

# Méthode d'apprentissage non supervisée pour la détection d'évènements inattendus

B. Luvison<sup>1,2</sup>

T. Chateau<sup>1</sup>

P. Sayd<sup>2</sup>

Q.C. Pham<sup>2</sup>

<sup>1</sup> LASMEA, Université Blaise Pascal, Aubière, France

<sup>2</sup> CEA, LIST, LSVE, Boîte Courrier 94, F-91191 Gif-sur-Yvette, France

bertrand.luvison@lasmea.univ-bpclermont.fr

## Résumé

*Cet article présente une méthode générique non supervisée pour la détection d'évènements depuis une caméra fixe non calibrée. Le système est formalisé dans un cadre probabiliste et la détection est effectuée à partir de fonctions de vraisemblance. Nous proposons une méthode originale, basée apprentissage, pour approximer la densité de probabilité de la fonction de vraisemblance en utilisant un modèle parcimonieux basé sur des fonctions noyaux. Ce modèle présente l'avantage des méthodes non-paramétriques tout en limitant le coût algorithmique souvent important qui leur est lié. Appliqué à un ensemble de directions de flots optiques dans chaque zone d'une grille régulière de l'image, le modèle conduit à l'estimation des zones dans lesquelles la direction du flot n'est pas conforme avec celles d'une série de séquences d'apprentissage. La méthode est testée dans le cas de la détection d'évènements anormaux liés au trafic routier.*

## Mots clefs

Apprentissage du mouvement, machine à noyau, vidéo surveillance.

## 1 Introduction

La surveillance est essentiellement réalisée par des agents de sécurité inspectant des flux vidéo issus de réseaux de caméras. Mais, dans des lieux publics toujours plus fréquentés et contrôlés, le nombre de caméras de surveillance ne cesse d'augmenter. Face à ce phénomène, des systèmes temps réel d'aide à la surveillance focalisant l'opérateur sur des évènements inattendus sont nécessaires. Ce type de système doit pouvoir gérer une grosse masse d'information (mouvement d'une foule importante) pour en extraire des évènements très ponctuels et généralement rares.

L'analyse de foule (au sens large, à savoir multitude d'objet en un même lieu) est d'autant plus délicate que la variété des évènements à détecter est grande. Afin de limiter l'éventail des évènements à détecter, les chercheurs supposent généralement des scénarios bien spécifiques tel que la détection de trajectoires anormales, la chute de personne, les mouvements de panique, les trajectoires dangereuses,

les vitesses inadaptées, etc... Le plus souvent cette détection d'évènements anormaux peut être vue comme un problème de classification, où une image ou une séquence vidéo doit être classifiée dans l'une des deux catégories : attendue ou inattendue. L'état de l'art sur la question peut être divisé en deux catégories : les approches basées modèle et les approches basées apprentissage.

Pour analyser la foule, certaines approches nécessitent des analyses et des traitements préalables pour extraire des primitives plus élaborées habituellement extraites à partir d'approches basées modèle, comme les personnes d'une foule ou leurs trajectoires. Dans [1], un modèle géométrique (des ellipses ou une forme oméga) est utilisé pour détecter les différentes parties du corps (corps et jambes ou épaules et tête). Les formes segmentées sont ensuite suivies afin de définir des trajectoires. D'autres approches supposent que les différents points appartenant à une même personne suivent la même trajectoire [2] [3]. Au lieu de segmenter la foule pour ensuite étudier les personnes individuellement, certaines approches proposent des analyses du mouvement de la scène dans sa globalité. Dans [4], un modèle de déplacement est construit à partir d'hypothèses sur les sens de circulation ou les zones d'arrêt. Le mouvement est estimé à l'aide de méthodes de flot optique auxquelles est associée une segmentation fond/forme. Ces approches basées modèle fonctionnent généralement sur un type d'environnement bien spécifique et nécessitent une expertise et des connaissances préalables qui ne sont pas toujours disponibles.

