les prochaines étapes d'analyse vidéo. Les étiquettes sont représentées par une valeur entre 0 et 255 afin de respecter les standards d'image 8 bits et afin d'être facilement visibles à l'oeil.

0 Statique : un pixel où il n'y a pas de mouvement important, tel que défini dans la section 1.3 ;

50 Ombre dure : un pixel dont l'intensité et la couleur ont été fortement influencés par l'ombre d'un objet en mouvement ;

85 Hors du ROI : un pixel hors de la région d'intéret ;

170 Inconnu : un pixel où il est difficile de déterminer s'il est statique ou en mouvement ;

255 En mouvement : un pixel associé à un objet en mouvement.

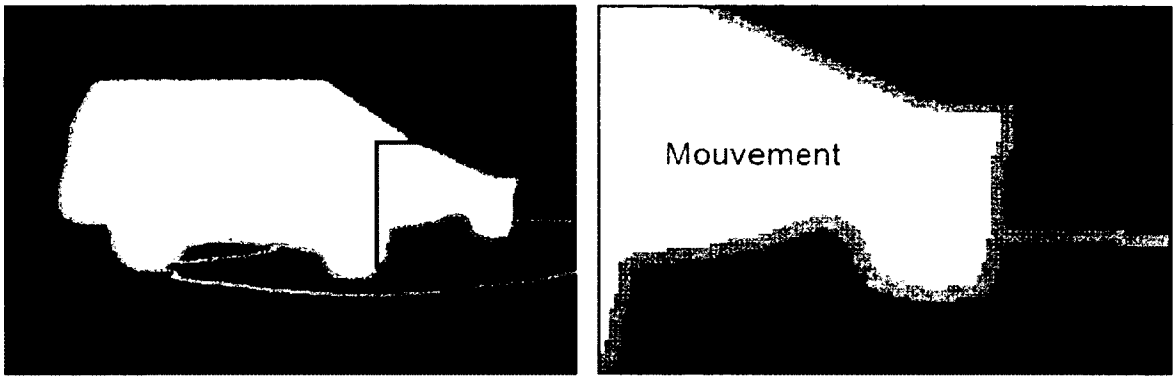


FIGURE 3.7 – Les 5 étiquettes utilisées dans la banque de données CDnet. Une image de la réalité de terrain de la séquence « Bungalows ».

Les régions d'intérêt, ROI, sont un outil utile pour contrôler la difficulté d'une séquence. Un fichier « ROI.bmp » accompagne chaque séquence afin de spécifier quelles zones sont valides et quelles ne le sont pas. Comme le montre la figure 3.8, il est possible de spécifier des zones où les résultats ne sont pas pertinents pour la catégorie en cours. Ceci nous permet de rendre une difficulté dominante dans une vidéo où elle ne l'est pas. Nous pourrions, par exemple, ignorer les mouvements parasites causés par le vent dans les arbres dans une séquence de la catégorie « Ombre ». Davantage de détails sur les étiquettes seront donnés dans la section 3.1.2.

Les pixels « Hors du ROI » et « Inconnu » ne sont pas comptabilisés lors des comparaisons. Dans le premier cas, c'est simplement parce qu'ils sont hors de la région d'intérêt. Ceci peut être causé par le fichier « temporalROI.txt » ou « ROI.bmp ». Dans l'autre cas, c'est parce qu'il est difficile de déterminer la vraie nature de ces pixels. Comme le montre la figure 3.9, les pixels bordant un objet en mouvement ont souvent une nature imprécise, surtout si l'objet se déplace rapidement (effet de « *motion blur* »). Ceci est une conséquence du mode d'acquisition des images et de la qualité des images. Cette bordure est en moyenne large de trois pixels, quelles que soient les propriétés des objets en mouvement. Ce chiffre

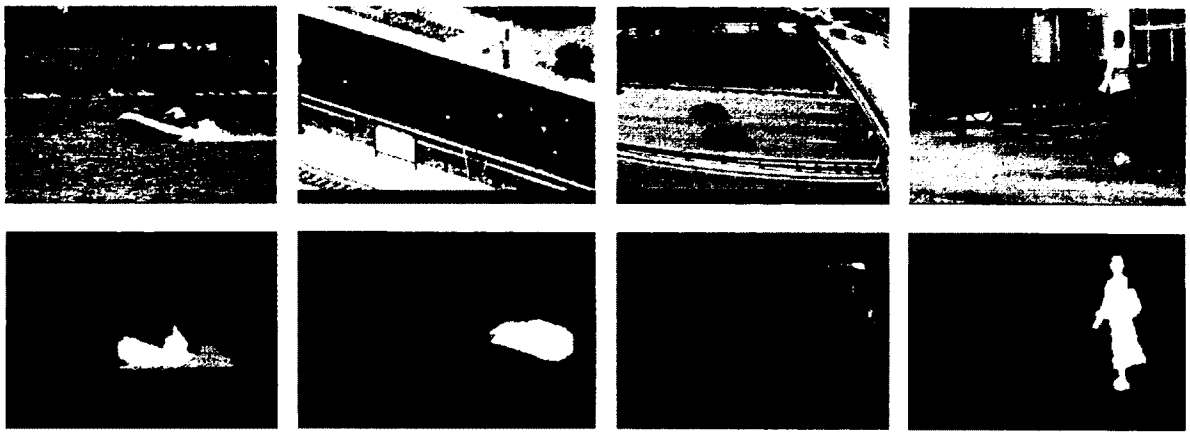


FIGURE 3.8 – Les zones gris foncé, visibles dans la deuxième rangée, seront ignorées lors de la comparaison. Ceci permet d'éviter les difficultés non pertinentes dans certaines catégories.

nous semblait adéquat pour la plupart des objets, mais il peut s'avérer un mauvais choix lorsque les séquences sont de grande taille ou lorsque les objets bougent très rapidement. L'automobile dans la figure 3.9 en est un bon exemple ; une bordure de 5, voire 7 pixels aurait été plus apte à gérer le flou.

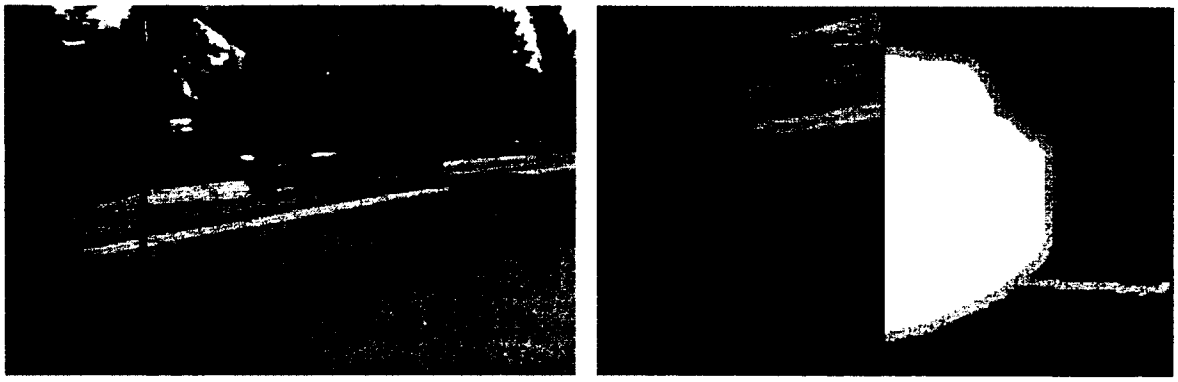


FIGURE 3.9 – L'étiquette « Inconnu » permet d'ignorer la bordure des objets en mouvement et tout autre pixel dont on ne connaît pas la nature réelle.

Un pixel classifié comme « Ombre dure » sera toutefois comptabilisé comme un pixel statique. Tel qu'expliqué précédemment, les ombres sont une des difficultés en détection de changement. L'existence de cette étiquette est surtout utile pour déterminer si les méthodes de détection de changement sont robustes en regard des ombres, un résultat qui sera expliqué dans la section 3.4.

3.1.3 Outil de segmentation

Annoter un grand nombre d'images requiert l'automatisation de certaines tâches. La création d'un programme d'annotation s'avère nécessaire et permet de sauver des semaines de travail. Sans outil de segmentation adapté, il faut charger chaque image dans un programme d'édition d'images, créer un nouveau calque, réorganiser l'ordre des calques afin que le nouveau calque soit au premier rang, modifier la transparence de ce calque, dessiner en blanc les objets en mouvement, replacer le niveau de transparence du calque et, finalement, sauvegarder cette nouvelle image en choisissant un nom significatif. Ensemble, ces tâches demandent quelques minutes de travail pour une seule image.

Un programme de segmentation permet d'automatiser ces tâches, sauf celle où il faut dessiner les objets en mouvement. Cette tâche peut être automatisée en partie, mais des retouches sont toutefois nécessaires afin d'obtenir des réalités de terrain de qualité. Pour le reste, le programme s'occupe de tout ; que ce soit l'ouverture et la création des images, la gestion des calques ou la sauvegarde des fichiers.

L'interface graphique du programme est présentée à la figure 3.10. La partie gauche permet de voir une ébauche de segmentation, fournie préalablement par l'utilisateur en utilisant une méthode de détection de changement. Cette dernière est parfois utile pour initialiser la segmentation, mais peut aussi être nuisible si l'ébauche est de mauvaise qualité. La partie droite permet de voir la segmentation finale.

Avec ce programme, la création d'une réalité de terrain se résume à changer d'image en appuyant sur la flèche droite et corriger l'annotation automatique au besoin. Au passage à la prochaine image, l'image actuelle est modifiée et sauvegardée. Tout le reste est géré automatiquement ; ceci inclut l'ajout de la bordure et l'application de la région d'intérêt, tous deux étiquetés « inconnu ».

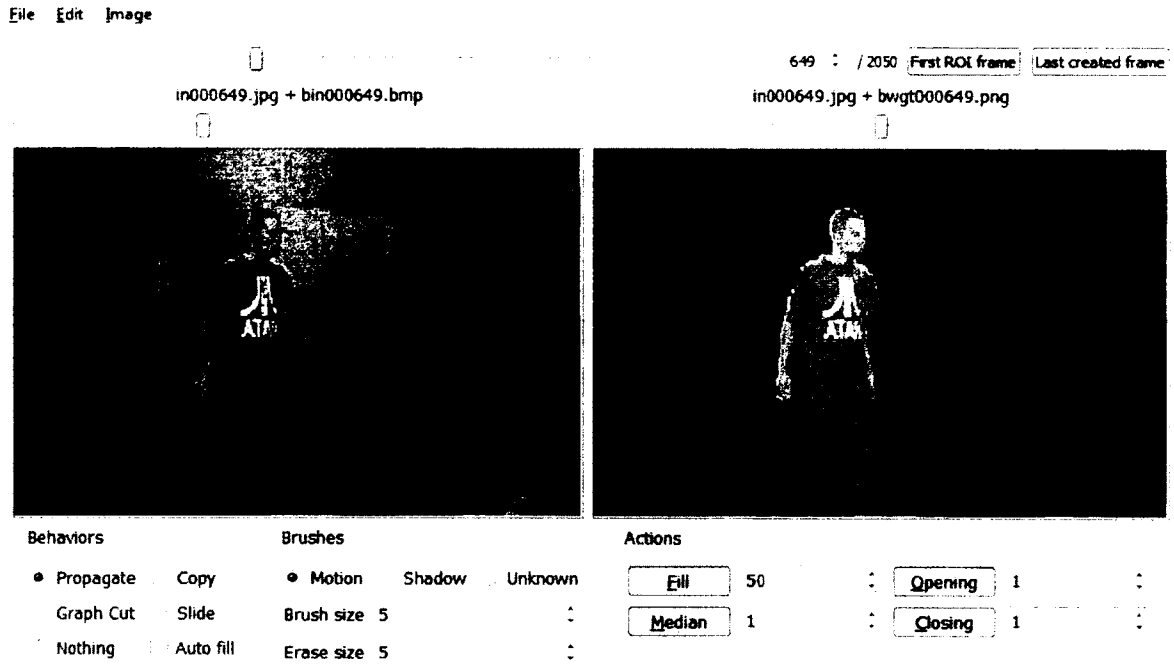


FIGURE 3.10 – Interface graphique du *GroundTruther*, le programme de segmentation utilisé lors de la création de la banque de données CDnet. Construction de la réalité de terrain de la séquence « Office ».

L'annotation automatique dépend du mode choisi. S'il n'y a aucun mouvement, les modes *Nothing* ou *Copy* sont de bons choix car la réalité de terrain sera simplement statique. Si les objets en mouvement bougent très peu durant quelques images, *Copy* est un bon choix car il ne fait que copier la réalité de terrain précédente. Lorsque les objets bougent à une vitesse normale, *Propagate* et *Graph Cut* sont les meilleurs choix. Si l'ébauche est de bonne qualité, *Propagate* utilise la réalité de terrain précédente et l'ébauche en cours pour créer une réalité de terrain qui n'aura peut-être pas besoin de retouche. Lorsque l'ébauche n'est pas de bonne qualité, il vaut mieux utiliser *Graph Cut*. La réalité de terrain de la dernière image est utilisée pour trouver des pixels assurément statiques et des pixels assurément en mouvement. Avec ces derniers, nous pouvons appliquer l'algorithme *Max-flow/min-cut* programmé par Yuri Boykov [1] [61].

1. `http://vision.csd.uwo.ca/code/`

Il y a aussi d'autres outils pratiques offerts dans le *GroundTruther*. Les filtres morphologiques en sont de bons exemples. La possibilité de copier-coller des zones en mouvement est aussi très pratique pour des objets qui ne changent pas de forme.

Ce programme et sa source sont disponibles gratuitement et simplement sur Internet [2]. Nous espérons qu'il convaincra les scientifiques à travers le monde de contribuer aux futurs développements de notre banque de données en fournissant des séquences et leur annotation. À défaut d'un tel enthousiasme, nous espérons qu'il sera utile à la communauté.

3.2 Évaluation

La méthodologie d'évaluation est tout aussi importante que la qualité de notre banque de données. Pour devenir le premier choix de la communauté de détection de changement, il est crucial de bien choisir nos mesures. Rappelons-nous qu'aucun système n'est irréprochable et qu'aucune mesure ne fait l'unanimité en raison de la diversité des applications d'analyse vidéo. Cependant, nous pouvons au moins choisir des mesures qui éviteront les biais autant que possible.

Nous avons vu dans la section 2.2 que les courbes Précision-Rappel sont fréquemment utilisées lors des évaluations. Il aurait alors été intéressant d'offrir ce type de courbes pour les méthodes soumises, mais il y a un point important à prendre en considération. Nous ne pouvons pas demander tout ce que nous désirons aux chercheurs. Il serait pratique d'avoir la version officielle du code des auteurs, mais ce n'est pas une demande réaliste. Demander aux auteurs d'exécuter leur méthode une vingtaine de fois avec différents seuils n'est pas non plus une demande réaliste, tant du point de vue des délais de calcul que de la faisabilité. Nous avons donc laissé tomber les courbes PR pour les mesures classiques en classification. Quoi qu'il en soit, les courbes PR pourraient difficilement être utilisées seules pour déterminer la meilleure méthode. Les courbes sont parfois trop collées ou à une distance similaire du score parfait ; établir un rang devient alors une tâche subjective. Nous prévoyons avoir plusieurs dizaines, voire des centaines de méthodes si notre projet fonctionne alors les courbes PR ne semblaient pas être le bon outil.

2. `https://bitbucket.org/nilgoyyou/groundtruther`

Bien que le temps d'exécution et la consommation mémoire sont parfois des critères importants pour la communauté de détection de changement, ces derniers ne sont pas pris en compte lors de l'évaluation. Le problème est le même que pour les courbes PR : nous ne pouvons pas demander aux chercheurs l'implémentation officielle de leurs méthodes. N'ayant pas leur version officielle, il est impossible d'estimer l'espace mémoire requis et le temps d'exécution. Quoi qu'il en soit, ces critères sont dépendants du langage de programmation choisi et de l'ordinateur sur lequel la méthode s'exécute ; il est alors peu utile de fournir ces renseignements. Nous avons néanmoins jugé pertinent de demander aux chercheurs de spécifier le nombre d'images par seconde et de décrire l'ordinateur, le langage et la taille de la vidéo utilisée. Comme nous n'avons aucun contrôle sur la qualité et la véracité de leurs réponses, ces informations sont données uniquement à titre informatif.

La f-mesure est parfois considérée comme une bonne mesure unique. Elle requiert autant une bonne précision qu'un bon rappel, donc un minimum de pixels mal classifiés. De ce fait, elle n'avantage, ni ne désavantage aucune méthode, qu'elles sous-estiment ou surestiment le nombre de pixels en mouvement. Nous avons donc jugé pertinent d'utiliser cette dernière.

