Détection d'objets stationnaires par une paire de caméras PTZ

Constant Guillot

N° d'ordre : D.U. 2219
EDSPIC : 552

Université Blaise Pascal - Clermont-Ferrand II

École Doctorale
Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée par :
Constant Guillot

en vue de l'obtention du grade de

Docteur d'Université

spécialité : Vision pour la robotique

Détection d'objets stationnaires par une paire de caméras PTZ

Soutenue publiquement le 23 janvier 2012 devant le jury :

M. José Luis Lazaro	Président
M. Serge Miguet	Rapporteur
M. Jean-Marc Odobez	Rapporteur
M. Quoc-Cuong Pham	Encadrant
M. Patrick Sayd	Encadrant
M. Christophe Tilmant	Encadrant
M. Jean-Marc Lavest	Directeur de thèse

Remerciements

Je souhaite tout d'abord remercier José Luis Lazaro d'avoir présidé mon jury de thèse, ainsi que Serge Miguet et Jean-Marc Odobez pour avoir pris le temps d'évaluer mes travaux de recherche.

Je souhaite également remercier mon directeur de thèse Jean-Marc Lavest, ainsi que Christophe Tilmant, Patrick Sayd et Quoc-Quoc Pham pour m'avoir proposé ce sujet et accompagné pendant ces trois années de thèse.

Je remercie également mes collègues, avec bien évidemment une pensée particulière à tous les locataires du bureau 32b qui ont contribué à établir une ambiance de travail agréable, mais aussi à mes compagnons de voyage, de pause, et à l'équipe de foot du laboratoire.

Dernier point, mais non le moindre, je remercie mes amis et ma famille pour m'avoir soutenu et encouragé.

Résumé

L'analyse vidéo pour la vidéo-surveillance nécessite d'avoir une bonne résolution pour pouvoir analyser les flux vidéo avec un maximum de robustesse. Dans le contexte de la détection d'objets stationnaires dans les grandes zones, telles que les parkings, le compromis entre la largeur du champ d'observation et la bonne résolution est difficile avec un nombre limité de caméras.

Nous allons utiliser une paire de caméras à focale variable de type Pan-Tilt-Zoom (PTZ). Les caméras parcourent un ensemble de positions (pan, tilt, zoom) prédéfinies afin de couvrir l'ensemble de la scène à une résolution adaptée. Chacune de ces positions peut être vue comme une caméra stationnaire à très faible taux de rafraîchissement.

Dans un premier temps notre approche considère les positions des PTZ comme des caméras indépendantes. Une soustraction de fond robuste aux changements de luminosité reposant sur une grille de descripteurs SURF est effectuée pour séparer le fond du premier plan. La détection des objets stationnaires est effectuée par ré-identification des descripteurs à un modèle du premier plan.

Dans un deuxième temps afin de filtrer certaines fausses alarmes et pouvoir localiser les objets en 3D une phase de mise en correspondance des silhouettes entre les deux caméras et effectuée. Les silhouettes des objets stationnaires sont placées dans un repère commun aux deux caméras en coordonnées rectifiées. Afin de pouvoir gérer les erreurs de segmentation, des groupes de silhouettes s'expliquant mutuellement et provenant des deux caméras sont alors formés. Chacun de ces groupes (le plus souvent constitué d'une silhouette de chaque caméra, mais parfois plus) correspond à un objet stationnaire. La triangulation des points frontière haut et bas permet ensuite d'accéder à sa localisation 3D et à sa taille.

Mots-clef : caméra PTZ, objet stationnaire, soustraction de fond, stéréovision

Abstract

Video analysis for video surveillance needs a good resolution in order to analyse video streams with a maximum of robustness. In the context of stationary object detection in wide areas a good compromise between a limited number of cameras and a high coverage of the area is hard to achieve.

Here we use a pair of Pan-Tilt-Zoom (PTZ) cameras whose parameter (pan, tilt and zoom) can change. The cameras go through a predefined set of parameters chosen such that the entire scene is covered at an adapted resolution. For each triplet of parameters a camera can be assimilated to a stationary camera with a very low frame-rate and is referred to as a view.

First each view is considered independently. A background subtraction algorithm, robust to changes in illumination and based on a grid of SURF descriptors, is proposed in order to separate background from foreground. Then the detection and segmentation of stationary objects is done by re-identifying foreground descriptor to a foreground model.

Then in order to filter out false alarms and to localise the objects in the 3D world, the detected stationary silhouettes are matched between the two cameras. To remain robust to segmentation errors, instead of matched a silhouette to another, groups of silhouettes from the two cameras and mutually explaining each other are matched. Each of the groups then correspond to a stationary object. Finally the triangulation of the top and bottom points of the silhouettes gives an estimation of the position and size of the object.

Key-words : PTZ camera, stationary object, background subtraction, stereo-vision

Le projet Subito

Ces travaux de thèses ont constitué l'essentiel de la contribution scientifique du Laboratoire Vision et Ingénierie des Contenus du CEA List au projet européen ICT-Security SUBITO (www.subito-project.eu). L'objectif de SUBITO est de développer un système automatisé de détection temps réel de bagage abandonné. Cette problématique est critique pour la protection des lieux publics contre les attaques terroristes. Toutefois, la quasi-totalité des bagages abandonnés dans les gares et aéroports ne présentent aucun danger et sont le fait de voyageurs distraits. La gestion des ces bagages suspect est très couteuse pour les opérateurs de transports car le principe de précaution impose des procédures très lourdes (évacuation des lieux, blocage du trafic.). Le déploiement d'un tel système permettrait de détecter rapidement le bagage abandonné pour espérer retrouver et contacter son propriétaire peut-être encore à proximité. Le projet Subito avait pour ambition de développer un système complet incluant la détection du bagage abandonné, l'identification rapide du propriétaire et si possible sa localisation courante. La notion de bagage abandonné nécessite de détecter les objets mobiles devenus statiques, d'associer un propriétaire à ces objets (proximité spatiale et simultanéité d'apparition) et le suivre pour pouvoir, s'il s'éloigne, considérer le bagage comme abandonné et pouvoir interagir avec le propriétaire.

Le consortium du projet SUBITO est constitué de fournisseurs de technologies, d'intégrateur et d'utilisateurs finaux venus de six pays européens. Le rôle du CEA List consistait a étudié l'opportunité d'utiliser des caméras Pan-Tilt-Zoom pour la phase particulière de la détection d'objets stationnaires. L'objectif était de montrer que ces caméras permettaient de couvrir de larges zones pour un nombre réduit de capteur et si possible en augmentant la robustesse du système. Les résultats expérimentaux présentés dans ce mémoire ont alimenté les démonstrations techniques du CEA List dans le projet SUBITO.

Table des matières

Table des figures

Liste des tableaux

Introduction

La dernière décennie a vue une importante généralisation de la vidéo-surveillance, en particulier dans l'espace public. Avec la multiplication des caméras se pose la question de leur efficacité. Il est en effet très difficile pour un opérateur humain de gérer un nombre conséquent de caméras (figure 1). De plus, même avec un nombre très restreint de caméras, les opérateurs ne peuvent maintenir en continu leur attention et manquent donc un nombre conséquent d'événements d'intérêt. Pour ces raisons il est très important d'automatiser autant que possible les tâches de vidéo-surveillance.