Les approches basées apprentissage visent à déduire les modèles utilisés pour la détection à partir d'une base. Cette base d'apprentissage peut être construite à partir de trajectoires d'objets [5, 3] pour ensuite détecter les trajectoires inattendues par des méthodes de classification. Les trajectoires peuvent aussi inférer les interactions entre les individus (personnes se séparant, se rapprochant, se battant, ...) [6]. Dans [7], ce sont les composantes principales (extraites d'une décomposition SVD) du mouvement qui sont apprises en utilisant une Chaîne de Markov Cachée.

Nous proposons un système basé sur une méthode générique non supervisée pour détecter les mouvements inat-

tendus sur des zones structurées (type surveillance routière ou de foule) à partir de caméras fixes. Dans l'optique d'être le plus générique possible, une approche basée apprentissage a été choisie.

Habituellement les problèmes de classification basée apprentissage utilisent une base d'exemples positifs et d'exemples négatifs. Les machines les plus populaires sont alors les machines à vecteurs supports (SVM) [8], les machines à vecteurs de pertinence (RVM) [9] et le Boosting [10]. Néanmoins, dans le cas de la détection d'événements anormaux, la construction de la base associée à ces derniers est un challenge car : (1) ce sont des événements rares et (2) la variabilité intra-classe est importante.

Nous proposons une méthode qui n'utilise que les exemples d'événements normaux dans l'apprentissage. Cette dernière, complètement automatique ne nécessite aucune connaissance a priori. La partie suivante décrit le mécanisme d'apprentissage mis au point ainsi que son application à la détection d'événements inattendus. Les performances du système seront évaluées dans la partie 3, à la fois sur la qualité du modèle appris et sur les résultats obtenus en conditions réelles sur une longue durée de test.

## 2 Méthode Proposée

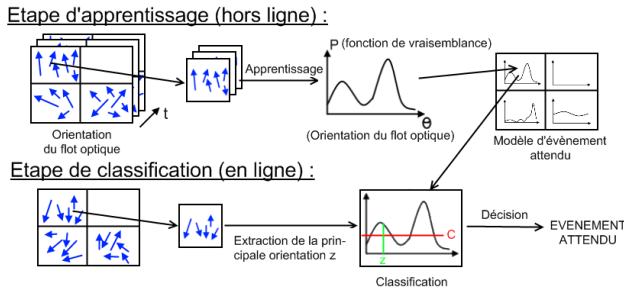


Figure 1 – Schéma du système.

La figure 1 illustre le fonctionnement de la méthode. Étant donné un flux vidéo issu d'une caméra fixe, le système est capable d'apprendre, à partir des orientations du flot optique cumulées au sein d'un bloc de l'image, une représentation non-paramétrique de la vraisemblance. La complexité de ce modèle dépendant directement de la taille de la base d'apprentissage, nous proposons une approche originale pour construire un modèle épars basé sur des fonctions noyaux. Ce modèle est obtenu par une sélection parcimonieuse des orientations les plus pertinentes afin d'approximer la fonction de vraisemblance. Enfin, une fonction discriminante est déduite de cette vraisemblance afin de classifier l'orientation principale de chaque bloc pendant une phase d'analyse en temps réel.

Cette méthode présente l'avantage d'être non supervisée : aucune calibration n'est nécessaire, aucune labélisation sur la base d'apprentissage n'est à réaliser. De plus, la représentation éparse de la vraisemblance est particulièrement adaptée pour effectuer une classification temps réel pen-

dant la phase d'analyse.

### 2.1 Fonction discriminante

Soit  $I$  la séquence vidéo d'apprentissage. Chaque image est divisée en  $N$  blocs et les orientations du flot optique sont associées à chaque bloc.  $\mathbf{Z}^{(n)} \doteq (z_1^{(n)}, z_2^{(n)}, \dots, z_K^{(n)})^T$  représente un vecteur composé de l'historique de toutes les orientations du flux optique (les primitives) pour un bloc. Nous définissons une variable aléatoire de Bernouilli  $\omega_k^{(n)} \in \{\omega_1; \omega_2\}$  pouvant prendre les deux états :  $\omega_k^{(n)} = \omega_1$  si l'évènement associé au bloc  $n$  est un évènement attendu et  $\omega_k^{(n)} = \omega_2$  si c'est un évènement inattendu. La fonction de vraisemblance  $P(z_k^{(n)} | \omega_2^{(n)})$  étant complètement inconnue, nous avons choisi de définir comme inattendue, toutes observations mal expliquées par la fonction de vraisemblance  $P(z_k^{(n)} | \omega_1^{(n)})$ . Les observations pour lesquelles les probabilités sont en dessous d'un seuil empirique  $C$  sont considérées comme inattendues.