Nous nous doutons bien que la f-mesure ne plaira pas à tous les chercheurs de la communauté de détection de changement. Il convient alors d'ajouter d'autres mesures sans biaiser le classement final. Parmi les mesures neutres, nous avons ajouté le pourcentage de mauvaises classifications. Pour ce qui est des mesures qui avantagent les méthodes qui sous-estiment le nombre de pixels en mouvement, nous avons ajouté la précision, la spécificité et le taux de faux négatifs. Pour celles qui surestiment, nous avons ajouté le rappel et le taux de faux positifs. Au final, nous utilisons les mesures du tableau 3.1.

TABLE 3.1 – Biais selon les mesures de classification

Sous-estime	Neutre	Surestime
Précision		
Spécificité		Taux de FN
Taux de FP		Rappel
	F-mesure	
	PMC	

Étant des opposés, 4 de ces mesures s'annulent, soient Sp et T_{FP}, et Ra et T_{FN}. Il reste alors la précision qui avantage légèrement les méthodes qui sous-estiment le nombre de pixels en mouvement. La mesure manquante à ajouter pour un système parfaitement neutre est le taux de vrais négatifs. Bien que ce système soit intéressant au niveau de l'impartialité, nous avons jugé préférable de laisser un léger biais puisque, en détection de changement, les faux positifs représentent souvent un problème plus grave que les faux négatifs [87]. Ceci est dû au fait que le nombre de VP est beaucoup plus bas que le nombre de VN et que les FP ont des conséquences plus graves sur les autres étapes de traitement en analyse vidéo.

L'ajout de l'étiquette « Ombre » à la banque de données CDnet [1] nous permet d'ajouter une mesure spécialisée pour valider la robustesse des méthodes face à l'ombre. Nous calculons le taux de faux positifs sur tous les pixels classifiés en ombre dure dans les réalités de terrain, T_{FPO}. Cette mesure n'influence pas directement le classement final ; elle est plutôt informative. Tel qu'expliqué dans la section 1.5.6, l'ombre n'est pas un objet en mouvement et devrait être classifié « Statique ». Cette mesure nous permet donc d'évaluer si une méthode reconnaît les ombres dures en tant qu'objets en mouvement ou en tant qu'arrière-plan, comme elle le devrait.

Ces métriques sont calculées pour chaque méthode sur chaque vidéo de chaque catégorie fournie par les chercheurs. Par exemple, le rappel d'une vidéo v de la catégorie c est calculé comme suit :

$$Ra_{c,v} = \frac{VP_{c,v}}{VP_{c,v} + FN_{c,v}}. \tag{3.1}$$

Une moyenne pour chaque catégorie est ensuite calculée avec les métriques de toutes les séquences de cette catégorie. Par exemple, le rappel d'une vidéo v de la catégorie c est calculé ainsi :

$$Ra_c = \frac{1}{|N_c|} \sum_v Ra_{c,v} \tag{3.2}$$

où $|N_c|$ est le nombre de vidéos dans la catégorie c. Le taux de faux positifs sur l'ombre, T_{FPO}, est calculé uniquement sur la catégorie « Ombre » car c'est la seule catégorie où cette mesure est pertinente et où les ombres sont prédominantes.

Une moyenne globale est ensuite calculée pour inclure toutes les catégories de la banque

de donnée. Par exemple, le rappel global est calculé ainsi :

$$Ra = \frac{1}{6} \sum_{c} Ra_c. \tag{3.3}$$

Comme on peut le voir à la figure 3.11, des calculs similaires sont effectués pour les 6 autres métriques utilisées dans ce projet.

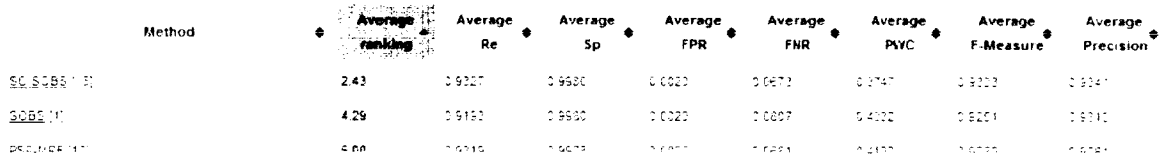


FIGURE 3.11 – Extrait du tableau des résultats pour la catégorie « Basique », pris sur le site [1]. a) La liste des méthodes. b) Classement des méthodes selon la catégorie en cours, calculé selon l'équation 3.4. c) Moyennes des différentes métriques calculées sur les séquences de la catégorie en cours, calculées selon l'équation 3.2.

Certaines vidéos comportant davantage d'images et une plus grande résolution, un système d'évaluation pourrait alors donner davantage de poids à ces vidéos. Additionner les valeurs de la matrice de confusion (VP, FP, FN et VN) entre les vidéos d'une même catégorie, par exemple, donnerait une énorme importance aux séquences de grande taille. Les moyennages des métriques expliquées dans cette section permettent d'éviter ce problème : les vidéos et les catégories ont un poids identique lors des calculs.

Nous avons maintenant des calculs individuels et globaux pour les vidéos, les catégories et pour la banque de données, mais nous n'avons pas encore de stratégie pour déterminer le classement des méthodes. Il est alors utile de fournir une mesure indiquant la qualité générale d'une méthode vis-à-vis des autres méthodes en combinant les performances des différentes mesures calculées précédemment. Cette mesure peut s'appliquer autant sur les catégories que sur la banque de données globalement. Pour ce faire, nous utilisons l'approche de Young et Ferryman [64] tirée de leur projet *PETS Metrics*, qui consiste à calculer le classement moyen par catégorie Cc et le classement moyen global C, le tout

en utilisant les moyennes calculées précédemment. Le classement moyen d'une méthode i pour une catégorie c est calculé ainsi :

$$Cc_{i,c} = \frac{1}{7} \sum_{m} rang_i(m, c) \tag{3.4}$$

où $rang_i \in \mathbb{N}$ et m désignent une des 7 mesures calculées avec l'équation 3.2. Il consiste donc à faire la moyenne des rangs pour toutes les métriques par rapport aux autres méthodes. Par exemple, si une méthode obtient un classement $Cc_{i,c} = 3.0$, cela signifie que cette méthode s'est classée au 3^e rang des métriques en moyenne.

Le rang global pour une méthode i est calculé ainsi :

$$C_i = \frac{1}{6} \sum_{c} Cc_{i,c}. \tag{3.5}$$

Les métriques moyennes globales pour une méthode i à travers toutes les catégories sont calculées ainsi :

$$Cg_i = \frac{1}{7} \sum_{m'} rang_i(m') \tag{3.6}$$

où m' est une des moyennes globales pour les métriques, telles que celles calculées dans l'équation 3.3. La figure 3.12 montre l'utilisation de ces équations sur le site [1].

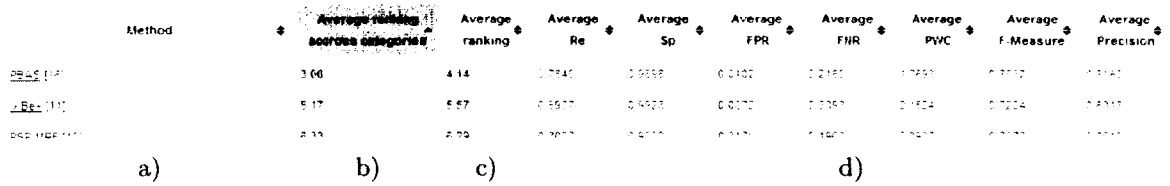


FIGURE 3.12 – Extrait du tableau des résultats globaux, pris sur le site [1]. a) La liste des méthodes. b) Classement global des méthodes, calculé selon l'équation 3.5. c) Classement des méthodes selon les moyennes calculées en d), calculé selon l'équation 3.6. d) Moyennes globales des différentes métriques, calculées selon l'équation 3.3.

Ces calculs divisés par catégorie et rapportés sur la banque de données globalement permettent d'obtenir un aperçu rapide des meilleures méthodes en ce moment. Une inspection des résultats dans une catégorie particulière permet de choisir une méthode

selon un besoin particulier. Les résultats globaux, quant à eux, permettent de choisir la meilleure méthode, toutes difficultés confondues.

Si, pour une raison ou pour une autre, notre méthode d'évaluation ne plaît pas à la communauté de détection de changement, le projet n'est pas inutile pour autant. Premièrement, il est aisé de changer la méthode d'évaluation en fonction sur le site web si tel est le désir de la communauté. Deuxièmement, nous savons que ces mesures ne plairont pas à tous, c'est pourquoi nous en avons choisi autant. De cette façon, les chercheurs peuvent ignorer ce qu'ils jugent impertinent. De plus, les données brutes sont disponibles pour toutes les méthodes. D'autres chercheurs peuvent alors télécharger ces données et utiliser une méthode d'évaluation adéquate selon leurs besoins. C'est d'ailleurs ce que le projet ViSOR [72] offrait ; il permettait de comparer des méthodes selon différents critères sur un ensemble de résultats fixes.

3.3 ChangeDetection.net

Ce site[3] est notre solution aux problèmes énoncés dans la section 1.7. Accessible à tous, il se veut la référence en matière de comparaison des méthodes de détection de changement. Il est présentement divisé en trois sections importantes : les pages pour l'ajout d'une méthode, la page de visualisation des résultats et une page archivée concernant la compétition à CVPR 2012, expliquée dans la section 3.3.4.

3.3.1 Création d'une méthode

Créer une nouvelle méthode sur le site demande au minimum de télécharger la banque de données CDnet, exécuter la méthode sur les séquences de CDnet, puis téléverser les masques binaires sur le site. Il est aussi possible de télécharger des outils et de la documentation afin de mieux comprendre les actions à effectuer.

Dans la page « DATASET, » il est possible de télécharger une partie ou la totalité de la banque de données CDnet. Dans le haut de la page, quelques informations sont fournies

3. `http://www.changedetection.net/`

sur la banque de données. Ensuite, il est possible de télécharger toute la banque de données sous deux formats compressés, soient zip et 7z. Comme on peut le voir dans la figure 3.13, il est aussi possible de télécharger une partie de la banque de données, individuellement par catégorie ou par vidéo. Il est définitivement plus pratique de télécharger la totalité de la banque de données, mais un téléchargement par parties pourrait être utile pour certains usagers disposant d'une mauvaise connexion Internet ou pour une vérification rapide.

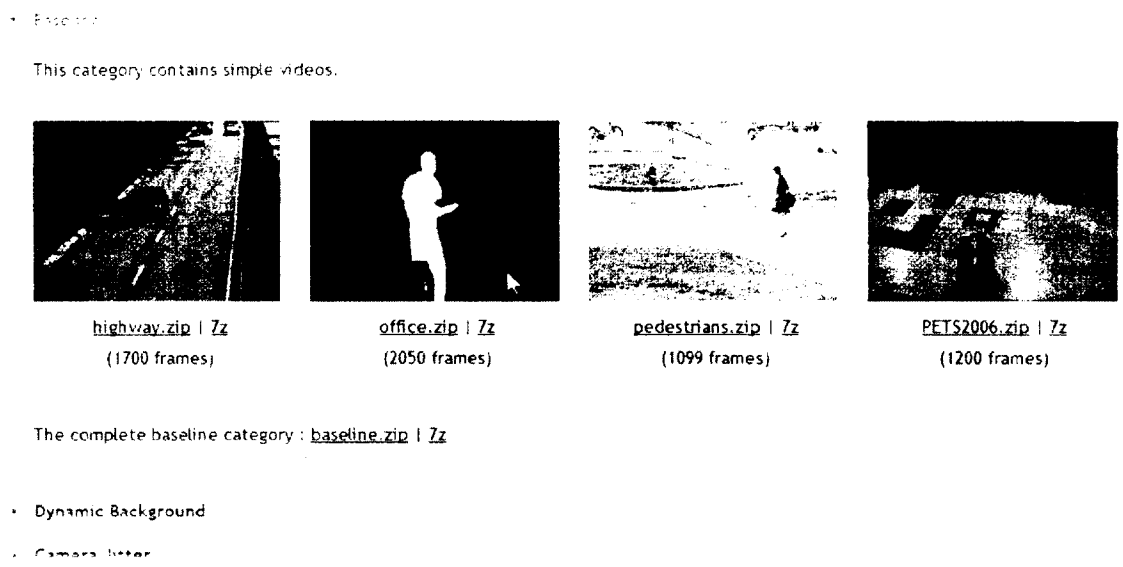


FIGURE 3.13 – Capture d'écran de la page « DATASET » où il est possible de télécharger la totalité ou une partie de la banque de données.

Comme le montre la figure 3.14, la page « UTILITIES » donne quelques informations sur ce que les méthodes doivent produire pour être acceptées sur le site. De plus, elle permet de télécharger des outils pour travailler avec la banque de données CDnet. La plupart des outils sont offerts en version Python[4] et C++ ou en version Matlab[5]. Offrant un outil commercial et un outil libre et gratuit, nous permettons à tous les chercheurs d'utiliser nos outils.

Nous offrons avant tout un gabarit d'une méthode de détection de changement. Ce dernier boucle sur les catégories et les séquences en exécutant une méthode laissée vide

4. `http://www.python.org/`
5. `http://www.mathworks.com/products/matlab/`

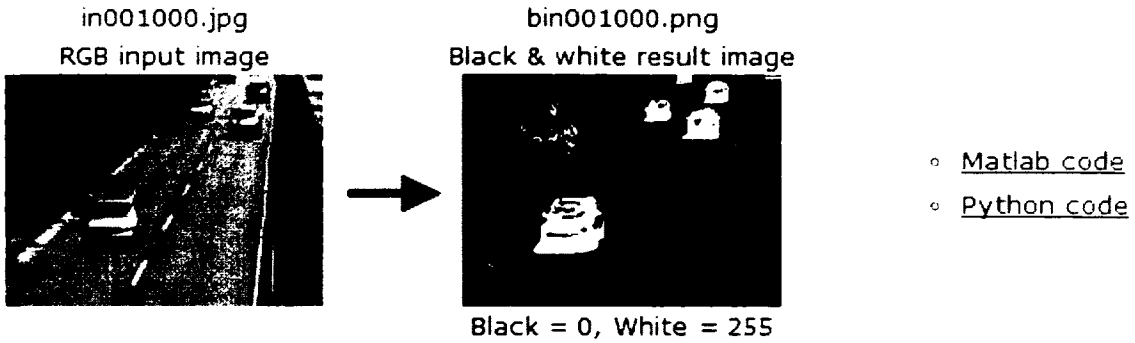


FIGURE 3.14 – Capture d'écran de la page « UTILITIES » où il est possible d'obtenir des informations et de télécharger des outils pour travailler sur la banque de données CDnet.

afin que les chercheurs puissent écrire leur code. La version Python appelle simplement un programme C++\OpenCV [6] avec les bons paramètres. La version Matlab utilise la *image processing toolbox* pour offrir les mêmes fonctionnalités.

Ce programme est normalement accompagné de l'outil de comparaison. Offert autant en Python qu'en Matlab, il compare toutes les images pixel à pixel avec les réalités de terrain afin de calculer les chiffres de la matrice de confusion et les autres mesures de classification expliquées dans la section 3.2. Ce programme crée un fichier de résultats « results.txt » en suivant les mêmes règles appliquées sur le site. Ces fichiers contiennent un bloc par catégorie et un bloc global, le tout en 5 colonnes de chiffres : VP, FP, FN, VN et le nombre de pixels ombragés mal classifiés. Ils ont la forme suivante :

```
cm video baseline highway 5220722 364651 237267 86289818 361263
cm video baseline PETS2006 4146397 574522 681793 366138782 0
cm video baseline pedestrians 640306 38050 30556 67531057 12502
cm video baseline office 8306975 173965 335119 116364721 0
cm category baseline 18314400 1151188 1284735 636324378 373765
...
cm overall 206147312 158636597 57525824 5758648629 4837894
```

Nous offrons aussi un programme Python permettant de calculer le classement de plusieurs méthodes. Il prend en entrée plusieurs fichiers de résultats pour ensuite donner

6. http://opencv.willowgarage.com/wiki/

le rang des méthodes. Cet outil peut être pratique pour un chercheur voulant comparer différents paramètres pour sa méthode avant de la soumettre.

Une fois les masques binaires générés avec une méthode et compressés en un fichier unique, le chercheur se dirige vers la page « UPLOAD » où il peut entrer les informations sur sa méthode et les détails pour le contacter. La taille du fichier compressé varie entre 100Mo et 200Mo, dépendant en grande partie du nombre de faux positifs car une image se compresse mieux lorsqu'elle a de grandes zones de couleur uniforme.