FIGURE 1 – Centre de vidéo-surveillance avec son mur d'écrans. L'objectif de la recherche en vidéo-surveillance est de simplifier la tâche des opérateurs tout en augmentant leur efficacité.

Les larges espaces sont particulièrement délicats à surveiller. Avec des caméras classiques, dont le champ de vue et la résolution sont limités, un nombre conséquent de caméras est nécessaire pour pouvoir surveiller toute la zone avec une résolution adaptée. Les caméras PTZ (Pan, Tilt, Zoom) disposent d'un zoom puissant et de deux axes de rotation, ce qui leur permet de se focaliser sur une partie de la scène. Ce type de caméra a donc pour

avantage sa résolution variable et sa possibilité de traiter des zones même très larges (grâce à la possibilité de rotation). Le principal inconvénient est qu'à chaque instant seulement une partie restreinte de la scène est visible. Celle ci est d'autant plus restreinte que le zoom est élevé.

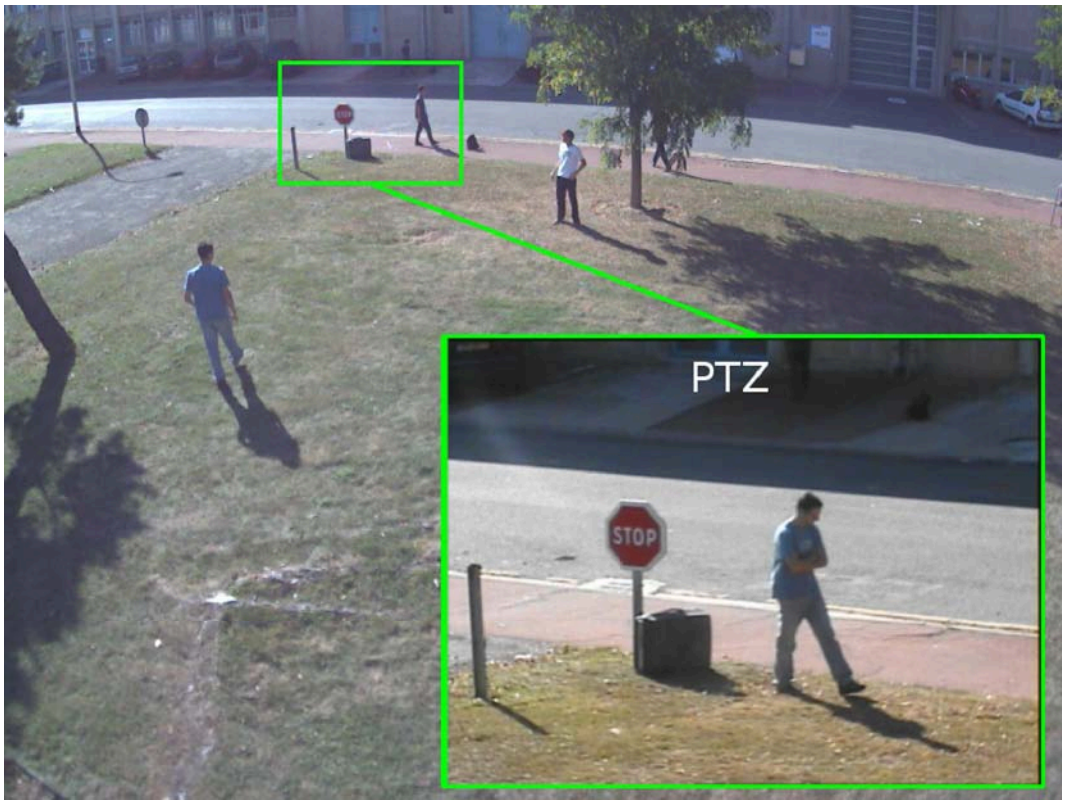


FIGURE 2 – Les caméras PTZ permettent de se focaliser sur une partie de la scène à une résolution adaptée. En contrepartie seulement une partie restreinte de la zone est visible.

Problématique

L'objectif de cette thèse est de proposer un système capable de détecter automatiquement des objets stationnaires dans une large scène. Pour augmenter la robustesse, deux caméras PTZ sont utilisées. Les enjeux sont multiples. La détection des objets stationnaires est un véritable problème en soi et est encore aujourd'hui l'objet de recherche [8]. Les applications sont multiples et concernent principalement la détection de colis abandonnés, puisque dans des zones sensibles ils peuvent constituer des menaces.

Afin d'éviter au maximum d'avoir des fausses alarmes nous allons utiliser une paire de caméras et mettre en correspondance leurs détections. Cela va permettre d'une part de filtrer des erreurs provenant de chaque caméra, mais aussi de calculer la position 3D et la taille de chacun de ces objets.

Contexte

Dans cette thèse nous nous intéressons au problème de la détection et localisation d'objets stationnaires par une paire de caméras PTZ. Dans ce but nous introduisons la notion de tour de garde d'une caméra PTZ. Un tour de garde est un ensemble prédéfini de positions (*pan*, *tilt*, *zoom*) que nous utilisons pour surveiller une grande zone. Chacune de ces positions, que l'on appellera aussi *vue*, est choisie de sorte que l'ensemble de la scène est couvert à une résolution adaptée, tel qu'illustré en figure 3. Chaque caméra parcourt donc en boucle son propre tour de garde, indépendamment de l'autre.

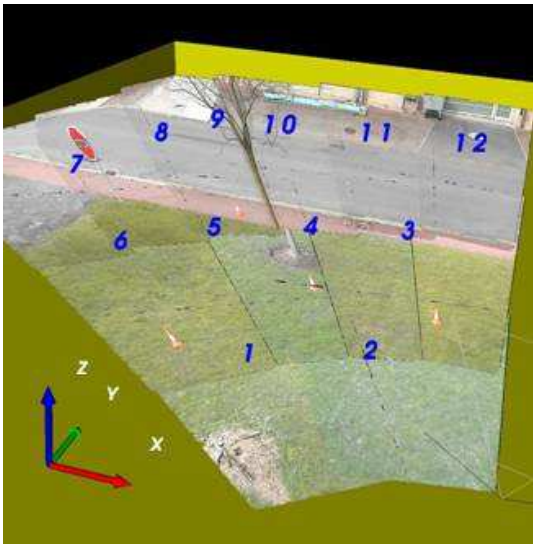


FIGURE 3 – Exemple de tour de garde d'une caméra PTZ. Les différentes *vues* sont numérotées.

On peut en outre remarquer que chaque vue de la caméra peut être considérée comme une caméra stationnaire classique, mais avec un taux de rafraîchissement très faible. Avant qu'une vue ne soit mise à jour il faut en effet que la caméra PTZ parcoure l'ensemble des autres vues du tour de garde. Ceci a pour effet de ne pas garantir de continuité entre deux images suc-

cessives d'une même vue, comme c'est le cas pour une caméra classique, et constituera donc une difficulté supplémentaire à gérer.

Contributions

Nos contributions peuvent être classées en trois catégories, qui correspondent aux trois étapes clefs de notre système de détection d'objets stationnaires. Les deux première étapes sont effectuées de manière indépendante dans chaque vue et l'on se ramène donc au cas d'une caméra fixe avec un faible taux de rafraîchissement.