$$\hat{\omega}^{(n)} = \omega_2^{(n)} \text{ si } P(z_k^{(n)} | \omega_1^{(n)}) < C \quad (1)$$

La vraisemblance  $P(z_k^{(n)} | \omega_1^{(n)})$  peut être représentée soit par une fonction paramétrique soit par un modèle non paramétrique.

Dans le premier cas, les modèles de mélange de gaussiennes (MMG) sont souvent utilisés pour approximer la vraisemblance. Par exemple, Stauffer et Grimson dans [11] utilisent un nombre fixe de gaussiennes pour modéliser une fonction de vraisemblance dans une application de soustraction de fond. Les récents algorithmes EM (Expectation Maximization) sont aussi utilisés pour estimer les paramètres des MMG sans hypothèse sur le nombre de gaussiennes [12]. Récemment, Han dans [13] propose une approche pour approximer une distribution par un MMG en s'appuyant sur l'algorithme mean-shift.

Les modèles non paramétriques approximent une distribution à partir d'exemples en utilisant une estimation basée sur des fonctions à noyau (modèle à fenêtre de Parzen [14]) :

$$\hat{P}(z_k^{(n)} | \omega_1^{(n)}) = K^{-1} \sum_{k'=1}^K \phi_{k'}^{(n)}(z_k^{(n)}) \quad (2)$$

où  $\phi_{k'}^{(n)}(z_k^{(n)})$  est une fonction noyau.

Dans le cadre de notre application, nous avons choisi d'opter pour ce dernier type de modèle, en faisant le choix classique d'utiliser des fonctions à base radiale,  $\phi_{k'}^{(n)}(z_k^{(n)}) = \frac{1}{\sigma_l \sqrt{2\pi}} \exp\{-\frac{d_\theta^2(z_k^{(n)}, z_{k'}^{(n)})}{2\sigma_l^2}\}$ . De plus,  $d_\theta(z_k^{(n)}, z_{k'}^{(n)})$  est une distance entre deux angles défini par :

$$d_\theta(\theta_1, \theta_2) = \frac{\min(|\theta_2 - \theta_1|, |\theta_2 - (\theta_1 + 2\pi)|)}{\pi}$$

$$\text{avec } \theta_1 < \theta_2$$

Néanmoins, cet estimateur de densité de probabilité a un coût algorithmique proportionnel au nombre de points de

l'ensemble d'apprentissage (taille de  $\mathbf{Z}^{(n)}$ ). Nous proposons d'approximer cet estimateur à l'aide d'un modèle épars obtenu à partir de l'équation (2) par une somme pondérée de fonctions noyaux.

## 2.2 Approximation de l'estimateur non paramétrique

L'équation 2 peut aussi être exprimée de manière réduite sous la forme :

$$\widehat{P}(z_k|\omega_1) \approx \mathbf{w}^T (\boldsymbol{\phi}(z_k)) \quad (3)$$

où  $\mathbf{w}^T = (1, \dots, 1)^T / K$  est un vecteur de taille  $K$  et  $\boldsymbol{\phi}(z_k)$  une fonction vectorielle définie par  $\boldsymbol{\phi}(z_k) = (\phi_1(z_k), \phi_2(z_k), \dots, \phi_K(z_k))$ . De plus, pour alléger les notations, l'indice  $n$  sera omis dans cette section. Ce modèle a la même forme que les modèles utilisés pour les SVM [8] ou les RVM [9]. Obtenir un modèle épars à partir de l'équation 3 est réalisé en fixant la plupart des paramètres de  $\mathbf{w}$  à zéro. Notons  $\Phi$ , la matrice de taille  $K \times K$  associée au bloc  $n$  et construisons la telle que l'élément de la ligne  $i$  et de la colonne  $j$  soit  $\Phi_{i,j} = \phi_i(z_j)$ .  $\Phi$  est une matrice carrée, symétrique, pour laquelle un estimateur de la vraisemblance associé à l'élément  $z_k$  de l'ensemble d'apprentissage est donné par :