Lorsque la méthode est soumise sur le site, nous validons les résultats afin d'éviter la triche et les erreurs de soumission. Lorsqu'un problème est détecté, le chercheur reçoit un courriel contenant un message expliquant la raison du rejet de sa soumission.

3.3.2 Résultats

Lorsque nous recevons une soumission, nous décompressons le fichier envoyé par les chercheurs afin de le comparer avec les réalités de terrain. Les métriques introduites dans la section précédente sont ensuite enregistrées dans la base de données afin d'être visualisées plus tard. Le chercheur reçoit alors un message lui permettant de voir ses résultats et son classement parmi les autres méthodes présentes sur le site. Cette nouvelle méthode est affichée sur le site uniquement lorsque le chercheur le demande. Une fois la méthode visible à tous, elle apparaît dans la page « RESULTS » dans toutes les catégories où des résultats ont été soumis. Elle sera aussi visible dans la section globale si des résultats ont été soumis pour toutes les catégories. Comme le montre la figure 3.15, il est possible d'obtenir les informations sur une méthode particulière en cliquant sur son nom.

Tous les résultats sont visibles dans la page « RESULTS ». L'onglet ouvert par défaut présente les résultats globaux, donc le classement et les métriques de chaque méthode (voir figure 3.16.) Les 6 prochains onglets présentent les résultats par catégorie. Le dernier onglet contient les courbes ROC de nos 9 premières méthodes (voir section 3.3.4). Cette page contient aussi des explications sur les mesures utilisées et les références vers les méthodes soumises sur le site.

La seule contrainte à respecter pour apparaître dans cette section est d'utiliser un seul jeu de paramètres pour toutes les catégories et vidéos lors de l'analyse. Les paramètres

Contact name	Lucia Maddalena
Contact email	lucia.maddalena@na.icar.cnr.it
Contact university/company	National Research Center of Italy
Method's name	SC-SOBS
Reference	L. Maddalena, A. Petrosino, "The SOBS algorithm: what are the limits?", in proc of IEEE Workshop on Change Detection, CVPR 2012
Processing time	~23fps for 320x240 video, ~4fps for 720x576 video, C code on Core i3-330M 2.13GHz
Web page	http://cvprlab.uniparthenope.it/
Parameters	none (all defaults)

You can download the change detection results of this method by clicking here.

FIGURE 3.15 – Capture d'écran du site [1] où il est possible de voir les informations sur la méthode et sur le scientifique à contacter pour toutes questions relatives à cette méthode.

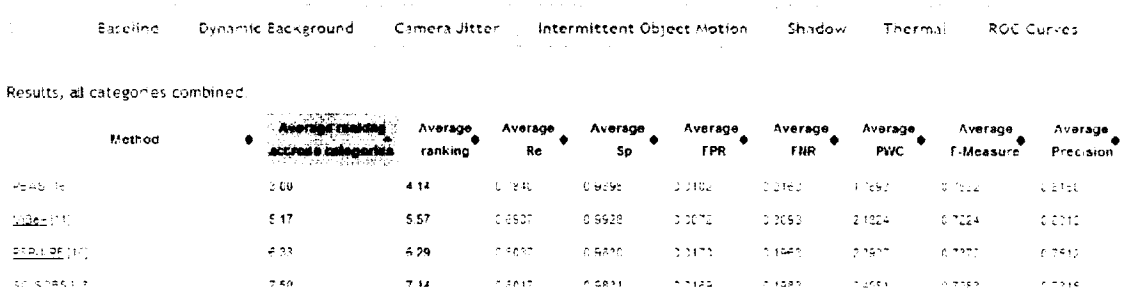


FIGURE 3.16 – Capture d'écran de résultats du site [1]. Les méthodes, leur classement et leurs métriques y sont visibles selon différentes catégories ou globalement.

choisis sont d'ailleurs visibles dans la page d'informations sur la méthode, tel que montré à la figure 3.15.

3.3.3 Discussion

Résumons rapidement la problématique actuelle pour la comparaison des méthodes de détection de changement :

- il n'existe aucune banque de données standard,
- il est difficile de reproduire les résultats des autres méthodes,
- aucune métrique d'évaluation n'a été retenue par la communauté.

À la lecture des derniers paragraphes, nous voyons que notre solution répond adéquatement à ces problèmes. Dans le premier cas, nous ne pouvons forcer la communauté à utiliser notre banque de données, mais nous avons mis tous les efforts nécessaires pour qu'elle devienne peu à peu le standard en détection de changement. Si notre projet est réellement adopté par la communauté scientifique, il va corriger par le fait même les deux autres problèmes.

Pour ce qui est de la difficulté de reproduire les résultats de l'état de l'art, ce n'est plus nécessaire car, si tout fonctionne comme nous l'espérons, ils seront tous disponibles sur le site. Tous les résultats téléversés sur le site sont disponibles en téléchargement peu de temps après. Les résultats des mesures de classification sont bien sûr disponibles, mais aussi l'archive soumise par les chercheurs. Un chercheur peut alors télécharger les images résultant d'une autre méthode et les insérer dans son article à des fins de comparaison. Il sera donc aussi simple de se comparer aux méthodes « simples » qu'aux meilleures méthodes du moment. Nul besoin de programmer la méthode du compétiteur ni de chercher le code source, il suffit de télécharger les résultats officiels de cette méthode.

Dans l'éventualité où notre projet devient le standard *de facto* en détection de changement, les comparaisons seront faites sur la même base. Tel qu'expliqué dans la section précédente, nous offrons une méthode d'évaluation qui se veut juste et non biaisée. De plus, elle permet aux chercheurs d'établir leur propre classement en utilisant les données sur le site, ce qui peut être utile dépendamment des besoins de leurs projets.

3.3.4 Compétition CVPR2012

La conclusion de ce projet dépend directement de la participation de la communauté ; il ne peut fonctionner sans publicité et sans aide. C'est pourquoi nous avons pris quelques mesures afin d'encourager la communauté scientique à adopter notre projet.

La première étape fut d'ajouter nous-mêmes quelques méthodes sur le site. Les méthodes choisies sont représentatives des grandes familles en détection de changement. Nous avons sélectionné 9 méthodes, calculé les masques binaires avec ces dernières et téléversé les résultats sur le site. Les méthodes sélectionnées sont les suivantes :

Distance euclidienne [58] : notre implémentation ;

Distance de Mahalanobis [58] : notre implémentation ;

GMM Stauffer-Grimson : implémentation de la célèbre méthode de Stauffer et Grimson [33], programmée par Donovan Parks [7] ;

GMM Zivkovic : implémentation de la méthode de Zivkovic [88], programmée par Donovan Parks ;

GMM KaewTraKulPong : implémentation de la méthode de KaewTraKulPong et Bowden [31], fournie par le projet libre OpenCV ;

KDE : notre implémentation de la méthode d'Elgammal *et al.* [35] ;

ViBe : implémentation officielle [8] de la méthode de Barnich et Van Droogenbroeck [41] ;

KNN : implémentation officielle [9] de la méthode de Zivkovic et van der Heijden [88] ;

SOBS : implémentation officielle [10] de la méthode de Maddalena et Petrosino [43].

Ensuite, nous avons organisé un atelier (*workshop*) et une compétition de détection de changement à la conférence CVPR 2012 [11]. Organisée par Pierre-Marc Jodoin [12], Fatih Porikli [13], Janusz Konrad [14] et Prakash Ishwar [15], le but était triple : faire connaître le site, ajouter des méthodes à la page de résultats et discuter directement avec une partie de la communauté de détection de changement. Jouissant d'une excellente notoriété dans le milieu académique, Fatih Porikli s'est chargé entre autre de publiciser le site et le projet. Au final, cette compétition nous a permis d'ajouter 11 méthodes sur le site :

PBAS : Hofmann *et al.* [89] ;

Chebyshev probability approach : Morde *et al.* [90] ;

Chebyshev probability with static object detection : Morde *et al.* [90] ;

SC-SOBS : Maddalena et Petrosino [44] ;

PSP-MRF : Schick *et al.* [91] ;

7. `http://dparks.wikidot.com/source-code`
8. `http://www2.ulg.ac.be/telecom/research/vibe/`
9. L'auteur nous a contacté pour nous offrir son code source
10. `http://www.na.icar.cnr.it/~maddalena.l/MODLab/SoftwareSOBS.html`
11. `http://www.cvpr2012.org/program-details/workshops`
12. `http://www.dmi.usherb.ca/~jodoin/`
13. `http://www.merl.com/people/fatih/`
14. `http://blogs.bu.edu/jkonrad/`
15. `http://blogs.bu.edu/pi/`

ViBe+ : Van Droogenbroeck et Paquot [92] ;

KDE Integrated Spatio-temporal Features : Nonaka *et al.* [93] ;

KDE Spatio-temporal change detection : [À paraître] ;

Local-Self similarity : Jodoin *et al.* [94] ;

GMM RECTGAUSS-Tex : Riahi *et al.* [95] ;

Bayesian Background : Porikli et Tuzel [50].

Six méthodes parmi ces dernières ont été retenues [89, 90, 44, 91, 92, 93] pour une présentation orale en raison de leur bon classement sur le site ou pour leur originalité.

3.3.5 Popularité

Puisque notre but est de devenir le standard *de facto* en détection de changement, les évènements sur notre site et sa popularité font partie intégrante des résultats. Ces critères semblent subjectifs, mais il y a des outils pour se faire une idée de la situation.

Il est possible de juger de la popularité de notre site en utilisant Google Analytics [16]. Nous l'avons activé quelques semaines avant de publiciser le site et la compétition alors il nous donne une vision globale des chercheurs qui visitent notre site. Comme le montre la figure 3.17, nous avons un flot constant de visiteurs uniques depuis avril 2012, date de sa mise en ligne. Nous recevons en moyenne 100 visiteurs uniques par semaine (369 par mois), pour une moyenne de 190 visites au cours d'une semaine (817 par mois). Le grand nombre de visiteurs en juin est attribuable à l'atelier de détection de changement CVPR 2012. L'augmentation soudaine en octobre est inexpliquée car il n'y a eu aucun évènement particulier à ce moment. La majorité des visiteurs viennent des États-Unis d'Amérique, de Chine, d'Allemagne et du Canada.

De plus, les commentaires émis par la communauté de détection de changement étaient pour la plupart très positifs. Le nombre de participants et les conversations durant la période de discussion étaient très encourageantes. Tout ceci nous confirme que ce projet est la réponse à un besoin réel.

16. `http://www.google.com/analytics/`

en utilisant les moyennes calculées précédemment. Le classement moyen d'une méthode i pour une catégorie c est calculé ainsi :

$$Cc_{i,c} = \frac{1}{7} \sum_{m} rang_i(m, c) \tag{3.4}$$

où $rang_i \in \mathbb{N}$ et m désignent une des 7 mesures calculées avec l'équation 3.2. Il consiste donc à faire la moyenne des rangs pour toutes les métriques par rapport aux autres méthodes. Par exemple, si une méthode obtient un classement $Cc_{i,c} = 3.0$, cela signifie que cette méthode s'est classée au 3^e rang des métriques en moyenne.

Le rang global pour une méthode i est calculé ainsi :

$$C_i = \frac{1}{6} \sum_{c} Cc_{i,c}. \tag{3.5}$$

Les métriques moyennes globales pour une méthode i à travers toutes les catégories sont calculées ainsi :

$$Cg_i = \frac{1}{7} \sum_{m'} rang_i(m') \tag{3.6}$$

où m' est une des moyennes globales pour les métriques, telles que celles calculées dans l'équation 3.3. La figure 3.12 montre l'utilisation de ces équations sur le site [1].

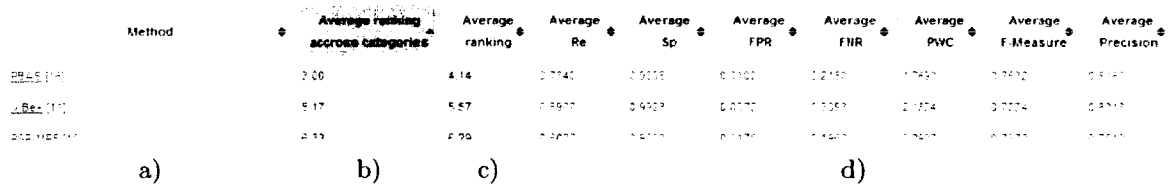


FIGURE 3.12 – Extrait du tableau des résultats globaux, pris sur le site [1]. a) La liste des méthodes. b) Classement global des méthodes, calculé selon l'équation 3.5. c) Classement des méthodes selon les moyennes calculées en d), calculé selon l'équation 3.6. d) Moyennes globales des différentes métriques, calculées selon l'équation 3.3.

Ces calculs divisés par catégorie et rapportés sur la banque de données globalement permettent d'obtenir un aperçu rapide des meilleures méthodes en ce moment. Une inspection des résultats dans une catégorie particulière permet de choisir une méthode

selon un besoin particulier. Les résultats globaux, quant à eux, permettent de choisir la meilleure méthode, toutes difficultés confondues.

Si, pour une raison ou pour une autre, notre méthode d'évaluation ne plaît pas à la communauté de détection de changement, le projet n'est pas inutile pour autant. Premièrement, il est aisé de changer la méthode d'évaluation en fonction sur le site web si tel est le désir de la communauté. Deuxièmement, nous savons que ces mesures ne plairont pas à tous, c'est pourquoi nous en avons choisi autant. De cette façon, les chercheurs peuvent ignorer ce qu'ils jugent impertinent. De plus, les données brutes sont disponibles pour toutes les méthodes. D'autres chercheurs peuvent alors télécharger ces données et utiliser une méthode d'évaluation adéquate selon leurs besoins. C'est d'ailleurs ce que le projet ViSOR [72] offrait ; il permettait de comparer des méthodes selon différents critères sur un ensemble de résultats fixes.

3.3 ChangeDetection.net

Ce site[3] est notre solution aux problèmes énoncés dans la section 1.7. Accessible à tous, il se veut la référence en matière de comparaison des méthodes de détection de changement. Il est présentement divisé en trois sections importantes : les pages pour l'ajout d'une méthode, la page de visualisation des résultats et une page archivée concernant la compétition à CVPR 2012, expliquée dans la section 3.3.4.

3.3.1 Création d'une méthode

Créer une nouvelle méthode sur le site demande au minimum de télécharger la banque de données CDnet, exécuter la méthode sur les séquences de CDnet, puis téléverser les masques binaires sur le site. Il est aussi possible de télécharger des outils et de la documentation afin de mieux comprendre les actions à effectuer.

Dans la page « DATASET, » il est possible de télécharger une partie ou la totalité de la banque de données CDnet. Dans le haut de la page, quelques informations sont fournies

3. `http://www.changedetection.net/`

sur la banque de données. Ensuite, il est possible de télécharger toute la banque de données sous deux formats compressés, soient zip et 7z. Comme on peut le voir dans la figure 3.13, il est aussi possible de télécharger une partie de la banque de données, individuellement par catégorie ou par vidéo. Il est définitivement plus pratique de télécharger la totalité de la banque de données, mais un téléchargement par parties pourrait être utile pour certains usagers disposant d'une mauvaise connexion Internet ou pour une vérification rapide.

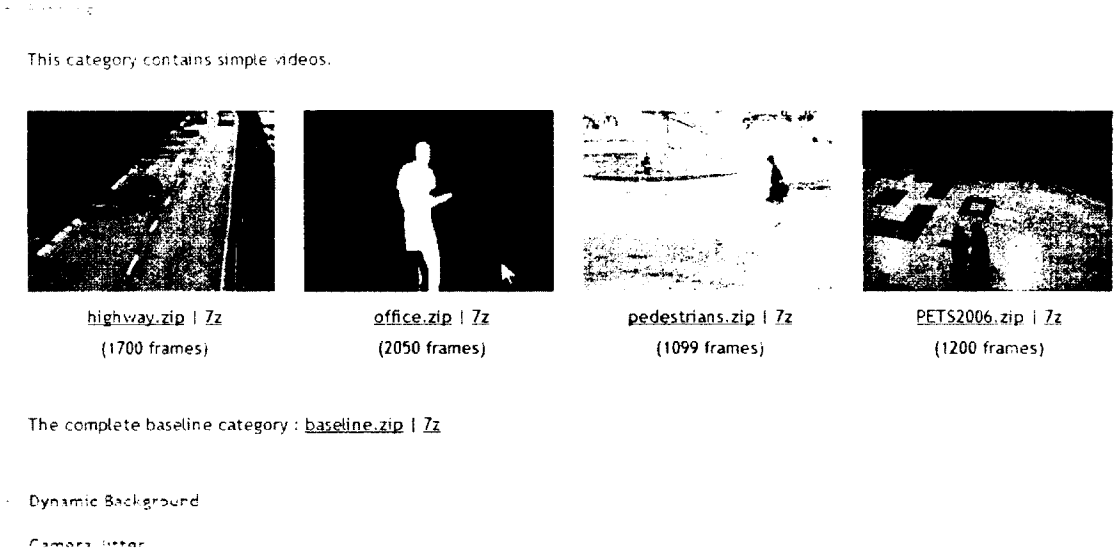


FIGURE 3.13 – Capture d'écran de la page « DATASET » où il est possible de télécharger la totalité ou une partie de la banque de données.