Nous avons tout d'abord proposé un algorithme de soustraction de fond robuste aux changements de luminosité et applicable au cas particulier d'une caméra PTZ effectuant un tour de garde, qui peuvent être assimilées à des caméras à faible taux de rafraîchissement.

Nous avons ensuite proposé une méthode originale de détection et segmentation des objets stationnaire par une caméra. Cette méthode de détection est robuste aux occultations et la phase de segmentation permet de tirer un maximum d'informations au niveau mono-caméra.

Finalement nous proposons une approche de mise en correspondance entre les deux caméras des silhouettes des objets stationnaires détectés dans chacune d'elles. Cette mise en correspondance est effectuée de manière à rester robuste aux erreurs de segmentation, aux occultations, ainsi qu'à la différence de point de vue des deux caméras.

Organisation

- Dans la première partie de cette thèse, nous nous sommes tout d'abord intéressé à la détection des régions stationnaires. Dans un premier temps nous rappelons les principales méthodes de soustraction de fond, puis nous proposons deux approches. Nous retenons alors l'une de ces approches que nous utilisons ensuite pour élaborer un système de détection d'objets stationnaires.
- La deuxième partie de la thèse concerne la segmentation mono-caméra des objets stationnaires puis leur mise en correspondance dans la paire de caméras. Après un rappel des méthodes de segmentation nous proposons une approche reposant sur les champs de Markov et qui permet de segmenter des objets déposés à des instants différents, tout en restant robuste aux occultations. Les silhouettes des objets stationnaires

ainsi détectées sont alors appariées dans la paire de caméras.
– La dernière partie de la thèse concerne l'application de notre approche au cas des caméras PTZ.

Chapitre 1

Détection d'objets stationnaires

1.1 Introduction

Dans cette première partie nous allons nous intéresser à la détection d'objets stationnaires dans le cas d'une caméra statique. Nous allons proposer une approche reposant sur la modélisation du fond et du premier plan. La soustraction de fond est une étape préliminaire courante dans le domaine de l'analyse vidéo. Elle permet en effet de détecter dans un flux vidéo des objets d'intérêt sans a priori sur leur forme ou leur apparence. Ces objets d'intérêt, qui sont en fait des zones de l'image où un changement significatif a été détecté, permettent de n'avoir à analyser qu'une sous partie "utile" de l'image originale. La soustraction de fond constitue donc une approche assez naturelle pour la détection d'objets stationnaires. Nous allons présenter dans une première partie une méthode de soustraction de fond robuste aux changements de luminosité et particulièrement adaptée à notre contexte d'utilisation. Dans une deuxième partie, les caractéristiques de notre algorithme de soustraction de fond seront utilisées pour définir un critère permettant d'isoler les régions stationnaires parmi celles du premier plan.

1.2 Détection d'objets par soustraction de fond

Dans ce paragraphe nous allons présenter un état de l'art sur les méthodes de détection d'objets reposant sur la modélisation du fond. Nous décrirons ensuite notre approche qui, par l'utilisation du descripteur de texture SURF [7], permet d'obtenir la robustesse aux changements soudains de luminosité nécessaire à notre cadre applicatif. Un dernier paragraphe a pour objectif l'évaluation de notre approche au regard des méthodes de l'état de l'art.

1.2.1 État de l'art sur la soustraction de fond

La soustraction de fond est un domaine très étudié en analyse d'images pour la vidéo-surveillance car elle constitue souvent la première étape dans une chaîne d'analyse plus complexe. Elle peut servir à déterminer des zones de l'image où une activité potentiellement intéressante est en train de se produire et où est susceptible de se trouver un piéton ou un objet. La soustraction de fond permet donc de restreindre l'analyse de l'image à une zone plus réduite.

Les difficultés rencontrées lors de l'étape de soustraction de fond sont multiples. Tout d'abord les images sont bruitées et peuvent aussi contenir des artefacts de compression plus ou moins importants. Ceci fait que, même dans le cas d'une scène parfaitement inchangée, effectuer une simple différence d'images peut donner des fausses alarmes. En plus de ces problèmes liés à la qualité de l'image il y a aussi des difficultés liées à la nature de la scène filmée. Parmi les difficultés principales que l'on rencontre en soustraction de fond on trouve les problèmes suivants, mis en évidence par Toyama *et al.* [97] : variations graduelles/soudaines de la luminosité, ombres portées au sol par les objets présents dans la scène, vibrations de la caméras, multimodalité du fond, saturation, faibles contrastes dans les zones peu lumineuses, déplacement d'objets du fond.

Nous allons maintenant présenter différents algorithmes de soustraction de fond. Même s'il existe des techniques très différentes, généralement les algorithmes s'articulent en trois parties :

- **Initialisation :** les paramètres du modèles sont appris sur une séquence d'initialisation. Cette étape n'est effectuée qu'une seule fois.
- **Détection/Classification :** les pixels sont classés comme objet ou fond en fonction des paramètres du modèle.
- **Mise à jour :** le modèle de fond est mis à jour pour prendre en compte les nouvelles observations. Le système est alors prêt pour une nouvelle détection.

1.2.1.1 Méthodes statistiques pour la soustraction de fond

Mélange de Gaussiennes : Une grande partie de la littérature concernant la soustraction de fond est inspirée de Stauffer et Grimson [95]. Leur approche permet notamment de gérer les variations graduelles de luminosité ainsi que les fonds multi-modaux. Le principe est le suivant : dans l'espace de couleur RVB chaque pixel est représenté par un mélange de k gaussiennes pondérées (en pratique k est souvent fixé entre 3 et 5). Ces gaussiennes vont servir à

représenter les distributions d'un historique récent des modes du fond ainsi que les objets du premier plan. Ainsi, pour un pixel donné à un temps t la probabilité d'observer X_t est :

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{1.1}$$

où chaque distribution gaussienne est représentée par sa moyenne $\mu_{i,t}$, sa matrice de covariance $\Sigma_{i,t}$, et est pondérée par $\omega_{i,t}$. $\omega_{i,t} \in [0, 1]$ est significatif du nombre de fois que le mode i a été observé. η est la densité de probabilité de la loi normale :

$$\eta(X; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1} (X-\mu)} \tag{1.2}$$

Durant la phase d'initialisation les paramètres $\omega_{i,t}$, $\mu_{i,t}$ et $\Sigma_{i,t}$ sont estimés à l'aide d'une approximation en ligne de l'algorithme K-means.

Durant la phase de classification les modes sont classés en modes représentant le fond et modes représentant un objet. Pour cela ils sont ordonnés par $r_{i,t} = \frac{\omega_{i,t}}{\sigma_{i,t}}$ décroissant. En effet pour un mode cette quantité est d'autant plus grande que sa variance est faible et qu'il a souvent été observé. Cela représente l'intuition que l'on se fait d'un pixel du fond, c'est à dire qu'il est souvent observé et à une valeur "stable". Les modes que l'on considère comme représentant le fond sont les premiers modes (pour cette nouvelle relation d'ordre) dont la somme des poids excède un certain seuil T.