$$\widehat{P}(z_k|\omega_1) = K^{-1} \sum_{k'=1}^K \Phi_{k,k'} \quad (4)$$

Un vecteur de vraisemblance  $\boldsymbol{\varphi}$  en relation avec l'ensemble d'apprentissage est construit à partir des estimateurs :

$$\boldsymbol{\varphi} = \left( \widehat{P}(z_1|\omega_1), \dots, \widehat{P}(z_K|\omega_1) \right)^T \quad (5)$$

Le modèle épars est construit via une méthode dans laquelle on sélectionne de manière récursive des vecteurs pertinents de l'ensemble d'apprentissage  $\mathbf{Z}$ . Cette dernière se décompose comme illustrée dans l'algorithme 1. La notation  $\mathbf{1}_K$  représente un vecteur de longueur  $K$  contenant uniquement des valeurs 1. La sortie de l'algorithme est un ensemble de vecteur pertinents et leur poids associé. Le paramètre  $t_v$  représente la précision de l'approximation de la vraisemblance. Pour une approximation grossière  $t_v$  doit être grand. Dans ce cas le nombre de vecteurs utilisés diminue. Des illustrations de l'effet de ce paramètre sont données en section 3.1.

Finalement, l'estimateur peut s'écrire de la façon suivante :

$$\widehat{P}(z_k|\omega_1) \approx \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\phi}}(z_k) \quad (6)$$

avec

$$\tilde{\boldsymbol{\phi}}(z_k) = (\phi_{v(1)}(z_k), \phi_{v(2)}(z_k), \dots, \phi_{v(M)}(z_k))^T \quad (7)$$

Cette approche permet d'approximer la vraisemblance à partir de quelques vecteurs seulement. Initialement, l'ensemble résultant de l'apprentissage de la méthode non paramétrique  $\mathbf{Z}$  contient  $K$  éléments alors que l'ensemble de la méthode éparse  $\tilde{\mathbf{Z}} = (v(1), v(2), \dots, v(M))$ , contient  $M$  éléments avec  $M \ll K$ .

---

**Algorithme 1** Algorithme d'approximation d'un estimateur non paramétrique

---

**ENTRÉES:** matrice de design  $\Phi$ , critère d'arrêt  $t_v$

Calcul du vecteur de vraisemblance initial :

$$\boldsymbol{\varphi}_1 = K^{-1} \Phi \mathbf{1}_K$$

Initialisation :  $m = 0$

**répéter**

$m = m + 1$

Extraction du point de vraisemblance maximale :

$$v(m) = \arg \max_i \varphi_{m,i}$$

Calcul du poids associé :  $\tilde{w}_m = \frac{\varphi_{m,v(m)}}{\Phi_{v(m),v(m)}}$

Mise à jour du vecteur de vraisemblance :

$$\boldsymbol{\varphi}_{m+1} = \boldsymbol{\varphi}_m - \tilde{w}_m \boldsymbol{\phi}(z_{v(m)})$$

**jusqu'à** critère d'arrêt atteint :  $\varphi_{m,v(m)} / \varphi_{1,v(1)} < t_v$

$M = m$

**retourne** Le vecteur des poids :  $\tilde{\mathbf{w}}^T = (\tilde{w}_1, \dots, \tilde{w}_M)$  et

les points sélectionnés :  $\tilde{\mathbf{Z}} = (z_{v(1)}, z_{v(2)}, \dots, z_{v(M)})$