Comme le montre la figure 3.14, la page « UTILITIES » donne quelques informations sur ce que les méthodes doivent produire pour être acceptées sur le site. De plus, elle permet de télécharger des outils pour travailler avec la banque de données CDnet. La plupart des outils sont offerts en version Python[4] et C++ ou en version Matlab[5]. Offrant un outil commercial et un outil libre et gratuit, nous permettons à tous les chercheurs d'utiliser nos outils.

Nous offrons avant tout un gabarit d'une méthode de détection de changement. Ce dernier boucle sur les catégories et les séquences en exécutant une méthode laissée vide

4. `http://www.python.org/`
5. `http://www.mathworks.com/products/matlab/`

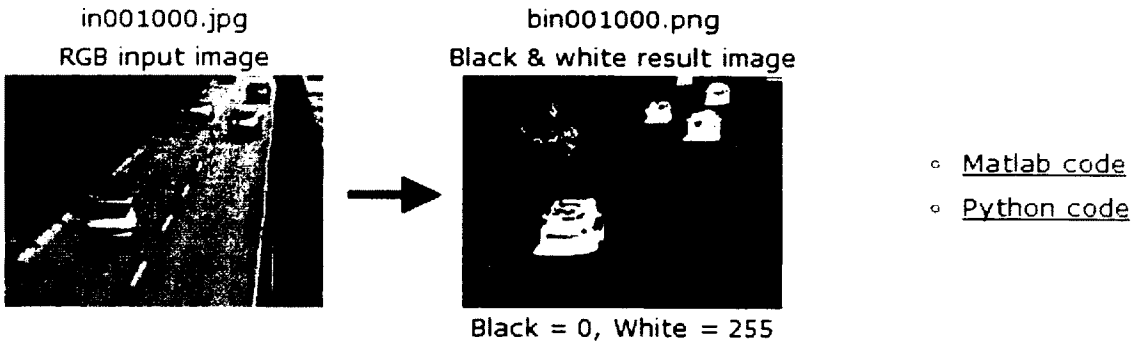


FIGURE 3.14 – Capture d'écran de la page « UTILITIES » où il est possible d'obtenir des informations et de télécharger des outils pour travailler sur la banque de données CDnet.

afin que les chercheurs puissent écrire leur code. La version Python appelle simplement un programme C++\OpenCV[6] avec les bons paramètres. La version Matlab utilise la *image processing toolbox* pour offrir les mêmes fonctionnalités.

Ce programme est normalement accompagné de l'outil de comparaison. Offert autant en Python qu'en Matlab, il compare toutes les images pixel à pixel avec les réalités de terrain afin de calculer les chiffres de la matrice de confusion et les autres mesures de classification expliquées dans la section 3.2. Ce programme crée un fichier de résultats « results.txt » en suivant les mêmes règles appliquées sur le site. Ces fichiers contiennent un bloc par catégorie et un bloc global, le tout en 5 colonnes de chiffres : VP, FP, FN, VN et le nombre de pixels ombragés mal classifiés. Ils ont la forme suivante :

```
cm video baseline highway 5220722 364651 237267 86289818 361263
cm video baseline PETS2006 4146397 574522 681793 366138782 0
cm video baseline pedestrians 640306 38050 30556 67531057 12502
cm video baseline office 8306975 173965 335119 116364721 0
cm category baseline 18314400 1151188 1284735 636324378 373765
...
cm overall 206147312 158636597 57525824 5758648629 4837894
```

Nous offrons aussi un programme Python permettant de calculer le classement de plusieurs méthodes. Il prend en entrée plusieurs fichiers de résultats pour ensuite donner

6. http://opencv.willowgarage.com/wiki/

le rang des méthodes. Cet outil peut être pratique pour un chercheur voulant comparer différents paramètres pour sa méthode avant de la soumettre.

Une fois les masques binaires générés avec une méthode et compressés en un fichier unique, le chercheur se dirige vers la page « UPLOAD » où il peut entrer les informations sur sa méthode et les détails pour le contacter. La taille du fichier compressé varie entre 100Mo et 200Mo, dépendant en grande partie du nombre de faux positifs car une image se compresse mieux lorsqu'elle a de grandes zones de couleur uniforme.

Lorsque la méthode est soumise sur le site, nous validons les résultats afin d'éviter la triche et les erreurs de soumission. Lorsqu'un problème est détecté, le chercheur reçoit un courriel contenant un message expliquant la raison du rejet de sa soumission.

3.3.2 Résultats

Lorsque nous recevons une soumission, nous décompressons le fichier envoyé par les chercheurs afin de le comparer avec les réalités de terrain. Les métriques introduites dans la section précédente sont ensuite enregistrées dans la base de données afin d'être visualisées plus tard. Le chercheur reçoit alors un message lui permettant de voir ses résultats et son classement parmi les autres méthodes présentes sur le site. Cette nouvelle méthode est affichée sur le site uniquement lorsque le chercheur le demande. Une fois la méthode visible à tous, elle apparaît dans la page « RESULTS » dans toutes les catégories où des résultats ont été soumis. Elle sera aussi visible dans la section globale si des résultats ont été soumis pour toutes les catégories. Comme le montre la figure 3.15, il est possible d'obtenir les informations sur une méthode particulière en cliquant sur son nom.

Tous les résultats sont visibles dans la page « RESULTS ». L'onglet ouvert par défaut présente les résultats globaux, donc le classement et les métriques de chaque méthode (voir figure 3.16.) Les 6 prochains onglets présentent les résultats par catégorie. Le dernier onglet contient les courbes ROC de nos 9 premières méthodes (voir section 3.3.4). Cette page contient aussi des explications sur les mesures utilisées et les références vers les méthodes soumises sur le site.

La seule contrainte à respecter pour apparaître dans cette section est d'utiliser un seul jeu de paramètres pour toutes les catégories et vidéos lors de l'analyse. Les paramètres

Contact name	Lucia Maddalena
Contact email	lucia.maddalena@na.icar.cnr.it
Contact university/company	National Research Center of Italy
Method's name	SC-SOBS
Reference	L. Maddalena, A. Petrosino, "The SOBS algorithm: what are the limits?", in proc of IEEE Workshop on Change Detection, CVPR 2012
Processing time	~23fps for 320x240 video, ~4fps for 720x576 video, C code on Core i3-330M 2.13GHz.
Web page	http://cvprlab.uniparthenope.it/
Parameters	none (all defaults)

You can download the change detection results of this method by clicking here.

FIGURE 3.15 – Capture d'écran du site [1] où il est possible de voir les informations sur la méthode et sur le scientifique à contacter pour toutes questions relatives à cette méthode.

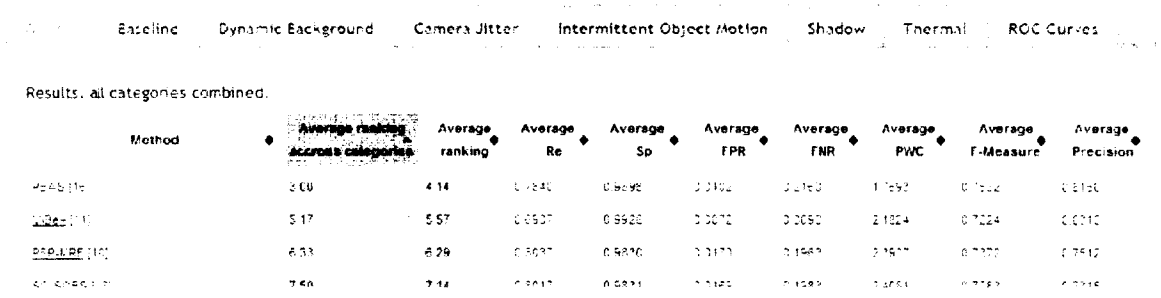


FIGURE 3.16 – Capture d'écran de résultats du site [1]. Les méthodes, leur classement et leurs métriques y sont visibles selon différentes catégories ou globalement.

choisis sont d'ailleurs visibles dans la page d'informations sur la méthode, tel que montré à la figure 3.15.

3.3.3 Discussion

Résumons rapidement la problématique actuelle pour la comparaison des méthodes de détection de changement :

- il n'existe aucune banque de données standard,
- il est difficile de reproduire les résultats des autres méthodes,
- aucune métrique d'évaluation n'a été retenue par la communauté.

À la lecture des derniers paragraphes, nous voyons que notre solution répond adéquatement à ces problèmes. Dans le premier cas, nous ne pouvons forcer la communauté à utiliser notre banque de données, mais nous avons mis tous les efforts nécessaires pour qu'elle devienne peu à peu le standard en détection de changement. Si notre projet est réellement adopté par la communauté scientifique, il va corriger par le fait même les deux autres problèmes.

Pour ce qui est de la difficulté de reproduire les résultats de l'état de l'art, ce n'est plus nécessaire car, si tout fonctionne comme nous l'espérons, ils seront tous disponibles sur le site. Tous les résultats téléversés sur le site sont disponibles en téléchargement peu de temps après. Les résultats des mesures de classification sont bien sûr disponibles, mais aussi l'archive soumise par les chercheurs. Un chercheur peut alors télécharger les images résultant d'une autre méthode et les insérer dans son article à des fins de comparaison. Il sera donc aussi simple de se comparer aux méthodes « simples » qu'aux meilleures méthodes du moment. Nul besoin de programmer la méthode du compétiteur ni de chercher le code source, il suffit de télécharger les résultats officiels de cette méthode.

Dans l'éventualité où notre projet devient le standard *de facto* en détection de changement, les comparaisons seront faites sur la même base. Tel qu'expliqué dans la section précédente, nous offrons une méthode d'évaluation qui se veut juste et non biaisée. De plus, elle permet aux chercheurs d'établir leur propre classement en utilisant les données sur le site, ce qui peut être utile dépendamment des besoins de leurs projets.

3.3.4 Compétition CVPR2012

La conclusion de ce projet dépend directement de la participation de la communauté ; il ne peut fonctionner sans publicité et sans aide. C'est pourquoi nous avons pris quelques mesures afin d'encourager la communauté scientique à adopter notre projet.

La première étape fut d'ajouter nous-mêmes quelques méthodes sur le site. Les méthodes choisies sont représentatives des grandes familles en détection de changement. Nous avons sélectionné 9 méthodes, calculé les masques binaires avec ces dernières et téléversé les résultats sur le site. Les méthodes sélectionnées sont les suivantes :

Distance euclidienne [58] : notre implémentation ;

Distance de Mahalanobis [58] : notre implémentation ;

GMM Stauffer-Grimson : implémentation de la célèbre méthode de Stauffer et Grimson [33], programmée par Donovan Parks [7] ;

GMM Zivkovic : implémentation de la méthode de Zivkovic [88], programmée par Donovan Parks ;

GMM KaewTraKulPong : implémentation de la méthode de KaewTraKulPong et Bowden [31], fournie par le projet libre OpenCV ;

KDE : notre implémentation de la méthode d'Elgammal *et al.* [35] ;

ViBe : implémentation officielle [8] de la méthode de Barnich et Van Droogenbroeck [41] ;

KNN : implémentation officielle [9] de la méthode de Zivkovic et van der Heijden [88] ;

SOBS : implémentation officielle [10] de la méthode de Maddalena et Petrosino [43].

Ensuite, nous avons organisé un atelier (*workshop*) et une compétition de détection de changement à la conférence CVPR 2012 [11]. Organisée par Pierre-Marc Jodoin [12], Fatih Porikli [13], Janusz Konrad [14] et Prakash Ishwar [15], le but était triple : faire connaître le site, ajouter des méthodes à la page de résultats et discuter directement avec une partie de la communauté de détection de changement. Jouissant d'une excellente notoriété dans le milieu académique, Fatih Porikli s'est chargé entre autre de publiciser le site et le projet. Au final, cette compétition nous a permis d'ajouter 11 méthodes sur le site :

PBAS : Hofmann *et al.* [89] ;

Chebyshev probability approach : Morde *et al.* [90] ;

Chebyshev probability with static object detection : Morde *et al.* [90] ;

SC-SOBS : Maddalena et Petrosino [44] ;

PSP-MRF : Schick *et al.* [91] ;

7. `http://dparks.wikidot.com/source-code`
8. `http://www2.ulg.ac.be/telecom/research/vibe/`
9. L'auteur nous a contacté pour nous offrir son code source
10. `http://www.na.icar.cnr.it/~maddalena.l/MODLab/SoftwareSOBS.html`
11. `http://www.cvpr2012.org/program-details/workshops`
12. `http://www.dmi.usherb.ca/~jodoin/`
13. `http://www.merl.com/people/fatih/`
14. `http://blogs.bu.edu/jkonrad/`
15. `http://blogs.bu.edu/pi/`

ViBe+ : Van Droogenbroeck et Paquot [92] ;

KDE Integrated Spatio-temporal Features : Nonaka *et al.* [93] ;

KDE Spatio-temporal change detection : [À paraître] ;

Local-Self similarity : Jodoin *et al.* [94] ;

GMM RECTGAUSS-Tex : Riahi *et al.* [95] ;

Bayesian Background : Porikli et Tuzel [50].

Six méthodes parmi ces dernières ont été retenues [89, 90, 44, 91, 92, 93] pour une présentation orale en raison de leur bon classement sur le site ou pour leur originalité.

3.3.5 Popularité

Puisque notre but est de devenir le standard *de facto* en détection de changement, les évènements sur notre site et sa popularité font partie intégrante des résultats. Ces critères semblent subjectifs, mais il y a des outils pour se faire une idée de la situation.

Il est possible de juger de la popularité de notre site en utilisant Google Analytics [16]. Nous l'avons activé quelques semaines avant de publiciser le site et la compétition alors il nous donne une vision globale des chercheurs qui visitent notre site. Comme le montre la figure 3.17, nous avons un flot constant de visiteurs uniques depuis avril 2012, date de sa mise en ligne. Nous recevons en moyenne 100 visiteurs uniques par semaine (369 par mois), pour une moyenne de 190 visites au cours d'une semaine (817 par mois). Le grand nombre de visiteurs en juin est attribuable à l'atelier de détection de changement CVPR 2012. L'augmentation soudaine en octobre est inexpliquée car il n'y a eu aucun évènement particulier à ce moment. La majorité des visiteurs viennent des États-Unis d'Amérique, de Chine, d'Allemagne et du Canada.

De plus, les commentaires émis par la communauté de détection de changement étaient pour la plupart très positifs. Le nombre de participants et les conversations durant la période de discussion étaient très encourageantes. Tout ceci nous confirme que ce projet est la réponse à un besoin réel.

16. `http://www.google.com/analytics/`

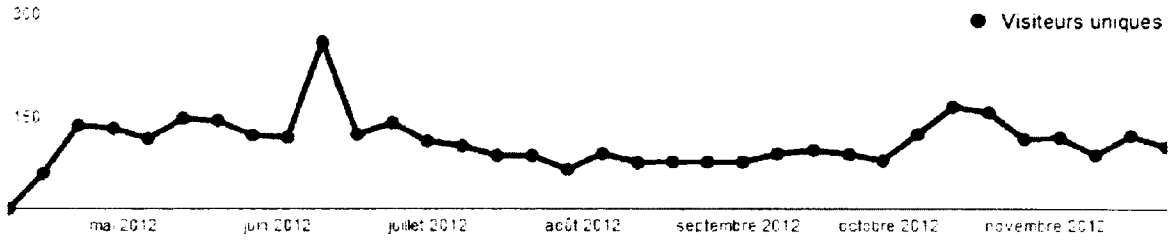


FIGURE 3.17 – Visiteurs uniques par semaine depuis l'ouverture du site. Le sommet en juin correspond à l'atelier de détection de changement CVPR 2012.

Bien que les ajouts de méthodes avant et durant l'atelier à CVPR 2012 ont été les plus importants, des méthodes s'ajoutent de temps à autre. Certaines méthodes restent invisibles sur le site car elles ont été soumises uniquement à des fins de tests, par exemple pour mieux comprendre l'effet d'une modification particulière sur les résultats. Les méthodes suivantes ont été ajoutées à la suite de l'atelier :

SGMM : Evangelio *et al.* [96].

Color Histogram Backprojection : Kit *et al.* [97] ;

UBA : Park et Byun [98] ;

Quasi-Continuous Histograms based Motion Detection : Sidibé *et al.* [99] ;

SGMM-SOD : Evangelio *et al.* [96] ;

CDPS : Hernandez *et al.* (à paraître) ;

DPGMM : Haines et Xiang [100].