Ainsi si les $\omega_{i,t}$ sont réindicés de sorte que $r_{i,t} \geq r_{i+1,t}$, alors les modes représentant le fond sont les B premiers :

$$B = \arg\min_{b} \left(\sum_{k=1}^{b} \omega_{k,t} > T \right) \tag{1.3}$$

où T représente la proportion des observations représentant le fond. Il est donc possible que le fond soit représenté par plusieurs modes. Quant aux modes restants ils représentent des objets du premier plan, dont la présence n'est que temporaire. Si X_t est la nouvelle valeur du pixel, un test d'appartenance est effectué pour lui attribuer un mode et donc le classer comme fond ou objet :

$$\sqrt{(X_t - \mu_{i,t})^T \Sigma_{i,t} (X_t - \mu_{i,t})} < k\sigma_{i,t} \tag{1.4}$$

où k est une constante fixée par les auteurs à $k = 2, 5$.

Si le pixel appartient effectivement à l'un des modes existant, les paramètres de ce mode sont mis à jour de la manière suivante :

$$\omega_{i,t+1} = (1-\alpha)\omega_{i,t} + \alpha \tag{1.5}$$
$$\mu_{i,t+1} = (1-\rho)\mu_{i,t} + \rho X_t \tag{1.6}$$
$$\sigma^2_{i,t+1} = (1-\rho)\sigma^2_{i,t} + \rho(X_t - \mu_{i,t+1})^T(X_t - \mu_{t+1}) \tag{1.7}$$
$$\text{avec } \rho = \alpha\eta(X_t, \mu_{i,t}, \sigma_{i,t}) \tag{1.8}$$

où $\alpha \in [0,1]$ est le coefficient d'apprentissage de l'algorithme, il contrôle la vitesse d'adaptation du modèle aux observations.

Pour les autres modes, seul le poids est mis à jour :

$$\omega_{i,t+1} = (1-\alpha)\omega_{i,t} \tag{1.9}$$

Si le pixel n'a pu être affecté à aucun des modes existants alors le mode de poids le plus faible est supprimé puis remplacé par un nouveau avec des paramètres initialisés arbitrairement :

$$\omega_{i,t+1} = \omega_0 \tag{1.10}$$
$$\mu_{i,t+1} = X_t \tag{1.11}$$
$$\sigma^2_{i,t+1} = \sigma_0 \tag{1.12}$$

avec ω_0 un poids faible et σ_0 une grande variance.

Le principal avantage de cet algorithme est de pouvoir gérer certains fonds dynamiques et de s'adapter aux variations graduelles de luminosité. Le principe a été repris et amélioré, de sorte qu'il existe désormais une multitude de méthodes reposant sur les mélanges de gaussiennes. Bouwmans *et al.* [15] donnent un état de l'art sur ces méthodes, ainsi qu'une classification des stratégies les plus courantes.

Estimation par noyau : Les méthodes par mélange de gaussiennes reposent sur l'hypothèse que les observations suivent un mélange de lois gaussiennes. Dans certains cas cette modélisation est suffisante mais elle ne reflète pas la distribution réelle des données. Elgammal *et al.* [31] ont proposé d'utiliser une méthode d'estimation non paramétrique pour estimer la densité de probabilité des valeurs des pixels du fond. Pour cela un historique de N observations $X_{t-N+1}, \ldots, X_t$ doit être conservé.

Un estimateur de la densité de probabilité des observations est :

$$\hat{P}(X_{t+1}) = \frac{1}{N} \sum_{i=t-N+1}^{t} K\left(X_{t+1} - X_i\right) \tag{1.13}$$

où K est une fonction noyau. Elgammal *et al.* choisissent comme fonction noyau une gaussienne centrée de matrice de covariance Σ.

$$K(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2} x^T \Sigma^{-1} x} \tag{1.14}$$

La discrimination entre fond et premier plan se fait avec un simple seuillage. Si $\hat{P}(X_t) \geq T$ alors c'est un pixel du fond, sinon c'est un pixel du premier plan.

Les auteurs utilisent deux modèles de fond mis à jour différemment.

Un *modèle court terme* prenant en compte les N dernières observations est utilisé. Il utilise les observations les plus récentes pour s'adapter rapidement aux changements de la scène. La mise à jour du modèle n'est donc effectuée que sur les pixels qui ont été classés comme fond. On peut s'attendre à deux types de faux positifs, ceux dûs aux événements rares non représentés dans le modèle, ceux dûs aux erreurs commises lors de la phase d'apprentissage. Ces faux positifs vont être réduits par l'ajout du second modèle de fond.

Un *modèle long terme* prenant en compte N observations sélectionnées aléatoirement dans une fenêtre temporelle beaucoup plus grande. Ce modèle de fond est plus stable que le précédent mais il va générer plus de faux positifs puisqu'il n'est pas le plus récent.

Le masque binaire final du premier plan est le "ET" logique des masques des deux modèles.

L'estimation non paramétrique a notamment était reprise par Mittal *et al.* [76].

Réduction de dimension par ACP : Oliver *et al.* [82] proposent d'effectuer une Analyse en Composantes Principale d'une séquence vidéo pour construire un modèle de fond (communément appelé *eigen background*). On considère qu'une image de taille $n \times m$ est un vecteur d'un espace vectoriel $E_{n \times m}(\mathbb{R})$. A partir d'un ensemble de N images il est possible de calculer l'image moyenne μ_B ainsi que la matrice de covariance associée. Cette matrice de covariance C_B peut être diagonalisée en D dans une base orthonormée de sorte que :

$$D = P^{-1} C_B P \tag{1.15}$$

où P est la matrice de passage associée. Dans cette nouvelle base on peut montrer que les termes diagonaux (valeurs propres λ_i) sont les variances des données selon l'axe défini par le vecteur propre associé et $\frac{\lambda_i}{\sum \lambda_i}$ donne la part d'inertie des données selon cet axe. En ne retenant que les M dimensions correspondant aux plus fortes valeurs propres on définit donc un sous-espace vectoriel E_{BG} de $E_{n\times m}(\mathbb{R})$. Notons P_M la matrice de projection sur E_{BG}. Puisque seules les dimensions correspondant aux plus fortes variances sont retenues E_{BG} devrait n'être représentatif que du fond. Pour trouver les objets en mouvement dans une nouvelle image I_t, il suffit alors de la projeter sur E_{BG} puis d'effectuer la différence avec l'image originale. Les coordonnées de la projection de I_t sur E_{BG} sont :

$$B_t = P_M(I_t - \mu_B) \tag{1.16}$$

La projection de I_t sur E_{BG} est alors :

$$I_{BG,t} = P_M^T B_t + \mu_B \tag{1.17}$$

Enfin les objets du premier plan sont obtenus par un simple seuillage de la différence entre la nouvelle image et sa projection sur l'espace de fond :

$$|I_t - I_{BG,t}| > T \tag{1.18}$$

Cette modélisation du fond a été reprise par Xu *et al.* [109] ainsi que Rymel *et al.* [90]. Han *et al.* [42] ont quant à eux proposé une approche similaire mais permettant de travailler sur des images couleur.

Plusieurs auteurs se sont intéressées à recenser et comparer les différentes techniques de soustraction de fond existantes. On peut particulièrement remarquer les états de l'art de Piccardi [85] et plus récemment celui de Elhabian *et al.* [32].

1.2.1.2 Soustraction de fond pour caméras PTZ

Plusieurs méthodes de soustraction de fond ont été développées dans le cadre spécifique des caméras PTZ. En effet ces caméras n'observent qu'une partie restreinte mais variable de la scène. La plupart de ces méthodes reposent sur la création d'un panorama et les contributions se focalisent sur la gestion en temps réel du modèle de fond de ce celui ci.