---

## 2.3 Étape de classification à la volée

La classification à la volée est basée sur l'équation (1) qui peut être réécrite à l'aide du modèle épars :

$$\widehat{\omega} = \omega_2 \text{ si } \tilde{\mathbf{w}}^T \tilde{\boldsymbol{\phi}}(z) < C \quad (8)$$

Pour un bloc donné, un ensemble d'orientations du flot optique est calculé. L'orientation dominante ( $z$ ) est obtenue par un maximum de vraisemblance. La vraisemblance de chaque point candidat est calculée par un estimateur non paramétrique dont la forme identique à celui présenté dans l'équation 3, en utilisant l'ensemble d'orientations du flot optique du bloc.

## 3 Applications et résultats

Cette partie présente les évaluations réalisées afin de valider la méthode proposée. La première partie montre la faculté du modèle épars d'approximer une distribution de l'orientation du flot optique. La seconde partie analyse les performances de l'algorithme en condition réelle sur deux scènes routières.

### 3.1 Approximation de la représentation non paramétrique

L'approximation éparse du modèle non paramétrique  $\tilde{\mathbf{Z}}^{(n)}$  présentée plus haut a été comparée au modèle complet  $\mathbf{Z}^{(n)}$ . Pour faire cette comparaison, la fonction de vraisemblance calculée à l'aide (2) est représentée en jaune délimitée par la courbe en pointillé noir sur la figure 2. La fonction de vraisemblance approximée (equation 6) est repré-

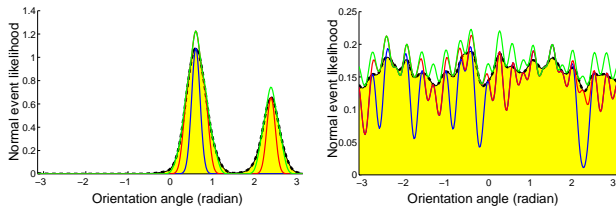


Figure 2 – Approximation de la densité non paramétrique basée sur les fonctions noyaux par un modèle épars de fonctions noyaux sur deux types différents de distribution : La zone jaune délimitée par la courbe en pointillé noir montre la représentation non paramétrique. La courbe bleue ( $t_v = 0.7$ ), la rouge ( $t_v = 0.4$ ) et la verte ( $t_v = 0.08$ ) montrent l'estimation du modèle épars pour différente valeur de  $t_v$ .

sentée pour différentes valeurs de  $t_v$  par les courbes bleu, rouge et verte sur la figure 2.

Les courbes du modèle épars sur la figure 2 montrent que la représentation réduite  $\tilde{Z}^{(n)}$  correspond au modèle complet. La largeur de bande  $\sigma_l$  a été fixée à  $\sigma_l = 0.1$ . Le paramètre  $t_v$  détermine la précision de l'approximation de la vraisemblance. La distribution à deux modes de la figure 2 montre que pour des grandes valeurs de  $t_v$  (courbes bleues et rouges), les largeurs de bande des modes de l'approximation sont tronquées comparées à la densité non paramétrique. Des modes plus petits peuvent même être oubliés dans certains cas (courbe bleue). Les nombres de vecteurs conservés, c'est-à-dire le nombre de vecteurs pertinents nécessaires à l'approximation sont respectivement 1, 4 et 8 pour la courbe bleue, rouge et verte. Pour une distribution plus uniforme, de nombreux vecteurs sont nécessaires pour une approximation correcte. En conséquence, le paramètre  $t_v$  doit être suffisamment petit pour approximer correctement toutes les orientations possibles, ce qui n'est pas le cas de la courbe bleue qui sous-estime certaines orientations. Ici, 21, 26 et 39 gaussiennes ont été nécessaires respectivement pour les courbes bleues, rouges et vertes.