Possiblement influencés par notre projet, certains auteurs ont rendu disponible à tous une partie de leur travail. Un tel mouvement de collaboration est important en science au niveau de la reproductibilité des résultats, surtout considérant que ces méthodes sont parmi les meilleures en ce moment sur le site. Les implémentations suivantes ont été ajoutées à la suite de l'atelier :

PBAS : code C++ et Matlab offert [17] ;

SC-SOBS : exécutable Windows offert [18] ;

17. `https://sites.google.com/site/pbassegmenter/download-1`
18. `http://www.na.icar.cnr.it/~maddalena.l/MODLab/SoftwareSC-SOBS.html`

ViBe+ et KNN : message des auteurs confirmant qu'ils allaient partager le code ou l'exécutable prochainement.

Ce qui s'ajoute aux quelques méthodes déjà disponibles en ligne. Espérons que cette philosophie de partage ne s'estompera pas avec le temps.

3.4 Résultats

Bien que ce soit le but premier, un projet de ce type n'a pas que des classements et une banque de données à offrir à la communauté de détection de changement. Ayant les résultats de dizaines de méthodes sur notre serveur, nous avons accès à une mine d'informations. Nous croyons que les données récoltées sont un échantillon représentatif des méthodes de détection de changement et qu'une analyse méticuleuse permettra de confirmer ou d'infirmer les conclusions de certains articles scientifiques, mais aussi d'émettre de nouvelles hypothèses.

Les résultats globaux sont compilés en annexe dans le tableau 3.3. Il est aussi possible de voir les résultats pour la catégorie « Basique » (tableau 3.4), « Arrière-plan dynamique » (tableau 3.5), « Mouvements de la caméra » (tableau 3.6), « Mouvement intermittent des objets » (tableau 3.7), « Ombres » (tableau 3.8) et « Thermique » (tableau 3.9). On remarque dans les résultats globaux du tableau 3.3 en annexe que les méthodes basées sur KDE semblent être meilleures que celles basées sur les mixtures de gaussiennes. En règle générale, les méthodes paramétriques (distance euclidienne, mahalonabis et modèle par histogramme) sont moins performantes. Inversement, les méthodes non paramétriques (PBAS, ViBe+, PSP-MRF), souvent récentes et plus complexes, sont généralement les meilleures. On constate aussi que le classement respecte presque le classement selon la f-mesure et le pourcentage de mauvaises classifications, ce qui confirme que ces mesures seraient un choix intéressant comme mesures uniques.

3.4.1 Difficultés non résolues

La quantité de données à notre disposition est rapidement devenue intéressante avec les méthodes ajoutées durant et après la compétition CVPR 2012. Nous les avons donc utilisées

afin de comprendre les tendances des méthodes de détection de changement. Analyser les résultats des méthodes peut nous indiquer si une difficulté est résolue ou non.

Nous avons tenté quelques expériences avec les 18 méthodes disponibles à ce moment. Pour être sélectionnée, une méthode devait avoir soumis des résultats pour toutes les catégories. Il aurait été possible de prendre toutes les méthodes sans distinction, mais certains tests auraient été impossibles, certaines méthodes ayant soumis des résultats pour un sous-ensemble de catégories.

Notre premier test fut de créer des images en niveaux de gris dans le but de visualiser les endroits où les méthodes croient collectivement qu'il y a du mouvement. Lorsqu'un grand nombre de méthodes classifient un pixel en mouvement, ce dernier apparaît en niveaux de gris clair,

$$I_{s,t} = \frac{N_s}{M} * 255 \tag{3.7}$$

où N_s est le nombre de méthodes ayant classifié le pixel s en mouvement et M le nombre total de méthodes, soit 18 dans le cas présent. Cette équation génère des images comme celles de la figure 3.18.

Ces images nous permettent déjà de tirer certaines conclusions, mais nous préférions améliorer la visualisation. Pour ce faire, nous nous sommes inspirés de l'évaluation subjective décrite dans l'article de Karaman *et al.* [69]. Comme le montre la figure 3.19, cette technique respecte le concept du dernier test, mais permet aussi de voir rapidement la différence entre les faux positifs et les vrais positifs à l'aide de l'équation suivante :

$$I_{s,t} = (255, 255, 255) - \frac{N_{FN}}{M} * (0, 255, 255) + \frac{N_{FP}}{M} * (0, 255, 0)) \tag{3.8}$$

où N_{FN} est le nombre de faux négatifs pour ce pixel parmi les M méthodes et N_{FP} est similaire, mais pour les faux positifs. Ainsi, plus les faux négatifs sont fréquents pour un pixel s, plus ce pixel sera rouge clair. Idem pour les faux positifs, mais en vert clair. Un pixel ne peut pas avoir des teintes rouges et vertes en même temps car selon les réalités de terrain, un pixel est soit en mouvement, soit statique, soit ignoré.

Il est difficile d'insérer suffisamment d'images dans ce document pour convaincre le lecteur, mais l'observation de ces images pour toutes les séquences nous montre clairement

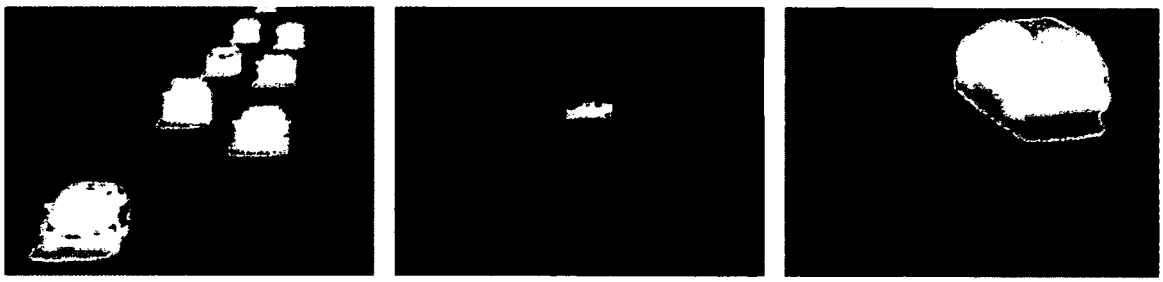
« Highway » « Fountain01 » « Traffic »

« Winter Driveway » « Backdoor » « Corridor »

FIGURE 3.18 – Échantillon des images en niveaux de gris selon le nombre de classifications en mouvement, calculées selon l'équation 3.7.

que les méthodes échouent collectivement dans certaines cisconstances, ce qui nous permet d'identifier des défis à venir.

Nos observations indiquent clairement que les ombres représentent encore un grave problème en détection de changement. La robustesse des méthodes face aux ombres dures a été testée à l'aide de l'étiquette « Ombre dure », il est donc possible de vérifier si l'ombre est réellement un problème. On peut voir dans la table 3.10 en annexe que nos données tendent à confirmer ce problème. Les meilleures méthodes à ce niveau sont en erreur dans le tiers des cas et plus. Il faut toutefois se méfier de ces résultats car on ne peut pas différencier les méthodes qui classifient bien l'ombre et celles qui détectent simplement moins de mouvement en général. On constate en regardant ce tableau que les meilleures méthodes pour la classification de l'ombre ne sont pas les meilleures globalement. Elles sont parmi les pires avec des taux de faux positifs sur l'ombre de 55% et plus. On peut voir dans les figures 3.19 et 3.20 que l'ombre est encore un problème d'actualité.



« Highway » « Fountain01 » « Traffic »

« Winter Driveway » « Backdoor » « Corridor »

FIGURE 3.19 – Échantillon des images couleur selon le nombre de faux positifs et de faux négatifs, calculées selon l'équation 3.8. La couleur est nécessaire pour bien visualiser cette figure.

FIGURE 3.20 – Les ombres douces et celles émises par les objets fixes sont moins problématiques, mais la grande majorité des méthodes échouent sur les ombres dures. Images de la séquence « People In Shade ».

Le camouflage est un problème grave et très fréquent en détection de changement. Diminuer le seuil est souvent une technique simple pour combattre le camouflage, au coût d'une augmentation du nombre de faux positifs. L'observation des images couleur indique clairement que ce problème est encore aussi grave et fréquent. La figure 3.21 en est un bon exemple.

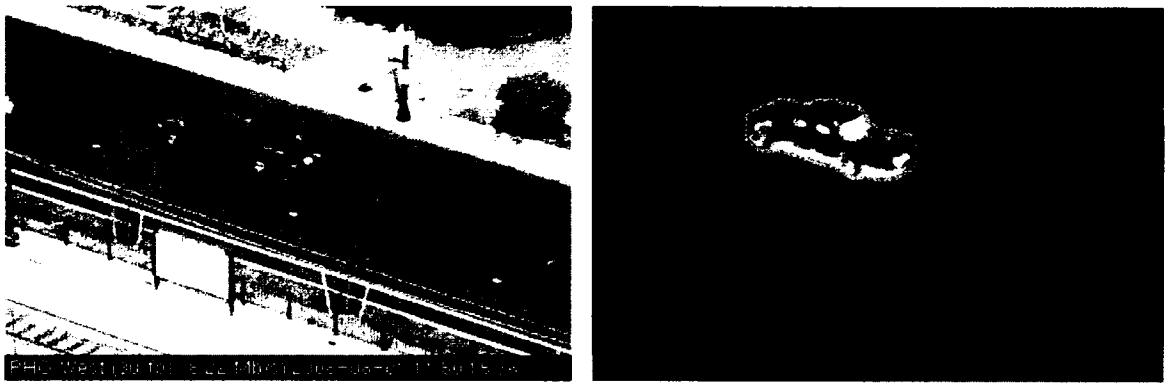


FIGURE 3.21 – Cette camionnette est considérée comme l'arrière-plan par la grande majorité des méthodes soumises sur le site. À l'oeil, son intensité est d'ailleurs assez semblable à celle de la rue. Images de la séquence « Boulevard ».

Bien que de nombreuses méthodes visent à corriger les problèmes causés par les arrière-plans dynamiques, ce problème n'est plus ce qu'il a déjà été. En effet, le mouvement dans les arbres et l'eau ne cause plus beaucoup de faux positifs. L'arrivée des méthodes multi-modales telles que GMM de Stauffer et Grimson [33] a apporté des pistes de solutions à la communauté de détection de changement.

Les problèmes causés par les mouvements de la caméra ont aussi perdu de leur importance grâce, encore une fois, aux méthodes multi-modales. On peut voir les vibrations de la caméra comme un arrière-plan dynamique au même titre que le mouvement des arbres et de l'eau, mais celui-ci est appliqué globalement sur l'image et le mouvement est souvent plus important.

L'utilisation de caméras thermiques apporte son lot de problèmes et nous l'avons confirmé en analysant les images couleur. Le problème de camouflage est très présent dans

ce type de séquences car les objets en mouvement peuvent être de la même température que l'arrière-plan. Ces résultats contredisent la croyance populaire selon laquelle l'utilisation de caméras thermiques simplifie largement l'analyse vidéo. Le retour en niveaux de gris cause aussi des problèmes au niveau du camouflage car il y a deux dimensions de moins avec lesquelles travailler.

Il est aussi possible d'utiliser des courbes PR pour constater le défi que représente certaines difficultés. La figure 3.22 montre clairement que la catégorie « Basique » est un moins grand défi pour la majorité des méthodes et que « Mouvement intermittent des objets » en est un. Les résultats sont moins clairs pour les autres catégories par contre. Ceci est probablement dû au fait que les méthodes se trompent généralement lorsqu'il y a du mouvement dans la scène, mais il y en a peu dans notre banque de données. Notre façon d'utiliser les régions d'intérêt peut aussi être en cause ; la catégorie « mouvement intermittent des objets » a des régions d'intérêts beaucoup plus restreintes que les autres et sa courbe PR est un meilleur indicateur de sa difficulté réelle. Une chose est certaine, à la lumière de ces courbes, les vidéos mettant en scène des objets ayant un mouvement intermittent posent un défi majeur. Le deuxième plus grand défi est la robustesse face aux mouvements de caméra, suivi de l'arrière-plan dynamique et des vidéos thermiques.

À la lumière des derniers paragraphes, il semble important pour la communauté de détection de changement de faire face à de nouveaux défis. Les méthodes sont souvent publicisées pour leur robustesse face aux mouvements parasites de l'arrière-plan, mais ce problème n'est plus ce qu'il a déjà été. Les nouvelles méthodes devraient se concentrer sur les autres difficultés importantes, soit l'ombre et le camouflage. Si les mouvements de la caméra et les caméras thermiques étaient plus fréquents, nous les considérerions probablement comme des problèmes cruciaux à corriger, mais l'ombre et le camouflage sont prioritaires.

3.4.2 Vote majoritaire

Un vote majoritaire s'avère parfois être une méthode intéressante pour des algorithmes de classification [101, 102]. Nous avons donc effectué quelques tests avec les méthodes de détection de changement répertoriées sur le site web. Les méthodes utilisées dans cette

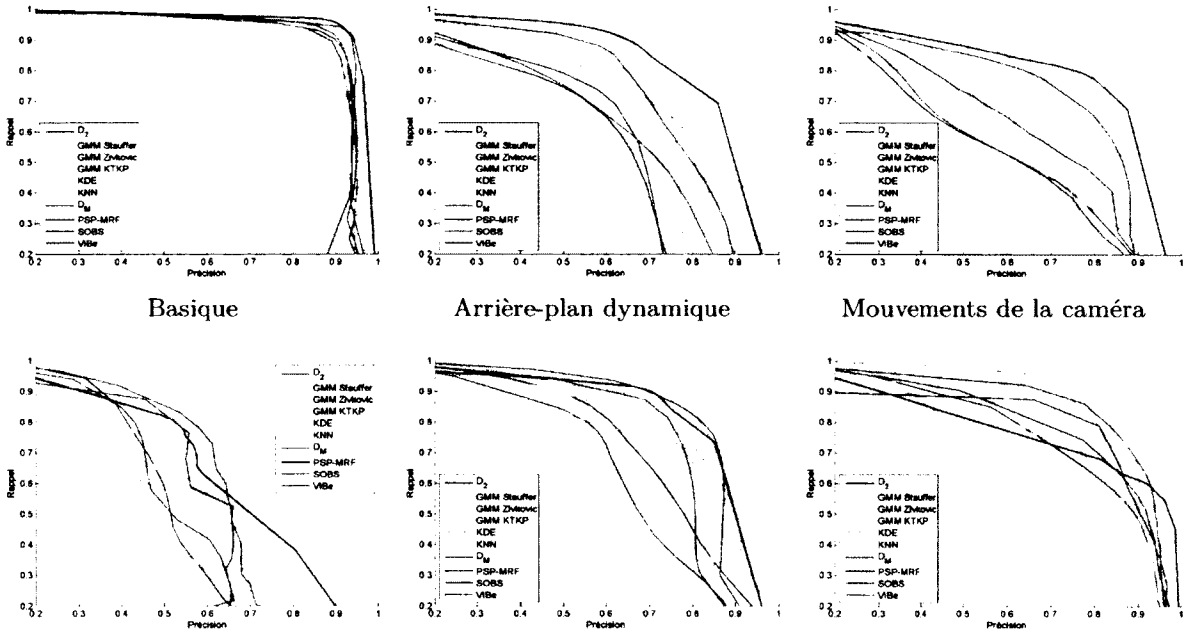


FIGURE 3.22 – Courbes PR des 6 catégories de notre banque de données. Ces dernières donnent certaines indications sur les difficultés inhérentes à ces catégories.

section sont visibles dans le tableau 3.3 en annexe.

Notre premier test est le vote majoritaire. Pour chaque pixel s, ce pixel est classifié en mouvement si une majorité des méthodes le classifie en mouvement :

$$\chi_t(s) = \left\lceil \frac{N_s}{M} \right\rfloor \tag{3.9}$$

où N_s est le nombre de méthodes ayant classifié le pixel s en mouvement et M le nombre de méthodes. Les résultats de cette nouvelle méthode sont impressionnants. Sans compter les méthodes que nous avons générées dans cette section, le vote majoritaire avec les 23 méthodes présentes sur le site est présentement la meilleure méthode sur le site. La figure 3.23 donne des indications sur la performance de cette méthode à l'aide de courbes ROC.