Bhat *et al.* [14] proposent de créer un panorama en recalant les images acquises par la caméra PTZ par une transformation affine. Le modèle de

fond consiste en deux images de luminance mise à jour par un filtre autorégressif. A la différence de Stauffer et Grimson [95] le deuxième modèle du fond, décalé dans le temps par rapport au premier permet d'éviter les fausses alarmes dues au mouvement des objets lents.

Cucchiara *et al.* [24] estiment le mouvement propre de la caméra en calculant un flot optique avec l'algorithme de Lucas Kanade [67]. Une translation est alors estimée de manière robuste. Pour cela un histogramme des orientations des flots est calculé. Une gaussienne centrée sur le mode principal est utilisée pour filtrer les orientations différentes de l'orientation principale. Le mouvement propre de la caméra est alors défini comme la translation moyenne induite par les flots non filtrés. Une fois le mouvement propre estimé, l'image courante est recalée sur le panorama. La soustraction du fond est alors réalisée simplement en effectuant une différence d'image. Finalement des contours actifs sont utilisés pour avoir une segmentation fine des objets d'intérêt. Le modèle de fond est mis à jour en moyennant le panorama de référence avec l'image courante, à l'exception des zones détectées comme objet d'intérêt.

Azzari *et al.* [6] proposent d'estimer une homographie pour recaler l'image courante en deux fois. Deux ensembles de points d'intérêt sont extraits de la partie extérieure de l'image courante et l'image précédente et sont mis en correspondance en utilisant le traqueur KLT [67]. Pour plus de précision, ce traqueur est initialisé avec une estimation grossière du déplacement entre les deux images trouvées par une méthode de corrélation de phase. Cette initialisation permet de pouvoir gérer des grands déplacements tout en conservant pour le traqueur une petite fenêtre de recherche pour la mise en correspondance. L'estimation du mouvement propre est ensuite effectuée en séparant les points d'intérêt du fond des objets d'intérêt par l'utilisation d'une simple heuristique sur leur vecteur vitesse. Une fois les points aberrants mis de côté une homographie est calculée. L'image courante est recalée sur le panorama en multipliant cette homographie avec celle obtenue au temps précédent. Cependant cette façon incrémentale de composer les homographies accumule les erreurs inhérentes à chaque nouvelle estimation. A partir de cette première estimation l'image courante est alors recalée directement sur le panorama complet. Ceci fait l'image courante est mélangée au panorama :

$$B_{x,t} = (1 - \alpha)B_{x,t-1} + \alpha I_{x,t} \tag{1.19}$$

où $B_{x,t}$ est la valeur du pixel x du panorama au temps t et $I_{x,t}$ est la valeur de pixel x de l'image courante recalée au temps t. Ce mélange est nécessaire

pour prendre en compte les variations de luminosité au cours du temps. La soustraction du fond [11] est faite simplement en soustrayant une image de fond à l'image courante et considérant une modélisation du bruit.

Robinault *et al.* [88] partent de la constatation que si l'on veut stocker un panorama complet projeté sur un cube sans perte d'information et avec une distance focale de 800 pixels et un modèle de fond constitué d'un mélange de deux gaussiennes, 540Mo de mémoire sont nécessaires. De plus les changements de luminosité qui se produisent dans la scène font que si une zone n'a pas été visitée depuis un certain temps le modèle de fond n'est plus nécessairement encore valide. Ce problème est d'autant plus important en extérieur. Pour ces raisons, ils choisissent de ne conserver que la partie "utile" du modèle de fond, c'est à dire la portion du panorama correspondant à l'image courante et précédente. A un instant donné on peut donc séparer le panorama en trois partie : la partie visible à la fois dans l'image courante et l'image précédente où le modèle de fond est utile, la partie visible seulement dans l'image précédente et dont le modèle de fond est supprimé, la partie visible seulement dans l'image courante et dont le modèle de fond n'est pas encore initialisé. Ceci est intéressant dans le cas d'applications pour le suivi ou la détection d'objets en mouvement. Le recalage est fait en calculant une homographie. Des points de Harris [43] sont détectés dans l'image courante $p_{j,I}$ et l'image de fond $p_{i,P}$. Une mesure de dissimilarité sur deux ensembles de points est définie pour quantifier la qualité du recalage :

$$D_H = \sum_j \min(\|n(p_{j,I}) - Hp_{j,I}\|) \tag{1.20}$$

où H est l'homographie estimée pour le recalage et $n(p_{j,I})$ est le point d'intérêt du panorama le plus proche de $p_{j,I}$.

L'homographie qui minimise D_H est trouvée avec un algorithme du simplex.

Le modèle de fond est quant à lui le mélange de trois gaussiennes tel qu'introduit par Stauffer et Grimson [95].

Ces modélisations du fond développées spécifiquement pour les caméras PTZ sont particulièrement adaptées à des problèmes de suivi d'objet ou de détection d'objet en mouvement. En effet la création d'un panorama permet d'avoir un modèle de fond pour n'importe quelle position de la caméra. Cependant les difficultés dues à la gestion du panorama, telles que le recalage de l'image courante sur le panorama, font qu'il est préférable d'éviter autant que possible l'utilisation de panorama. De plus ces approches ne s'attaquent pas au problème de la validité du modèle de fond dans les zones qui n'ont pas

été visitées depuis un certain temps, ce qui est crucial pour notre application. Ces éléments nous ont conduits à proposer une approche basée sur un tour de garde. Il apparaît donc nécessaire de se concentrer sur le développement d'un modèle de fond suffisamment robuste pour supporter des mises à jour temporellement espacées (intervalle de temps nécessaire pour effectuer un tour complet).

1.2.1.3 Descripteurs robustes aux changements de luminosité

Quelques méthodes ont été développées spécifiquement pour être robustes aux changements de luminosité. Elles exploitent une description de la texture du voisinage des points.

Une des améliorations existantes de l'approche de Stauffer et Grimson, proposée par Chen *et al.* [22] et dont il a été démontré qu'elle faisait partie des méthodes les plus efficaces par Dhome *et al.* [29], repose sur le calcul d'un descripteur de contraste au niveau d'un bloc 8×8 de l'image. Ceci permet de gagner en robustesse par rapport à l'approche pixellique.

Plutôt que de se placer dans l'espace de couleur RVB, Chen *et al.* choisissent de diviser l'image en blocs définis par une grille régulière. Dans chacun de ces blocs un descripteur de taille 48 est calculé. Comme illustré en figure 1.1 les blocs sont divisés en 4 quadrants et un pixel central est défini comme étant la moyenne des 4 pixels situés les plus au centre. Un histogramme de contraste entre les composantes j et k est calculé de la manière suivante. Dans chaque quadrant q_i la moyenne des contrastes positifs $CH_{q_i}^{j,k,+}(p_c)$ et la moyenne des contrastes négatifs $CH_{q_i}^{j,k,-}(p_c)$ sont calculées :

$$CH_{q_i}^{j,k,+}(p_c) = \frac{\sum_{p \in q_i} C^{j,k}(p, p_c), C^{j,k}(p, p_c) > 0}{\#_{q_i}^{+}} \tag{1.21}$$

$$CH_{q_i}^{j,k,-}(p_c) = \frac{\sum_{p \in q_i} C^{j,k}(p, p_c), C^{j,k}(p, p_c) \leq 0}{\#_{q_i}^{-}} \tag{1.22}$$

où $C^{j,k}(p, p_c)$ est la valeur du contraste entre la composante j du pixel p et la composante k du pixel p_c. Parmi les 9 couples de composantes seules les 6 combinaisons $(R,R), (R,V), (R,B), (V,V), (V,B), (B,B)$ sont considérées. Il y a 6 paires avec 4 quadrants ayant 2 statistiques donc le descripteur final est de dimension $6 \times 4 \times 2 = 48$.