### 3.2 Classification

Les performances du système ont été évaluées sur deux scènes différentes : un carrefour routier et un péage. Les séquences pour chacune des scènes sont issues de webcams réelles dans des conditions climatiques et d'illumination très différentes (e.g. fig.4). Chaque vidéo de la base de données est au format de 320x240 en 12 images par seconde. Les vidéos de la base de données ont été prises toutes les 20 minutes et durent 30 secondes chacune. L'apprentissage a été réalisé sur cent vidéos. Les blocs ont été fixés à une taille de 10x10 pour un total de 768 blocs par image. Les scènes routières ont été préférées aux scènes de foules de personnes, pour des raisons de commodité d'acquisition de données et de variété d'évènements rencontrés. Cependant cet algorithme peut aussi servir à la surveillance de foule dans des couloirs par exemple, comme illustré sur la figure

3, où deux personnes sont détectées à contre sens dans une zone où le déplacement normal s'effectue dans le sens trigonométrique.



Figure 3 – Exemple de surveillance de foule. Le sens de circulation attendu est représenté par les flèches vertes.

Pour l'estimation du mouvement, différentes techniques de flot optique ont été évaluées, de celle de Lucas et Kanade et ses variantes à la méthode de "Block matching" [15]. De toutes ces techniques, celle de Black et Anandan [16] a été conservée pour sa robustesse et la propreté de ses résultats comparée aux autres méthodes, mais aussi pour son temps de calcul relativement rapide.



Figure 4 – Exemples des conditions climatiques (soleil, pluie) et d'illumination (reflet, nuit) des séquences étudiées.

Les résultats de la classification du système ont été évalués en conditions réelles. L'analyse a été réalisée sur 162 vidéos pour le carrefour routier et 256 vidéos pour le péage. Les anomalies recherchées vont du piéton en dehors des passages piétons, aux scooters quittant le trottoir en passant par des marches arrières ou des voitures déboitant juste avant le péage. La classification a été réalisée sur un Pentium 4, 3.4GHz sur lequel, le calcul du flot optique prend 0.11s par image. Le temps de calcul pour l'analyse d'un

bloc dépend du nombre de vecteurs de déplacement présents dans ce bloc. L'analyse sur les blocs sans mouvement n'étant pas effectuée, celle-ci prend environ 5ms, soit 0.2s en moyenne par image. Pour les séquences étudiées jusqu'à présent, ces performances sont proches du temps réel bien que le code ne soit pas optimisé. De plus, la classification réalisée s'adapte bien à la parallélisation sur carte graphique ou sur architecture multi-cœur, en vue d'augmenter encore la fréquence de traitement.

Une anomalie est considérée comme détectée si le système met en exergue au moins un bloc au niveau de l'objet coupable durant le temps de la manoeuvre (e.g. fig.5(b)). Les fausses alarmes ont été comptées de la même manière. Lorsqu'au moins un bloc apparaît pour un mouvement sans réelle anomalie sur la durée du mouvement (e.g. fig.6(b)). Ce compte manuel par évènement a été préféré au compte par image (le classifieur donnant une alerte par image) car c'est une mesure des performances du système proche de ce qui est attendu par l'utilisateur. En effet, le système a pour but d'alerter un agent de sécurité dès que quelque chose est suspect.

Les paramètres du système ont été fixés de manière empirique. Pour  $C = 0.3$ , 17 alarmes ont été relevées sur l'ensemble des séquences du péage dont la durée totale est de 7680 secondes, soit une alarme toutes les 451 secondes en moyenne. Sur ces 17 alarmes, 16 étaient de véritables évènements inattendus alors que la dernière était une fausse alarme. Le carrefour est bien plus complexe à surveiller car les sens de circulation sont plus variés et les véhicules sont plus petits. Cependant, le système alerte 7 fois sur les 5160 secondes de la durée totale, soit une alerte toutes les 737 secondes. Sur ces 7 détections seulement 3 étaient de véritables évènements inattendus. Plus d'évènements auraient pu être détectés pour chacune des scènes mais avec une augmentation sensible du nombre de fausses alarmes comme montré dans le tableau 1. Les figures 5(b)-5(d) illustrent certains évènements détectés, la direction principale pour chaque bloc fautif ainsi que les conditions de détection pour chaque vidéo (en particulier les conditions d'éclairage). La figure 5(a) montre une détection ratée en l'encadrant par un rectangle vert. Il convient également de noter que la détection d'un évènement inattendu est parfois subjective et dépend de l'expert qui fournit les vérités terrain.