On peut alors se demander s'il est possible d'obtenir des résultats équivalents ou meilleurs avec un minimum de méthodes. Un vote majoritaire entre 23 méthodes pour obtenir de bons résultats est un processus assez complexe et long ; il serait donc préférable

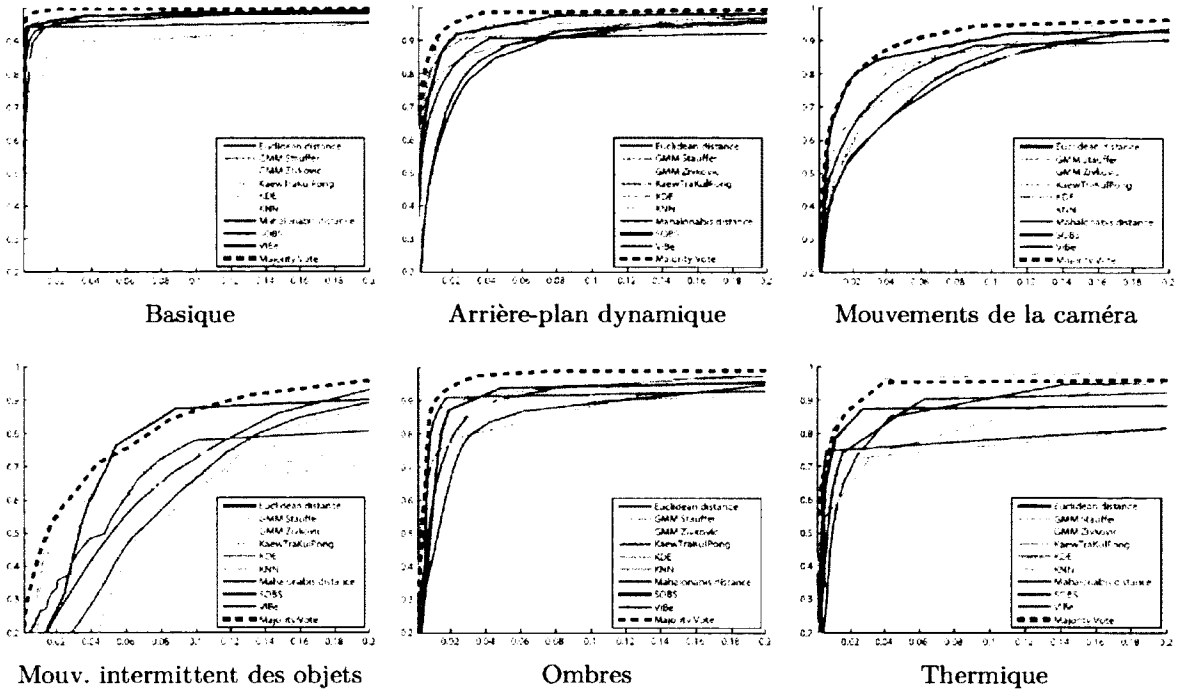


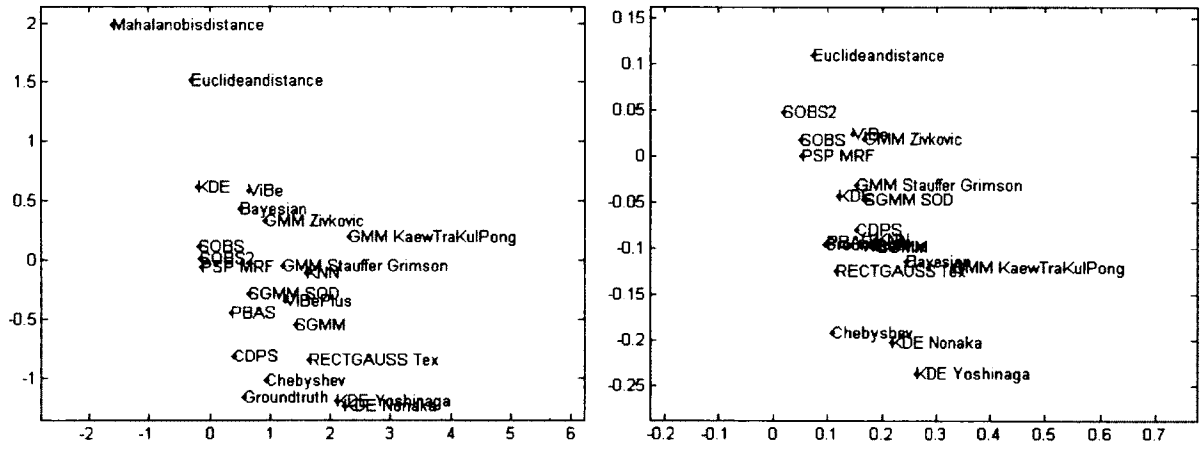
FIGURE 3.23 – Courbes ROC des 6 catégories de notre banque de données. La ligne pointillée large représente le vote majoritaire avec les méthodes disponibles sur le site lors du test.

d'utiliser moins de méthodes. On se demande alors quelles méthodes utiliser parmi celles soumises sur le site.

Un premier réflexe est d'utiliser les 5 meilleures méthodes disponibles sur le site. Un vote majoritaire avec les 5 meilleures méthodes devrait intuitivement aboutir au meilleur vote possible. À défaut d'être vrai, ce raisonnement nous donne encore une fois la meilleure méthode sur le site à ce jour. Le même test est effectué avec les trois meilleures méthodes et nous obtenons encore une fois une méthode ayant un meilleur classement que celles soumises par les chercheurs. Elle obtient toutefois un score légèrement moins bon que le vote majoritaire avec les 5 meilleures méthodes.

Il est sans doute possible d'obtenir une méthode encore meilleure avec une combinaison des méthodes soumises sur le site. Toutefois, la question sur le choix des méthodes à

utiliser revient. Nous avons tenté de placer toutes les méthodes et la réalité de terrain de la banque de données sur un plan construit avec du positionnement multidimensionnel, *multidimensional scaling* [103]. Cette technique consiste à placer des objets dans un monde 2D ou 3D de sorte que les distances entre les objets soit respectées. Une personne pourrait, par exemple, reconstruire la carte du Québec en 2D avec les distances entre les villes. Nous pourrions ensuite choisir les 5 méthodes les plus près de la réalité de terrain sur ce plan. L'idée est logique et aurait peut-être fonctionné si nous avions trouvé une mesure de distance sensée entre les méthodes. Nous avons seulement testé une mesure de distance, soit le nombre de pixels classifiés différemment selon deux méthodes. Malgré plusieurs tests avec ces données, cette mesure de distance n'était pas concluante. Deux de ces tests sont visibles à la figure 3.24. On remarque rapidement que les méthodes proches de la réalité de terrain (« Groundtruth ») ne sont pas nécessairement bien classées sur le site. Bien que certaines distances semblent logiques, le tout a une apparence aléatoire.



a) Moyenne par catégorie

b) Moyenne par vidéo

FIGURE 3.24 – Cartes des distances entre les méthodes et la réalité de terrain selon deux mesures de distance. Ces cartes ne concordent pas avec les résultats.

Face à cette impasse, nous avons simplement testé toutes les possibilités. Toutes les combinaisons de 3 et 5 méthodes parmi les 23 ont été soumises au vote majoritaire afin d'obtenir la meilleure combinaison. Nous avons donc testé $\binom{3}{23} + \binom{5}{23} = 35420$ combinaisons. La meilleure combinaison de trois méthodes (CDPS, PSP-MRF, ViBe+) et de 5 méthodes

(Chebyshev, PBAS, PSP-MRF, GMM - Riahi, ViBe+) donne encore une fois les meilleures méthodes parmi celles soumises sur le site. Les résultats sont présentés à la table 3.2 sous les noms « comb-3 » et « comb-5 ».

Les résultats des 5 méthodes que nous avons générées durant les derniers paragraphes sont répertoriés dans la table 3.2. Le vote majoritaire avec les 5 méthodes les plus performantes sur notre site web et la combinaison de 5 méthodes sont les gagnantes de cette compétition. On retrouve ensuite le vote majoritaire avec les 3 meilleures méthodes, la combinaison avec 3 méthodes et le vote majoritaire avec les 23 méthodes. Ces méthodes ne sont pas automatiquement meilleures que celles soumises sur le site. La meilleure méthode sur le site est aussi présente dans le tableau 3.2 ; on constate qu'elle est très près du vote majoritaire avec toutes les méthodes et de la combinaison avec 3 méthodes.

TABLE 3.2 – Résultats globaux pour les méthodes expliquées dans cette section et pour la méthode la plus performante sur le site.

Méthodes	C_i	Cg_i	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
5 meilleures méthodes	1.67	2.0	0.789	0.993	0.007	0.211	1.428	0.788	0.856
comb-5	1.67	1.86	0.782	0.994	0.006	0.217	1.422	0.783	0.867
3 meilleures méthodes	3.17	3.0	0.765	0.994	0.006	0.235	1.491	0.775	0.868
comb-3	4.5	4.29	0.777	0.992	0.008	0.223	1.733	0.771	0.824
PBAS [89]	4.83	4.57	0.784	0.99	0.01	0.216	1.769	0.753	0.816
Vote majoritaire	4.83	5.29	0.753	0.992	0.008	0.247	1.999	0.75	0.82

On suppose que les meilleures combinaisons avec 7 méthodes et avec 9 méthodes donnent des résultats d'une qualité supérieure, mais ces calculs deviennent rapidement très coûteux. Une combinaison avec 7 méthodes requiert $\binom{7}{23} = 245157$ tests, soit environ 7 fois plus que la combinaison avec 5 méthodes qui avait demandé plusieurs jours de calcul. Un outil pour déterminer les meilleures combinaisons de méthodes est nécessaire pour continuer ces expériences. Sans un tel outil, on ne peut que supposer une amélioration avec des combinaisons de 7, 9, 11 et peut-être 13 méthodes, suivie d'une chute lorsque le nombre de méthodes devient trop grand. Cette chute serait justifiée par le fait que les nouvelles méthodes à ajouter ne sont plus complémentaires car elles n'ajoutent plus d'informations utiles au vote et elles ne font que répéter les erreurs des autres méthodes. Finalement, avec une combinaison de 23 méthodes, nous atteignons la qualité du vote majoritaire car

ces deux tests travaillent sur les mêmes méthodes. Bref, la meilleure combinaison possible demande probablement entre 5 et 23 méthodes.

Conclusion

En réponse aux divers problèmes d'évaluation des méthodes de détection de changement, nous avons créé une banque de données comprenant 31 séquences divisées en 6 catégories et couvrant les difficultés usuelles du domaine. Nous avons également une méthodologie d'évaluation équitable et un site où les chercheurs peuvent téléverser leurs résultats et se comparer à leurs pairs. Avec une participation adéquate de la communauté de détection de changement, ce site peut résoudre les problèmes de comparaison qui affectent la communauté depuis quelques années. La solution tient en deux points : permettre aux chercheurs de se comparer sur une base commune, autant au niveau des vidéos que des métriques utilisées, et offrir les résultats des autres méthodes afin que les chercheurs n'aient plus à les implémenter eux-mêmes ou à choisir les mêmes méthodes dont le code est déjà disponible en ligne.

À cause de la nature de ce projet, qui a pour but d'être le standard *de facto* en détection de changement, il est encore tôt pour déterminer si nous avons atteint nos objectifs. Cela dit, l'atelier et la compétition à CVPR 2012, la réponse de la communauté, le nombre de visiteurs uniques sur le site (en moyenne 369 visiteurs uniques par mois) et le nombre de soumissions sont tous des facteurs encourageants quant à la réussite de ce projet. Ce travail a déjà eu un impact majeur sur la communauté de détection de changement et il est probable que cet impact ne cesse de grandir avec le temps.

Il est clair que ce projet pourrait être amélioré. Le temps et les ressources à notre disposition étaient limités ; nous avons donc dû couper certaines séquences et quelques catégories sur la banque de données. Il serait important d'ajouter des séquences avec beaucoup d'objets en mouvenent, de type aéroport, des séquences avec des conditions météorologiques particulières (brume, neige, etc.), des séquences capturées de nuit afin d'analyser l'effet du bruit et du manque de constraste, des séquences avec une compression

plus forte et peut-être même variable, des séquences avec un nombre faible ou irrégulier d'images par seconde et des séquences avec de nombreux objets en mouvement dès la première image pour voir à quelle vitesse les méthodes apprennent le modèle de l'arrière-plan.

De plus, lors de la séance de discussion à l'atelier CVPR 2012, les demandes étaient nombreuses et diverses. Certains chercheurs souhaitent que des boîtes englobantes soient ajoutées autour des objets afin que notre banque de données soit utilisée en suivi d'objet. Ceci nous permettrait de pondérer les pixels mal classifiés selon la distance à l'objet le plus près. Certains demandent de nouvelles façons de calculer le classement, notamment un classement 2 à 2. D'autres désirent de nouveaux types de séquences, par exemple des séquences stéréos, aériennes, et des cartes de disparité.

Il n'y a pas que la banque de données qui pourrait être améliorée. La méthode d'évaluation pourrait possiblement l'être également. L'ajout d'une méthode influence présentement le classement de la plupart des méthodes sur le site, ce qui est généralement souhaitable. Le problème est que l'ajout d'une méthode inverse parfois le rang de deux autres méthodes. Il est illogique qu'une méthode devienne soudainement meilleure qu'une autre méthode lors de l'ajout d'une méthode tierce. Il serait aussi utile de trouver une mesure de distance entre les méthodes afin de construire un plan des méthodes par rapport aux autres méthodes et à la réalité de terrain. Ceci nous permettrait peut-être aussi de trouver la bonne combinaison de méthodes pour un vote populaire au lieu de tester tous les cas possibles.

Heureusement, ces travaux futurs ont d'excellentes chances d'être effectués. Des discussions sont déjà en cours avec divers chercheurs aux États-Unis, en Europe et en Asie afin d'avoir un maximum de contributions aux prochaines versions de la banque de données et aux prochaines compétitions. Ce projet a peu de chances d'être un échec ; il se dirige plutôt vers une réussite et il ouvre la porte à de nombreuses compétitions futures.

Annexe

TABLE 3.3 – Résultats globaux selon les méthodes appliquées sur la banque de données CDnet et soumises sur le site CDnet.

Méthodes	C_i	Cg_i	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
PBAS [89]	4.0	4.86	0.784	0.99	0.01	0.216	1.769	0.753	0.816
SGMM-SOD [104]	4.33	4.14	0.759	0.993	0.007	0.241	1.589	0.758	0.835
ViBe+ [92]	6.17	7.0	0.691	0.993	0.007	0.309	2.182	0.722	0.832
PSP-MRF [91]	7.5	7.57	0.804	0.983	0.017	0.196	2.394	0.737	0.751
CDPS (à paraître)	8.83	7.71	0.777	0.985	0.015	0.223	2.275	0.728	0.761
Chebyshev prob. SOD [90]	9.17	8.57	0.713	0.989	0.011	0.287	2.386	0.7	0.786
SC-SOBS [44]	9.17	8.43	0.802	0.983	0.017	0.198	2.408	0.728	0.732
SGMM [96]	10.67	8.43	0.707	0.991	0.009	0.293	2.531	0.701	0.781
KNN [32]	11.5	10.71	0.671	0.991	0.009	0.329	2.795	0.678	0.788
SOBS [43]	11.83	10.86	0.788	0.982	0.018	0.212	2.564	0.716	0.718
KDE - Nonaka [93]	12.33	11.29	0.651	0.993	0.007	0.349	2.89	0.642	0.766
KDE - ElGammal [35]	12.5	15.14	0.744	0.976	0.024	0.256	3.46	0.672	0.684
GMM - KaewTraKulPong [31]	12.83	12.14	0.507	0.995	0.005	0.493	3.105	0.59	0.823
ViBe [41]	13.0	15.0	0.682	0.983	0.017	0.318	3.118	0.668	0.736
KDE - Yoshinaga (à paraître)	14.33	12.86	0.658	0.991	0.009	0.342	3.002	0.644	0.734
Bayesian Background [50]	15.17	17.29	0.602	0.983	0.017	0.398	3.388	0.627	0.744
GMM - Stauffer [33]	16.5	12.86	0.711	0.986	0.014	0.289	3.104	0.662	0.701
GMM - Zivkovic [88]	18.33	14.71	0.696	0.985	0.015	0.304	3.15	0.66	0.708
Local-Self similarity [94]	18.67	16.71	0.935	0.851	0.149	0.065	14.295	0.502	0.414
GMM - Riahi [95]	18.83	17.14	0.516	0.986	0.014	0.484	3.684	0.522	0.719
Histogramme [26]	19.33	17.57	0.77	0.934	0.066	0.23	6.968	0.548	0.525
Distance de Mahalonabis [58]	20.5	16.86	0.761	0.96	0.04	0.239	4.663	0.626	0.604
Distance euclidienne [58]	22.0	18.14	0.705	0.969	0.031	0.295	4.346	0.611	0.622

TABLE 3.4 – Résultats pour la catégorie « Basique » selon les méthodes appliquées sur la banque de données CDnet et soumises sur le site CDnet.