Ce descripteur repose sur une mesure du contraste ce qui lui confère une certaine robustesse aux changements de luminosité. De plus l'information

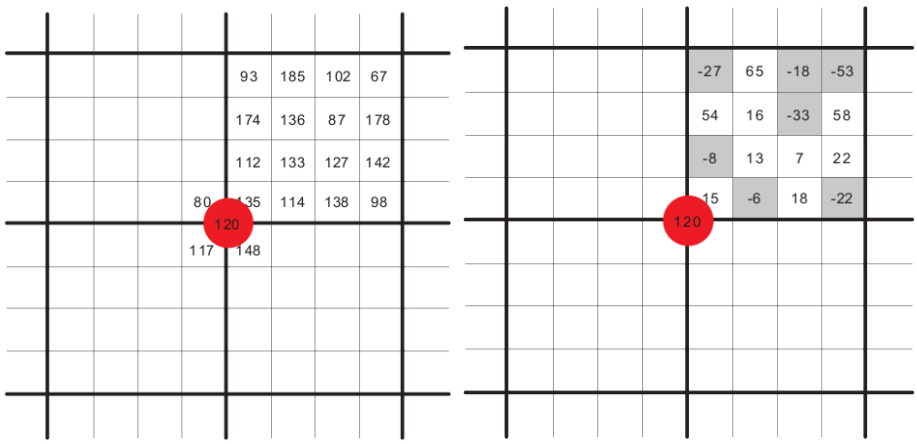


FIGURE 1.1 – Exemple de calcul du descripteur de contraste de Chen *et al.* [22]. Un seul quadrant est montré. Le pixel central (ici en rouge) est la moyenne des quatre pixels les plus au centre. Gauche : valeurs des pixels. Droite : valeurs des contrastes.

n'est plus seulement au niveau pixellique mais elle est agrégée au niveau d'un bloc.

De manière assez similaire à Chen *et al.* , Yao *et al.* [110] utilisent un descripteur de texture dans leur modélisation du fond. Pour chaque pixel la texture de son voisinage est représentée par un *Local Binary Patern* (LBP), déjà utilisé en soustraction de fond par Heikkila *et al.* [46]. Un LBP décrit la texture d'un voisinage circulaire de rayon R d'un pixel x en codant le signe du contraste par rapport au pixel central x de P pixels du cercle. Ainsi on a :

$$LBP_{P,R}(x) = \{LBP^{(p)}_{P,R}(x)\}_{p=1,\dots,P} \tag{1.23}$$

avec

$$LBP^{(p)}_{P,R}(x) = s(I^g(v_p) - I^g(x) + n), \text{ avec } s(x) = \begin{cases} 1 & \text{si } x \geq 0, \\ 0 & \text{si } x < 0, \end{cases} \tag{1.24}$$

où v_p est le $p^{ième}$ pixel du cercle de rayon R centré en x parmi les P pixels régulièrement espacés considérés, $I^g(x)$ est la valeur en niveaux de gris du pixel x. Le paramètre n permet de rendre la signature du LBP plus stable face au bruit dû notamment à la compression. n représente la variation tolérée de la valeur du niveau de gris d'un pixel qui n'affecte pas la signature. Pour palier la faiblesse des LBP sur les régions non texturées de l'image les auteurs proposent de considérer la valeur des pixels dans l'espace RGB. Afin de

conserver au plus une invariance aux changements de luminosité deux points de l'espace RGB sont comparés non pas comme souvent avec une distance euclidienne mais en considérant l'angle qu'ils forment avec l'origine. Finalement c'est une distance hybride qui est utilisée combinant une distance sur les LBP et la similarité dans l'espace RGB qui est utilisée pour comparer deux pixels.

Tian *et al.* [64] utilisent un mélange de gaussiennes classique basé sur la méthode de Stauffer et Grimson [95]. Cependant pour gérer les changements soudains de luminosité, qui génèrent inévitablement des faux positifs, ils proposent d'utiliser en plus une information de texture. Les valeurs du gradient sont connues pour être moins sensibles aux changements de luminosité. Ils proposent alors d'utiliser une mesure de similarité de texture entre l'image courante et l'image de fond, déjà introduite par Li et Leung [61] :

$$S(X) = \frac{\sum_{u \in W_x} 2\|g(u)\|\|g_b(u)\| \cos\theta}{\sum_{u \in W_x} (\|g(u)\|^2 + \|g_b(u)\|^2)} \tag{1.25}$$

où X est le pixel pour lequel est calculé la similarité, W_x est un voisinage de X, g et g_b sont les vecteurs gradients de l'image courante et de l'image de fond et θ est l'angle entre ces deux vecteurs. Cette mesure est utilisée pour filtrer les faux positifs détectés par la méthode de Stauffer et Grimson. Elle est donc calculée pour tous les pixels détectés comme premier plan. Si pour un de ces pixel $S(X) \geq T_s = 0,7$ alors la texture du premier plan est très similaire à celle du fond et le pixel X est en fait re-classé comme pixel du fond, sinon il est classé comme pixel du premier plan.

Noriega *et al.* [79] [80] proposent l'utilisation d'un histogramme local de contour. L'histogramme est constitué de 8 classes correspondant à des orientations du gradient. Afin d'éviter certains effets indésirables de quantification les données sont lissées par un noyau gaussien avant d'être comptabilisées dans chacune des classes. La norme et l'orientation des gradients sont calculées. Chaque classe de l'histogramme est définie comme suit :

$$h_o^A = \sum_{S \in A} G_o\left(o, \mu_o(E_S), \sigma_o(E_S)\right) G_S\left(S, \mu_S, \sigma_S\right) \|E_S\| \tag{1.26}$$

où o est une orientation, A un voisinage du pixel courant S, E_S le gradient au pixel S, $\mu_o(E_S)$ l'orientation de E_S, $\sigma_o(E_S) = a_0\|E_S\| + b_o$ est définie expérimentalement.

Bien qu'elles semblent robustes, ces méthodes ne permettent pas la création d'un modèle du premier plan auquel on pourrait se ré-identifier tel que nous allons le proposer en section 1.2.2.2 et 1.2.2.3.