$C$	Séquences Péage		Séquences carrefour	
	DC	FA	DC	FA
0.3	16	1	3	4
0.43	28	10	4	38
0.58	38	113	13	164

Tableau 1 – Résultats de la détection pour différentes valeurs de  $C$ . DC signifie Détection Correcte et FA Fausse Alarme.

La plupart des fausses détections sont essentiellement dues

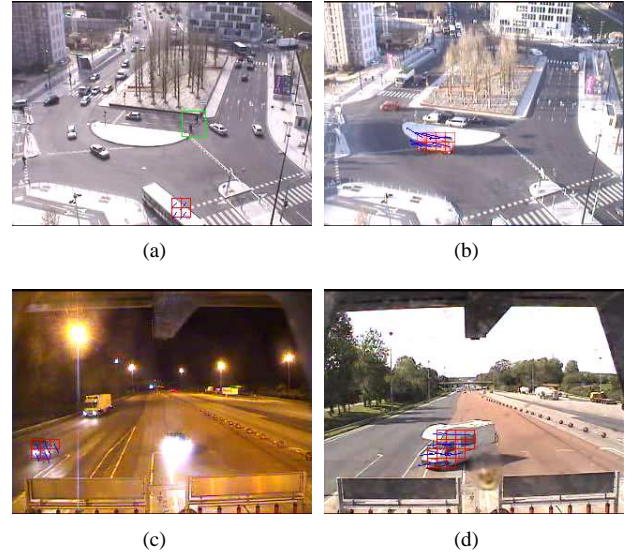


Figure 5 – Exemples d'évènements inattendus. Les carrés rouges représentent les blocs inattendus détectés en accord avec la direction principale de ce bloc (flèche bleue) à l'instant  $t$ . Le carré vert met en évidence les évènements inattendus ratés. (a) Piétons traversant en dehors des passages piétons et exemple de fausse alarme sur le bus en bas de l'image, (b) Camion de pompier allant à contre sens pour éviter des voies de circulation encombrées, (c) Voiture changeant de direction à la dernière minute sur la gauche de l'image, (d) Camping car traversant plusieurs voies pour atteindre la file poids lourd.

à deux raisons : la première est un éventuel mouvement possible qui n'a pas été appris (ce mouvement n'a pas été suffisamment vu durant l'apprentissage). Une illustration de ce phénomène est montré sur la figure 6(a) où les orientations apprises,  $z_{v(i)}$ , ont été représentées par des vecteurs vert dont la norme est pondérée par le poids associé cette orientation  $\tilde{w}_M$ . On peut voir que l'apprentissage de ce bloc n'a pas retenu l'orientation vers la gauche (vue trop rarement pendant l'apprentissage), ce qui entraîne des fausses alarmes lorsqu'un véhicule provient de la droite. La seconde raison intervient lorsque le bas niveau (le flot optique) fournit des primitives bruitées. Ici, le problème d'ouverture du flot optique est la principale cause des primitives bruitées. Ce problème, intrinsèque à la manière dont le flot optique est calculé, apparaît souvent sur les ombres ou les faisceaux lumineux des phares retournant des directions de déplacement complètement erronées (e.g. fig.6(b)). Pour atténuer ces effets non désirés, un calcul du flot optique est effectué uniquement sur des points possédants un score de Harris [17] important.

## 4 Perspectives et Conclusion

Nous avons proposé un système générique pour la détection d'évènements inattendus par vision. Ce dernier apprend les orientations du mouvement sur une scène et vé-



Figure 6 – Exemples de fausses alarmes. (a) Oubli lors de l'apprentissage. Les trois vecteurs du modèle épars ont été représentés en vert. Pour chacun des vecteurs la direction est donnée par l'orientation apprises alors que la norme caractérise le poids associé. (b) Problème d'ouverture. Le flot optique est orthogonal à la bordure du faisceau de lumière, donnant des résultats complètement faux.

rifie ces orientations pendant une phase d'analyse temps réel. Nous proposons une représentation éparse de la distribution des orientations locales du flot optique par une combinaison linéaire de fonctions noyaux. L'algorithme a été testé sur des séquences réelles et nous avons montré qu'il est capable de classifier des événements inattendus tel que des mouvements à contre sens, des véhicules déboitant subitement, des marches arrière, etc.