Méthodes	$Cc_{i,c}$	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
SC-SOBS [44]	3.0	0.933	0.998	0.002	0.067	0.375	0.933	0.934
SOBS [43]	4.86	0.919	0.998	0.002	0.081	0.433	0.925	0.931
PSP-MRF [91]	5.57	0.932	0.998	0.002	0.068	0.413	0.929	0.926
KDE - ElGammal [35]	8.29	0.897	0.998	0.002	0.103	0.55	0.909	0.922
PBAS [89]	8.57	0.959	0.997	0.003	0.041	0.486	0.924	0.894
SGMM-SOD [104]	8.71	0.936	0.997	0.003	0.064	0.558	0.917	0.902
CDPS (à paraître)	9.57	0.949	0.997	0.004	0.051	0.624	0.921	0.897
ViBe [41]	9.71	0.82	0.998	0.002	0.18	0.887	0.87	0.929
Histogramme [26]	10.0	0.878	0.997	0.003	0.122	0.668	0.9	0.925
ViBe+ [92]	11.14	0.828	0.997	0.003	0.172	0.963	0.872	0.926
Bayesian Background [50]	11.29	0.733	0.998	0.002	0.267	0.904	0.827	0.962
Distance de Mahalonabis [58]	12.29	0.887	0.996	0.004	0.113	0.729	0.895	0.907
Chebyshev prob. SOD [90]	13.29	0.827	0.997	0.003	0.173	0.83	0.865	0.914
KNN [32]	13.43	0.793	0.998	0.002	0.207	1.284	0.841	0.924
Distance euclidienne [58]	14.29	0.839	0.996	0.004	0.162	1.026	0.872	0.911
GMM - KaewTraKulPong [31]	14.57	0.586	0.999	0.001	0.414	1.938	0.712	0.953
GMM - Zivkovic [88]	15.57	0.808	0.997	0.003	0.192	1.33	0.838	0.899
Local-Self similarity [94]	15.57	0.973	0.987	0.013	0.027	1.335	0.849	0.756
SGMM [96]	16.14	0.868	0.995	0.005	0.132	1.244	0.859	0.858
GMM - Riahi [95]	16.29	0.667	0.998	0.002	0.333	1.534	0.75	0.917
UBA [98]	17.43	0.902	0.991	0.009	0.098	1.017	0.813	0.742
GMM - Stauffer [33]	19.29	0.818	0.995	0.005	0.182	1.532	0.825	0.846
KDE - Nonaka [93]	20.86	0.747	0.995	0.005	0.253	1.806	0.739	0.8
KDE - Yoshinaga (à paraître)	21.43	0.755	0.994	0.006	0.245	1.915	0.755	0.783
Quasi-Continuous Histograms [99]	23.86	0.704	0.992	0.008	0.296	2.214	0.662	0.7

TABLE 3.5 – Résultats pour la catégorie « Arrière-plan dynamique » selon les méthodes appliquées sur la banque de données CDnet et soumises sur le site CDnet.

Méthodes	$Cc_{i,c}$	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
Chebyshev prob. [90]	4.57	0.818	0.998	0.002	0.182	0.344	0.766	0.763
Chebyshev prob. SOD [90]	5.57	0.818	0.998	0.002	0.182	0.409	0.752	0.734
ViBe+ [92]	8.14	0.762	0.998	0.002	0.238	0.384	0.72	0.729
PBAS [89]	8.43	0.696	0.999	0.001	0.304	0.539	0.683	0.833
KNN [32]	9.14	0.805	0.994	0.006	0.195	0.806	0.686	0.693
CDPS (à paraître)	9.14	0.759	0.995	0.005	0.241	0.728	0.75	0.809
KDE - Yoshinaga (à paraître)	9.29	0.893	0.991	0.009	0.106	1.014	0.657	0.589
GMM - KaewTraKulPong [31]	9.86	0.63	0.998	0.002	0.37	0.54	0.67	0.77
SGMM-SOD [104]	10.0	0.754	0.996	0.004	0.246	0.612	0.686	0.739
PSP-MRF [91]	10.14	0.895	0.986	0.014	0.104	1.451	0.696	0.658
SGMM [96]	12.14	0.771	0.993	0.007	0.229	0.913	0.638	0.666
KDE - Nonaka [93]	12.29	0.84	0.991	0.009	0.16	1.15	0.602	0.541
Quasi-Continuous Histograms [99]	12.29	0.891	0.99	0.01	0.109	1.13	0.643	0.535
GMM - Stauffer [33]	12.86	0.834	0.99	0.01	0.166	1.208	0.633	0.599
SC-SOBS [44]	13.29	0.892	0.984	0.016	0.108	1.69	0.669	0.628
SOBS [43]	13.86	0.88	0.984	0.016	0.12	1.637	0.644	0.586
GMM - Zivkovic [88]	13.86	0.802	0.99	0.01	0.198	1.173	0.633	0.621
Bayesian Background [50]	16.29	0.596	0.992	0.008	0.404	1.243	0.537	0.69
KDE - ElGammal [35]	17.29	0.801	0.986	0.014	0.199	1.639	0.596	0.573
ViBe [41]	18.0	0.722	0.99	0.01	0.278	1.28	0.565	0.535
Local-Self similarity [94]	18.86	0.898	0.769	0.231	0.102	22.787	0.095	0.052
Distance de Mahalonabis [58]	19.14	0.813	0.97	0.03	0.187	3.141	0.526	0.452
Distance euclidienne [58]	20.43	0.776	0.971	0.029	0.224	3.01	0.508	0.449
Histogramme [26]	20.86	0.807	0.94	0.06	0.193	6.049	0.243	0.152
GMM - Riahi [95]	21.14	0.478	0.984	0.016	0.522	1.974	0.43	0.648
Color Histogram Backproj. [97]	24.14	0.631	0.891	0.109	0.369	11.049	0.268	0.198

TABLE 3.6 – Résultats pour la catégorie « Mouvements de la caméra » selon les méthodes appliquées sur la banque de données CDnet et soumises sur le site CDnet.

Méthodes	$Cc_{i,c}$	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
PSP-MRF [91]	4.43	0.821	0.983	0.018	0.179	2.278	0.75	0.701
ViBe+ [92]	4.86	0.729	0.991	0.009	0.271	1.847	0.754	0.806
PBAS [89]	6.29	0.737	0.984	0.016	0.263	2.488	0.722	0.759
KDE - Yoshinaga (à paraître)	7.57	0.756	0.982	0.018	0.244	2.745	0.712	0.679
KDE - Nonaka [93]	7.71	0.732	0.986	0.014	0.268	2.424	0.711	0.699
SGMM-SOD [104]	7.71	0.631	0.992	0.008	0.369	2.163	0.699	0.827
SOBS [43]	8.0	0.801	0.979	0.021	0.199	2.748	0.709	0.64
SGMM [96]	8.0	0.709	0.987	0.013	0.291	2.376	0.725	0.775
SC-SOBS [44]	8.71	0.811	0.977	0.023	0.189	2.879	0.705	0.629
KNN [32]	10.14	0.735	0.978	0.022	0.265	3.11	0.689	0.702
Bayesian Background [50]	12.14	0.544	0.989	0.011	0.456	2.881	0.599	0.668
GMM - KaewTraKulPong [31]	12.29	0.507	0.989	0.011	0.493	3.023	0.576	0.69
Chebyshev prob. SOD [90]	13.29	0.722	0.973	0.028	0.278	3.62	0.642	0.596
GMM - Stauffer [33]	14.29	0.733	0.967	0.033	0.267	4.227	0.597	0.513
ViBe [41]	14.86	0.711	0.969	0.031	0.289	4.015	0.6	0.529
KDE - ElGammal [35]	14.86	0.738	0.956	0.044	0.263	5.135	0.572	0.486
GMM - Riahi [95]	15.43	0.765	0.95	0.05	0.235	5.666	0.537	0.418
Color Histogram Backproj. [97]	16.57	0.469	0.982	0.018	0.531	3.717	0.482	0.53
Local-Self similarity [94]	17.43	0.976	0.616	0.384	0.024	36.957	0.207	0.12
GMM - Zivkovic [88]	17.57	0.69	0.967	0.034	0.31	4.406	0.567	0.487
Distance de Mahalonabis [58]	17.71	0.736	0.943	0.057	0.264	6.439	0.496	0.381
Distance euclidienne [58]	19.29	0.712	0.946	0.054	0.288	6.296	0.487	0.375
CDPS (à paraître)	19.57	0.603	0.961	0.039	0.398	5.359	0.486	0.44
Histogramme [26]	21.29	0.711	0.841	0.159	0.289	16.28	0.278	0.175

TABLE 3.7 – Résultats pour la catégorie « Mouvement intermittent des objets » selon les méthodes appliquées sur la banque de données CDnet et soumises sur le site CDnet.

Méthodes	$Cc_{i,c}$	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
SGMM-SOD [104]	4.86	0.72	0.983	0.017	0.28	3.05	0.687	0.774
UBA [98]	5.43	0.721	0.983	0.017	0.28	3.054	0.689	0.731
CDPS (à paraître)	5.43	0.808	0.977	0.024	0.192	3.465	0.741	0.762
KDE - Nonaka [93]	8.29	0.451	0.996	0.004	0.549	4.419	0.545	0.817
PBAS [89]	9.43	0.67	0.975	0.025	0.33	4.287	0.575	0.705
SC-SOBS [44]	9.86	0.724	0.961	0.039	0.276	5.221	0.592	0.59
SGMM [96]	10.0	0.501	0.985	0.015	0.499	4.918	0.54	0.699
GMM - Stauffer [33]	10.43	0.514	0.984	0.017	0.486	5.196	0.521	0.669
KDE - Yoshinaga (à paraître)	11.29	0.437	0.992	0.008	0.563	4.7	0.504	0.721
KNN [32]	11.57	0.462	0.987	0.013	0.538	5.137	0.503	0.712
ViBe+ [92]	12.43	0.473	0.982	0.018	0.527	5.428	0.509	0.751
PSP-MRF [91]	13.0	0.701	0.953	0.047	0.299	6.059	0.565	0.573
GMM - Zivkovic [88]	13.14	0.547	0.971	0.029	0.453	5.499	0.532	0.646
SOBS [43]	13.71	0.706	0.951	0.049	0.294	6.132	0.563	0.553
GMM - KaewTraKulPong [31]	14.86	0.348	0.989	0.011	0.652	5.985	0.39	0.695
GMM - Riahi [95]	14.86	0.219	0.998	0.002	0.781	5.255	0.315	0.585
ViBe [41]	15.86	0.512	0.953	0.047	0.488	7.743	0.507	0.651
Local-Self similarity [94]	16.0	0.903	0.822	0.178	0.097	15.883	0.533	0.445
Histogramme [26]	16.0	0.751	0.866	0.134	0.249	13.136	0.511	0.486
Chebyshev prob. SOD [90]	16.14	0.357	0.981	0.019	0.643	6.47	0.386	0.769
Quasi-Continuous Histograms [99]	17.14	0.441	0.98	0.02	0.559	6.049	0.437	0.538
Distance euclidienne [58]	17.29	0.592	0.934	0.066	0.408	8.998	0.489	0.499
Distance de Mahalonabis [58]	17.86	0.717	0.889	0.111	0.283	11.534	0.497	0.454
KDE - ElGammal [35]	19.71	0.503	0.931	0.069	0.496	10.069	0.409	0.461
Bayesian Background [50]	20.43	0.481	0.93	0.07	0.519	9.963	0.408	0.474

TABLE 3.8 – Résultats pour la catégorie « Ombres » selon les méthodes appliquées sur la banque de données CDnet et soumises sur le site CDnet.

Méthodes	$Cc_{i,c}$	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
SGMM-SOD [104]	4.14	0.918	0.99	0.01	0.082	1.258	0.861	0.819
PBAS [89]	4.57	0.913	0.99	0.01	0.087	1.275	0.86	0.814
Chebyshev prob. [90]	7.29	0.867	0.989	0.011	0.133	1.555	0.833	0.81
ViBe+ [92]	7.57	0.811	0.991	0.009	0.189	1.652	0.815	0.83
Chebyshev prob. SOD [90]	8.0	0.867	0.989	0.011	0.133	1.556	0.833	0.81
ViBe [41]	8.14	0.783	0.992	0.008	0.217	1.655	0.803	0.834
SGMM [96]	9.71	0.858	0.989	0.011	0.142	1.796	0.794	0.762
CDPS (à paraître)	9.86	0.923	0.985	0.015	0.077	1.952	0.809	0.757
KDE - Nonaka [93]	10.57	0.72	0.993	0.007	0.28	2.129	0.754	0.824
KDE - ElGammal [35]	11.0	0.854	0.989	0.011	0.146	1.688	0.803	0.766
KNN [32]	11.86	0.748	0.992	0.008	0.252	2.057	0.747	0.779
GMM - KaewTraKulPong [31]	12.71	0.632	0.994	0.006	0.368	2.301	0.718	0.858
PSP-MRF [91]	13.57	0.874	0.983	0.017	0.126	2.241	0.791	0.728
SC-SOBS [44]	15.29	0.85	0.983	0.017	0.15	2.3	0.779	0.723
SOBS [43]	15.86	0.835	0.984	0.016	0.165	2.337	0.772	0.722
Bayesian Background [50]	15.86	0.654	0.992	0.008	0.346	2.47	0.696	0.779
GMM - Stauffer [33]	16.29	0.796	0.987	0.013	0.204	2.195	0.737	0.716
GMM - Riahi [95]	16.43	0.719	0.989	0.011	0.281	2.411	0.733	0.784
GMM - Zivkovic [88]	16.86	0.777	0.988	0.012	0.223	2.196	0.732	0.723
KDE - Yoshinaga (à paraître)	16.86	0.697	0.99	0.01	0.303	2.486	0.714	0.756
UBA [98]	17.71	0.908	0.971	0.029	0.092	3.225	0.712	0.61
Quasi-Continuous Histograms [99]	18.57	0.695	0.989	0.011	0.305	2.587	0.707	0.738
Local-Self similarity [94]	18.86	0.958	0.944	0.056	0.042	5.55	0.595	0.467
Distance euclidienne [58]	20.14	0.8	0.978	0.022	0.2	2.899	0.678	0.611
Histogramme [26]	21.14	0.831	0.969	0.031	0.169	3.71	0.659	0.601
Distance de Mahalonabis [58]	22.14	0.784	0.971	0.029	0.215	3.79	0.635	0.569

TABLE 3.9 – Résultats pour la catégorie « Thermique » selon les méthodes appliquées sur la banque de données CDnet et soumises sur le site CDnet.

Méthodes	$Cc_{i,c}$	Ra	Sp	T_{FP}	T_{FN}	PMC	fm	Pr
Chebyshev prob. [90]	7.0	0.694	0.996	0.004	0.306	1.329	0.726	0.891
Chebyshev prob. SOD [90]	7.14	0.689	0.996	0.004	0.311	1.428	0.723	0.891
KDE - ElGammal [35]	9.0	0.672	0.996	0.004	0.328	1.679	0.742	0.897
SGMM-SOD [104]	9.14	0.594	0.997	0.003	0.406	1.893	0.695	0.952
PBAS [89]	9.29	0.728	0.993	0.007	0.272	1.54	0.756	0.892
PSP-MRF [91]	10.29	0.599	0.996	0.004	0.401	1.919	0.693	0.922
ViBe+ [92]	11.14	0.541	0.997	0.003	0.459	2.82	0.665	0.948
CDPS (à paraître)	12.0	0.62	0.995	0.005	0.381	1.52	0.662	0.901
UBA [98]	12.29	0.688	0.994	0.006	0.312	1.668	0.728	0.796
SC-SOBS [44]	12.71	0.6	0.996	0.004	0.4	1.984	0.692	0.886
Bayesian Background [50]	13.14	0.603	0.995	0.005	0.397	2.868	0.697	0.888
ViBe [41]	13.43	0.543	0.996	0.004	0.457	3.127	0.665	0.936
SGMM [96]	13.57	0.536	0.997	0.003	0.464	3.939	0.648	0.926
GMM - KaewTraKulPong [31]	14.14	0.34	0.999	0.001	0.66	4.842	0.477	0.971
Local-Self similarity [94]	14.14	0.904	0.969	0.031	0.096	3.261	0.73	0.643
Histogramme [26]	14.43	0.641	0.993	0.007	0.359	1.967	0.7	0.811
SOBS [43]	14.57	0.589	0.996	0.004	0.411	2.098	0.683	0.875
GMM - Riahi [95]	15.29	0.246	0.999	0.001	0.754	5.266	0.368	0.962
Distance de Mahalonabis [58]	15.43	0.627	0.991	0.009	0.373	2.346	0.707	0.862
KNN [32]	15.43	0.482	0.997	0.003	0.518	4.378	0.605	0.919
KDE - Nonaka [93]	15.71	0.415	0.998	0.002	0.585	5.415	0.499	0.916
Quasi-Continuous Histograms [99]	17.43	0.335	0.998	0.002	0.665	5.149	0.465	0.878
KDE - Yoshinaga (à paraître)	17.57	0.406	0.997	0.003	0.594	5.153	0.52	0.876
GMM - Stauffer [33]	17.86	0.569	0.995	0.005	0.431	4.264	0.662	0.865
GMM - Zivkovic [88]	18.71	0.554	0.994	0.006	0.446	4.3	0.655	0.871
Distance euclidienne [58]	20.14	0.511	0.991	0.009	0.489	3.852	0.631	0.887

TABLE 3.10 – Taux de faux positifs sur l'ombre, T_{FPO}, selon différentes méthodes.