1.2.2 Notre approche

La figure 1.2 montre une même scène sous trois conditions très différentes de luminosité et met en évidence les difficultés dues aux spécularités, aux ombres portées ou encore à la saturation. Nous avons vu que, dans notre contexte de tour de garde d'une caméra PTZ, nous nous ramenons au cas d'une caméra fixe avec un très faible taux de rafraîchissement. La robustesse aux changements soudains de luminosité devient alors un problème critique. En effet, à cause du très faible taux de rafraîchissement, qui suivant la longueur du tour de garde peut être de l'ordre d'une image pour plusieurs dizaines de secondes, la variation de luminosité n'est pas continue entre deux images consécutives. Ceci est illustré par la figure 1.3. La non continuité qui apparaît entre deux images consécutives laisse penser que les méthodes statistiques ne sont pas particulièrement adaptées à ce type de problème. Nous nous intéresserons donc dans cette section à un descripteur de texture reconnu pour ses qualités de robustesse aux changements de luminosité et que nous allons par la suite intégrer dans un algorithme de soustraction de fond.

Nous avons testé deux approches de soustraction de fond reposant sur l'utilisation du descripteur SURF [7]. Celles-ci sont présentées en section 1.2.2.2 et 1.2.2.3 après la section introduisant la version allégée du descripteur SURF que nous avons utilisée. Pour notre application il n'est en effet pas nécessaire d'avoir d'invariance à l'échelle ni à la rotation.

1.2.2.1 Le descripteur SURF

Le descripteur SURF a été introduit par Bay *et al.* [7]. Il peut être considéré comme une approximation du descripteur SIFT de Lowe [66] dont il a été démontré qu'il était l'un des descripteurs les plus robustes [74]. Ces descripteurs ont été développés de sorte qu'ils puissent être utilisés dans des contextes très variés. Pour cette raison ils sont invariants par rotation et par changement d'échelle. Pour notre application, la soustraction de fond, les images sont toujours acquises à la même échelle et la scène est toujours vue du même angle. Ces deux propriétés d'invariance ne nous sont donc pas nécessaires et ne seront pas abordées.

Le descripteur SURF décrit la répartition des gradients dans un voisinage du point d'intérêt, c'est ce qui lui confère une bonne robustesse aux changements de luminosité. Des ondelettes de Haar du premier ordre sont calculées dans les directions x et y.

La première étape du calcul du descripteur consiste à considérer une fenêtre carrée centrée sur le point d'intérêt et d'une largeur de 24 pixels. Cette

FIGURE 1.2 – Trois exemples d'une même scène sous des conditions très différentes de luminosité. On peut noter la présence de spécularité sur la route due à la pluie, de fortes ombres portées au sol, une importante saturation en cas de luminosité trop forte et une importante variation de la teinte.

FIGURE 1.3 – Exemple de changement brusque de luminosité entre deux images consécutives à un faible taux de rafraîchissement.

fenêtre est divisée en 4×4 sous-régions dans lesquelles sont calculés les gradients horizontaux et verticaux d_x et d_y comme illustré en figure 1.4.

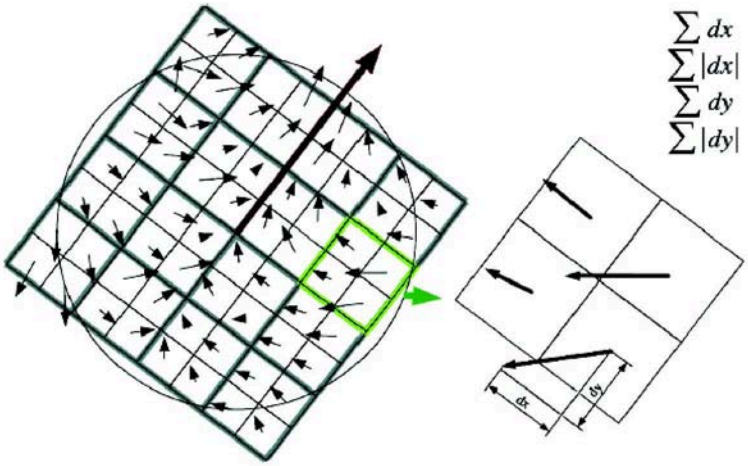


FIGURE 1.4 – Calcul du descripteur SURF.

Pour donner plus d'importance aux gradients proches du point d'intérêt ceux ci sont pondérés par une gaussienne centrée sur le point d'intérêt et d'écart type $\sigma = 3,3$. Dans chaque sous-région i, quatre grandeurs sont calculées pour former un vecteur caractéristique :

$$v_i = \begin{pmatrix} \sum d_x \\ \sum |d_x| \\ \sum d_y \\ \sum |d_y| \end{pmatrix} \tag{1.27}$$

Le vecteur caractéristique final v est la concaténation des vecteurs caractéristiques v_i de seize sous-régions et est donc de dimension $16 \times 4 = 64$. Finalement pour plus d'invariance aux changements de contraste le vecteur descripteur v est normalisé.

La figure 1.5 montre par trois exemples comment le voisinage d'un point d'intérêt influe sur les composantes du descripteur v_i d'une sous-région.

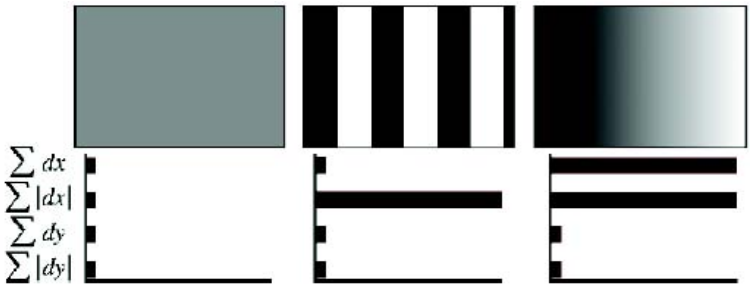


FIGURE 1.5 – Influence des gradients d'une sous région sur le vecteur descripteur associé.

1.2.2.2 Soustraction de fond par densité de points d'intérêt

Notre première approche repose sur l'extraction de points de Harris [43] auxquels nous allons associer un descripteur SURF [7]. L'idée est de détecter des points d'intérêt dans l'image courante et de les comparer avec ceux du modèle de fond. Le modèle de fond doit donc être une image de luminance.

Les points d'intérêt sont détectés avec un seuil faible sur le score de Harris afin d'avoir des points sur toutes les zones texturées de l'image. Cependant l'extraction de points d'intérêt, surtout avec un seuil faible, n'est pas un processus stable (comme illustré en figure 1.6). Il est possible qu'un point d'intérêt détecté dans l'image de fond ou l'image courante ne soit pas détecté dans l'autre image. Pour cette raison une fois que les points d'intérêt sont détectés dans chacune des images, nous construisons la liste de l'union des points d'intérêt détectés dans les deux images. Ainsi même si un point d'intérêt n'est détecté que dans l'une des images il sera tout de même pris en compte dans les deux et est donc garanti d'avoir un candidat potentiel à l'appariement.