Les travaux futurs visent à diminuer le nombre de mauvaises détections, souvent dues à la qualité de l'extraction du flot optique. Différentes solutions sont à l'étude : remplacer le flot optique par des approches types détection/mise en correspondance, ajouter des filtrages temporels afin d'obtenir des alertes par événement au lieu d'alerte par image, étudier l'influence du choix de la fonction noyau pour les modélisations de distribution et enfin calculer un seuil de détection en fonction de la distribution apprise. De plus, le système détecte seulement les événements liés au mouvement. Une extension à la détection d'anomalies liées à l'arrêt ou au faible mouvement (personne au sol, stationnement suspect) est aussi à l'étude. D'autre part, une étude comparative de l'approche proposée avec les one class SVM est envisagée.

## Références

- [1] Tao Zhao et Ram Nevatia. Tracking multiple humans in complex situations. *IEEE Pattern Analysis And Machine Intelligence*, 26 :1208–1221, 09 2004.
- [2] Gabriel j. Brostow et Roberto Cipolla. Unsupervised bayesian detection of independant motion in crowds. Dans *Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [3] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, et Steve Maybank. A system for learning statistical motion patterns. *IEEE Pattern Analysis And Machine Intelligence*, 28 :1450–1464, Septembre 2006.
- [4] Sergio A. Velastin, Boghos A. Boghossian, et Maria Alicia Vicencio-Silva. A motion-based image processing system for detecting potentially dangerous situations in underground railway stations. *Transportation Research Part C : Emerging Technologies*, 14, 05 2006.
- [5] Imran N. Junejo et Hassan Foroosh. Trajectory rectification and path modeling for video surveillance. Dans *International Conference on Computer Vision (ICCV)*, 2007.
- [6] S. Blunsden, E. Andrade, et R. Fisher. Non parametric classification of human interaction. Dans *Pattern Recognition and Image Analysis (PRIA)*, 2007.
- [7] E. L. Andrade, S. Blunsden, et R. B. Fisher. Hidden markov models for optical flow analysis in crowds. Dans *International Conference on Pattern Recognition (ICPR)*, volume 1, pages 460–463, 2006.
- [8] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2 :121–167, 1998.
- [9] Michael E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* 1, 2001.
- [10] Yoav Freund et Robert E. Schapire. Experiments with a new boosting algorithm. Dans *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 148–156, 1996.
- [11] Chris Stauffer et W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Pattern Analysis And Machine Intelligence*, 22 :747–757, 08 2000.
- [12] Pekka Paalanen, Joni-Kristian Kamarainen, Jarmo Ilonen, et Heikki Kälviäinen. Feature representation and discrimination based on gaussian mixture model probability densities-practices and algorithms. *Pattern Recognition*, 39 :1346–1358, 2006.
- [13] Bohyung Han, Dorin Comaniciu, Ying Zhu, et Larry S. Davis. Sequential kernel density approximation and its application to real-time visual tracking. *Pattern Analysis and Machine Intelligence*, 30 :1186–1197, 2008.
- [14] R.O. Duda, P.E. Hart, et D.G. Stork. *Pattern Classification*. John Wiley & Sons Inc., 2001.
- [15] J.L. Barron, D.J. Fleet, S.S. Beauchemin, et T.A. Burkitt. Performance of optical flow techniques. *Computer Vision and Pattern Recognition (CVPR)*, 92 :236–242, 1992.
- [16] Michael J. Black et P. Anandan. The robust estimation of multiple motions : parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding (CVIU)*, 63(1) :75–104, 1996.
- [17] C. Harris et M. Stephens. A combined corner and edge detector. Dans *Alvey Vision Conference*, pages 147–151, 1988.