Méthodes	T_{FPO}
Bayesian Background [50]	0.329
Quasi-Continuous Histograms [99]	0.348
KDE - Nonaka [93]	0.39
KNN [32]	0.398
KDE - Yoshinaga (à paraître)	0.398
GMM - KaewTraKulPong [31]	0.407
Chebyshev prob. [90]	0.42
Chebyshev prob. SOD [90]	0.42
GMM - Riahi [95]	0.476
SGMM [96]	0.486
ViBe+ [92]	0.531
GMM - Stauffer [33]	0.535
GMM - Zivkovic [88]	0.543
ViBe [41]	0.546
SOBS [43]	0.569
Distance euclidienne [58]	0.576
PBAS [89]	0.579
PSP-MRF [91]	0.586
Histogramme [26]	0.588
CDPS (à paraître)	0.59
Distance de Mahalonabis [58]	0.59
SGMM-SOD [104]	0.601
SC-SOBS [44]	0.604
UBA [98]	0.615
KDE - ElGammal [35]	0.622
Local-Self similarity [94]	0.638

Bibliographie

[1] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. Changedetection. net : A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2012.

[2] Valérie Gouaillier. La vidéosurveillance intelligente : promesses et défis. Technical report, Centre de recherche informatique de Montréal, 2009.

[3] H.M. Dee and S.A. Velastin. How close are we to solving the problem of automated visual surveillance ? *Machine Vision and Applications*, 19(5) :329–343, 2008.

[4] A. Hampapur, L. Brown, J. Connell, S. Pankanti, A. Senior, and Y. Tian. Smart surveillance : applications, technologies and implications. In *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, volume 2, pages 1133–1138. IEEE, 2003.

[5] X. Lu, M. Jin, S. Yu, L. Wang, and H. Lu. A real-time anomaly intrusion and theft items detecting system for surveillance videos. In *Audio Language and Image Processing (ICALIP), 2010 International Conference on*, pages 1217–1221. IEEE, 2010.

[6] A.M. Ibrahim, AA Shafie, and MM Rashid. Human identification system based on moment invariant features. In *Computer and Communication Engineering (ICCCE), 2012 International Conference on*, pages 216–221. IEEE, 2012.

[7] M. Brulin, H. Nicolas, and C. Maillet. Video surveillance traffic analysis using scene geometry. In *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, pages 450–455. IEEE, 2010.

[8] AW Senior, L. Brown, A. Hampapur, C.F. Shu, Y. Zhai, RS Feris, Y.L. Tian, S. Borger, and C. Carlson. Video analytics for retail. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, pages 423–428. IEEE, 2007.

[9] V.B. Subburaman, A. Descamps, and C. Carincotte. Counting people in the crowd using a generic head detector. *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 470–475, 2012.

[10] M. Bozzoli, L. Cinque, and E. Sangineto. A statistical method for people counting in crowded environments. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 506–511. IEEE, 2007.

[11] G. Castanon, V. Saligrama, A.L. Caron, and P.M. Jodoin. Real-time activity search of surveillance video. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 246–251. IEEE, 2012.

[12] A.D. Bagdanov, M. Bertini, A. Del Bimbo, and L. Seidenari. Adaptive video compression for video surveillance applications. In *Multimedia (ISM), 2011 IEEE International Symposium on*, pages 190–197. IEEE, 2011.

[13] marketsandmarkets.com. Video surveillance market - global forecast & analysis. Technical report, Markets and Markets, 2011.

[14] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10) :1337–1342, 2003.

[15] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, pages 627–630. IEEE, 2000.

[16] Q. Zhou and J.K. Aggarwal. Tracking and classifying moving objects from video. In *Proceedings of IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, pages 52–59. Hawaii, USA, 2001.

[17] Q. Zhu, S. Avidan, and K.T. Cheng. Learning a sparse, corner-based representation for time-varying background modelling. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 678–685. IEEE, 2005.

[18] S.Y. Chien, S.Y. Ma, and L.G. Chen. Efficient moving object segmentation algorithm using background registration technique. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(7) :577–586, 2002.

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

[20] M. Seki, H. Fujiwara, and K. Sumi. A robust background subtraction method for changing background. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pages 207–213. IEEE, 2000.

[21] A. Srivastava, A.B. Lee, E.P. Simoncelli, and S.C. Zhu. On advances in statistical modeling of natural images. *Journal of mathematical imaging and vision*, 18(1) :17–33, 2003.

[22] J. Heikkila and O. Silvén. A real-time system for monitoring of cyclists and pedestrians. In *Visual Surveillance, 1999. Second IEEE Workshop on,(VS'99)*, pages 74–81. IEEE, 1999.

[23] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. Probabilistic posture classification for human-behavior analysis. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 35(1) :42–54, 2005.

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

[25] S. Greenhill, S. Venkatesh, and G. West. Adaptive model for foreground extraction in adverse lighting conditions. *PRICAI 2004 : Trends in Artificial Intelligence*, pages 805–811, 2004.

[26] J. Zheng, Y. Wang, N.L. Nihan, and M.E. Hallenbeck. Extracting roadway background image : Mode-based approach. *Transportation Research Record : Journal of the Transportation Research Board*, 1944(-1) :82–88, 2006.

[27] S.S. Ghidary, Y. Nakata, T. Takamori, and M. Hattori. Human detection and localization at indoor environment by home robot. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 2, pages 1360–1365. IEEE, 2000.

[28] S.C.S. Cheung and C. Kamath. Robust techniques for background subtraction in urban traffic video. In *Proceedings of SPIE*, volume 5308, pages 881–892, 2004.

[29] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder : Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7) :780–785, 1997.

[30] N. Friedman and S. Russell. Image segmentation in video sequences : A probabilistic approach. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc., 1997.

[31] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, volume 25, pages 1–5, 2001.

[32] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7) :773–780, 2006.

[33] Stauffer C. and Grimson W.E.L. Adaptive background mixture models for real-time tracking. *international Conference on Computer Vision and Pattern Recognition*, 2 :246–252, 1999.

[34] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3) :172–185, 2005.

[35] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *Computer Vision—ECCV 2000*, pages 751–767, 2000.

[36] L. Lacassagne, A. Manzanera, and A. Dupret. Motion detection : Fast and robust algorithms for embedded systems. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3265–3268. IEEE, 2009.

[37] S. Toral, M. Vargas, F. Barrero, and MG Ortega. Improved sigma-delta background estimation for vehicle detection. *Electronics Letters*, 45(1) :32–34, 2009.

[38] L. Lacassagne, A. Manzanera, J. Denoulet, and A. Mérigot. High performance motion detection : some trends toward new embedded architectures for vision systems. *Journal of Real-Time Image Processing*, 4(2) :127–146, 2009.

[39] H. Wang and D. Suter. Background subtraction based on a robust consensus method. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 223–226. IEEE, 2006.

[40] H. Wang and D. Suter. A consensus-based method for tracking : Modelling background scenario and foreground appearance. *Pattern Recognition*, 40(3) :1091–1105, 2007.

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

[42] H.H. Lin, T.L. Liu, and J.H. Chuang. Learning a scene background model via classification. *Signal Processing, IEEE Transactions on*, 57(5) :1641–1654, 2009.

[43] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *Image Processing, IEEE Transactions on*, 17(7) :1168–1177, 2008.

[44] L. Maddalena and A. Petrosino. The sobs algorithm : what are the limits ? In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 21–26. IEEE, 2012.

[45] I. Haritaoglu, D. Harwood, and L.S. Davis. W< sup> 4</sup> : real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8) :809–830, 2000.

[46] M. Seki, T. Wada, H. Fujiwara, and K. Sumi. Background subtraction based on cooccurrence of image variations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–65. IEEE, 2003.

[47] D.M. Tsai and S.C. Lai. Independent component analysis-based background subtraction for indoor surveillance. *Image Processing, IEEE Transactions on*, 18(1) :158–167, 2009.

[48] C. Ridder, O. Munkelt, and H. Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of International Conference on recent Advances in Mechatronics*, pages 193–199. Citeseer, 1995.

[49] J. Zhong and S. Sclaroff. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 44–50. IEEE, 2003.

[50] F. Porikli and C. Wren. Change detection by frequency decomposition : Wave-back. In *Proc. of Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[51] W. Wang, J. Yang, and W. Gao. Modeling background and segmenting moving objects from compressed video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(5) :670–681, 2008.

[52] P. Meer. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5) :564–577, 2003.

[53] P. Withagen, F. Groen, and K. Schutte. Emswitch : A multi-hypothesis approach to em background modeling. *Proceedings of the IEEE Advanced Concepts for Intelligent Vision Systems, ACIVS*, 1 :199–206, 2003.

[54] J. Lindstrom, F. Lindgren, K. Ltrstrom, J. Holst, and U. Holst. Background and foreground modeling using an online em algorithm. *IEEE Int Workshop on Visual Surveillance VS 2006 in conjunction with ECCV 20 06*, pages 9–16, 2006.

[55] JL Landabaso and M. Pardas. Cooperative background modelling using multiple cameras towards human detection in smart-rooms. In *invited paper). In Proceedings of European Signal Processing Conference*, 2006.

[56] J. Migdal and W.E.L. Grimson. Background subtraction using markov thresholds. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, volume 2, pages 58–65. IEEE, 2005.

[57] D.H. Parks and S.S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pages 192–199. IEEE, 2008.

[58] Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3) :033003–033003, 2010.

[59] S. Brutzer, B. Hoferlin, and G. Heidemann. Evaluation of background subtraction techniques for video surveillance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1937–1944. IEEE, 2011.

[60] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli. Real-time discriminative background subtraction. *Image Processing, IEEE Transactions on*, 20(5) :1401–1414, 2011.

[61] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9) :1124–1137, 2004.

[62] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1) :7–42, 2002.

[63] D. Lopresti and G. Nagy. Issues in ground-truthing graphic documents. *Graphics Recognition Algorithms and Applications*, pages 46–67, 2002.

[64] D.P. Young and J.M. Ferryman. Pets metrics : On-line performance evaluation service. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 317–324. IEEE, 2005.

[65] Compétition VSSN 2006. `http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC/`. Visité le 2 janvier 2013.

[66] Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[67] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower : Principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261. IEEE, 1999.

[68] L. Li, W. Huang, I.Y.H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *Image Processing, IEEE Transactions on*, 13(11) :1459–1472, 2004.

[69] M. Karaman, L. Goldmann, D. Yu, and T. Sikora. Comparison of static background segmentation methods. In *Proc. SPIE*, volume 5960, pages 2140–2151, 2005.

[70] P. Yin, A. Criminisi, J. Winn, and M. Essa. Tree-based classifiers for bilayer video segmentation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[71] F. Tiburzi, M. Escudero, J. Bescós, and J.M. Martínez. A ground truth for motion-based video-object segmentation. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 17–20. IEEE, 2008.

[72] R. Vezzani and R. Cucchiara. Annotation collection and online performance evaluation for video surveillance : The visor project. In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pages 227–234. IEEE, 2008.

[73] T. Ellis. Performance metrics and methods for tracking in surveillance. In *Proceedings of the 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS'02)*, pages 26–31. Citeseer, 2002.

[74] J. Davis and M. Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[75] T. Bouwmans, F. El Baf, B. Vachon, et al. Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents on Computer Science*, 1(3) :219–237, 2008.

[76] T. Bouwmans. Recent advanced statistical background modeling for foreground detection : A systematic survey. *Recent Patents on Computer Science*, 4(3) :147–176, 2011.

[77] L.M. Brown, A.W. Senior, Y. Tian, J. Connell, A. Hampapur, C.F. Shu, H. Merkl, and M. Lu. Performance evaluation of surveillance systems under varying conditions. In *Proceedings of IEEE PETS Workshop*, pages 1–8. Citeseer, 2005.

[78] P.L. Rosin and E. Ioannidis. Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, 24(14) :2345–2356, 2003.

[79] T.H. Chalidabhongse, K. Kim, D. Harwood, and L. Davis. A perturbation method for evaluating background subtraction algorithms. In *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 172–185, 2003.

[80] S. Panahi, S. Sheikhi, S. Hadadan, and N. Gheissari. Evaluation of background subtraction methods. In *Computing : Techniques and Applications, 2008. DICTA'08. Digital Image*, pages 357–364. IEEE, 2008.

[81] S. Herrero and J. Bescós. Background subtraction techniques : Systematic evaluation and comparative analysis. In *Advanced Concepts for Intelligent Vision Systems*, pages 33–42. Springer, 2009.

[82] J. Aguilera, H. Wildenauer, M. Kampel, M. Borg, D. Thirde, and J. Ferryman. Evaluation of motion segmentation quality for aircraft activity surveillance. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 293–300. IEEE, 2005.

[83] N. Lazarevic-McManus, J. Renno, D. Makris, and GA Jones. Designing evaluation methodologies : the case of motion detection. In *Proceedings of 9th IEEE International Workshop on PETS*, pages 23–30. Citeseer, 2006.

[84] A. Senior, A. Hampapur, Y.L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handling. *Image and Vision Computing*, 24(11) :1233–1243, 2006.

[85] F. Bashir and F. Porikli. Performance evaluation of object detection and tracking systems. *In PETS*, 6 :7–14, 2006.

[86] J.C. Nascimento and J.S. Marques. Performance evaluation of object detection algorithms for video surveillance. *Multimedia, IEEE Transactions on*, 8(4) :761–774, 2006.

[87] P. Villegas and X. Marichal. Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *Image Processing, IEEE Transactions on*, 13(8) :1092–1103, 2004.

[88] 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. IEEE, 2004.

[89] M. Hofmann, P. Tiefenbacher, and G. Rigoll. Background segmentation with feedback : The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 38–43. IEEE, 2012.

[90] A. Morde, X. Ma, and S. Guler. Learning a background model for change detection. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 15–20. IEEE, 2012.

[91] A. Schick, M. Bauml, and R. Stiefelhagen. Improving foreground segmentations with probabilistic superpixel markov random fields. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 27–31. IEEE, 2012.

[92] M. Van Droogenbroeck and O. Paquot. Background subtraction : Experiments and improvements for vibe. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 32–37. IEEE, 2012.

[93] Y. Nonaka, A. Shimada, H. Nagahara, and R. Taniguchi. Evaluation report of integrated background modeling based on spatio-temporal features. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 9–14. IEEE, 2012.

[94] J.P. Jodoin, G.A. Bilodeau, and N. Saunier. Background subtraction based on local shape. Technical report, 2012.

[95] D. Riahi, P. St-Onge, and G. Bilodeau. Rectgauss-tex : Blockbased background subtraction. Technical report, Technical Report EPM-RT-2012-03, Ecole Polytechnique de Montreal, 2012. 6, 7, 8, 2012.

[96] R.H. Evangelio, M. Patzold, and T. Sikora. Splitting gaussians in mixture models. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 300–305. IEEE, 2012.

[97] D. Kit, B. Sullivan, and D. Ballard. Novelty detection using growing neural gas for visuo-spatial memory. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1194–1200. IEEE, 2011.

[98] D. Park and H. Byun. Object-wise multilayer background ordering for public area surveillance. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 484–489. IEEE, 2009.

[99] D.D. Sidibe, O. Strauss, W. Puech, et al. Automatic background generation from a sequence of images based on robust mode estimation. *Proc. SPIE 7250*, 2009.

[100] T.S.F. Haines and T. Xiang. Background subtraction with dirichlet processes. 2012.

[101] L. Lam and SY Suen. Application of majority voting to pattern recognition : An analysis of its behavior and performance. *Systems, Man and Cybernetics, Part A : Systems and Humans, IEEE Transactions on*, 27(5) :553–568, 1997.

[102] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms : Bagging, boosting, and variants. *Machine learning*, 36(1) :105–139, 1999.

[103] I. 'Borg and P.' Groenen. *'Modern Multidimensional Scaling : theory and applications'*. 'New York : Springer-Verlag', 2 edition, 2005.

[104] R.H. Evangelio and T. Sikora. Complementary background models for the detection of static and moving objects in crowded environments. In *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, pages 71–76. IEEE, 2011.