Les points n'ayant pu être appariés mettent donc en évidence un changement de texture et donc la présence d'un objet d'intérêt. Nous utilisons pour cela une méthode d'estimation non paramétrique (appelée aussi méthode à noyaux) pour estimer la fonction densité de probabilité de présence

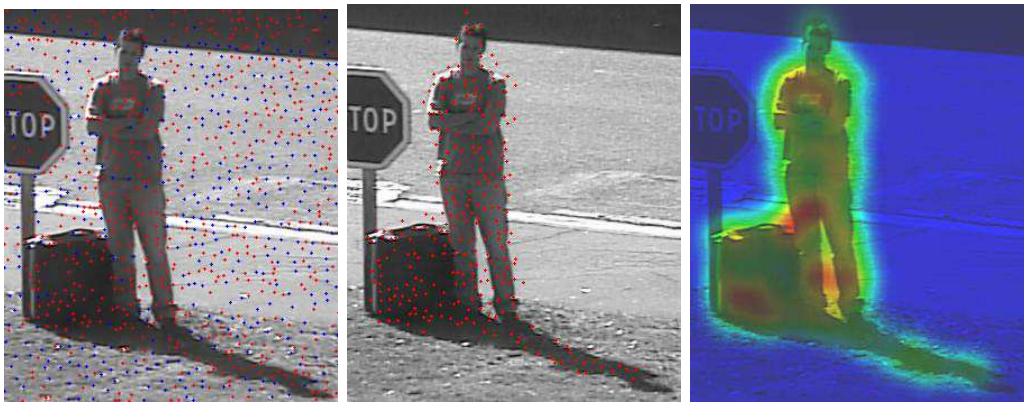


FIGURE 1.6 – Gauche : points d'intérêt détectés dans l'image de fond et l'image courante. En bleu ceux détectés dans les deux images, en rouge les autres. Milieu : Points d'intérêt non appariés. Droite : Estimation de la densité de probabilité de présence de points non appariés.

des points d'intérêt non appariés sur toute l'image et ainsi obtenir une information dense à partir d'une information parcimonieuse. On considère ainsi la position des points d'intérêt non appariés comme l'observation d'une variable aléatoire de fonction densité de probabilité d que nous allons estimer.

Soient $(p_1, \ldots, p_N)$ les N points d'intérêt non appariés, alors

$$\hat{d}_h(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{\|x - p_i\|}{h}\right) \tag{1.28}$$

avec K la fonction noyau, est une estimation de la valeur de d au pixel x. h est un paramètre de lissage qui spécifie l'importance de l'influence des observations sur leur voisinage. Nous avons choisi un noyau gaussien 2D :

$$K(x) = \frac{1}{2\pi} e^{-\frac{1}{2}x^2} \tag{1.29}$$

Plutôt que de travailler directement avec $\hat{d}_h$ nous considérons la quantité $N\hat{d}_h$ qui est invariante au nombre de points d'intérêt détectés dans l'image. De cette façon un seul seuil peut être utilisé sur des séquences de résolutions différentes. Finalement les pixels du premier plan sont ceux vérifiant $N\hat{d}_h > s$, avec s notre seuil de détection, et les autres sont des pixels du fond.

Afin de prendre en compte les variations de la scène au cours du temps, notre modèle de fond doit être mis à jour. Le principe est simplement de remplacer les pixels de l'image de fond par ceux de l'image courante. Nous appliquons donc la règle de mise à jour suivante.

Entrées: image de fond : bg, nouvelle image : img
calculer l_1 points d'intérêt de bg
calculer l_2 points d'intérêt de img
calculer d_1 descripteurs de l_1 sur bg
calculer d_2 descripteurs de l_2 sur img
calculer d'_1 descripteurs de l_1 sur img
calculer d'_2 descripteurs de l_2 sur bg
apparier les points $l_1 \cup l_2$ de bg décrits par $d_1 \cup d'_2$
aux points $l_1 \cup l_2$ de img décrits par $d_2 \cup d'_1$
calculer la densité $\hat{d_h}$ de points non appariés
msk = $N\hat{d_h} > s$
mettre à jour bg
Sorties: masque du premier plan : msk

Algorithme 1: Soustraction de fond par densité de points d'intérêt

Si $\hat{d_h}(x) > s$, alors

$$bg_n(x) = bg_{n-1}(x) \tag{1.30}$$

Sinon

$$bg_n(x) = bg_{n-1}(x)\frac{N\hat{d_h}(x)}{s} + img_n(x)\left(1 - \frac{N\hat{d_h}(x)}{s}\right) \tag{1.31}$$

Le deuxième terme de l'équation 1.31 permet de lisser l'image de fond à la frontière des objets d'intérêt, là où $N\hat{d_h}$ est proche de s. Cela permet de prévenir la création d'arêtes saillantes artificielles dans le modèle de fond. Les conséquences de l'apparition de telles arêtes seraient la création de points d'intérêt artificiels et la modification non justifiée de la texture localement et donc la création de fausses alarmes. Ceci est illustré en figure 1.7.

1.2.2.3 Soustraction de fond par grille de descripteurs modifiés

L'approche par points d'intérêt présentée à la section précédente oblige à représenter le fond par une image. Ceci engendre des difficultés lors de la mise à jour du modèle de fond qui nécessite un lissage spatial qui au fil des mises à jour risque de dégrader l'image de fond. On peut de plus remarquer que si les points d'intérêt sont très nombreux et bien répartis dans l'image alors on peut s'interroger sur la pertinence de leur utilisation par rapport à une grille régulière. Pour ces raisons nous allons présenter une méthode reposant sur une grille régulière de descripteurs SURF. Cette approche va permettre

FIGURE 1.7 – Influence du lissage spatial sur le modèle de fond. Gauche : segmentation fond/premier plan. Milieu : mise à jour du fond sans lissage. Un saut de luminosité apparait à la frontière de la zone mise à jour et modifie la texture de l'image. Droite : mise à jour du fond avec lissage. La texture originale est conservée.

une représentation plus classique du modèle de fond et donc d'utiliser des mécanismes plus simples pour sa mise à jour.

L'approche par blocs de Chen *et al.* [22] ayant démontré son efficacité, nous allons l'utiliser avec le descripteur SURF [7] présenté en section 1.2.2.1. L'idée est de combiner les gains en robustesse de l'approche par bloc avec la faible sensibilité du descripteur SURF aux changements de luminosité. Le descripteur SURF présente cependant une faiblesse. Il a en effet été développé pour mettre en correspondance des points d'intérêt en utilisant une information de texture. Dans un cas d'utilisation classique, ces points d'intérêt sont trouvés avec un détecteur garantissant une information de texture suffisante pour la pertinence du descripteur. Dans notre cas le descripteur est calculé sur une grille uniforme de l'image et donc potentiellement dans des régions homogènes ne contenant que très peu d'information de gradient. Dans ces régions le bruit de la caméra ou les artefacts de compression JPEG peuvent être plus importants que l'information réelle de texture de la scène. Or le descripteur SURF original est normalisé pour conserver un maximum d'invariance aux changements de contraste. Ceci a pour effet que dans ces régions le descripteur n'est pas pertinent et son utilisation peut engendrer des faux positifs comme on peut le constater en figure 1.8.

Pour pallier ce problème nous allons pondérer les descripteurs en fonction de la quantité d'information qu'ils contiennent. L'idée est que si le descripteur est calculé sur une zone texturée sa norme sera 1, comme pour le descripteur original. Si il est calculé sur une zone non texturée sa norme sera fixée 0, car il ne contient pas d'information. Enfin il y a une transition continue entre ces deux cas. Afin de mesurer le caractère "texturé" de la zone où est

FIGURE 1.8 – Bien que peu visibles les artefacts JPEG peuvent causer des faux positifs sur les régions les plus homogènes de la scène. En effet le descripteur SURF est normalisé et donne trop d'importance au bruit dans les zones peu texturées. Gauche : Image originale. Milieu : Détection sans correction de la norme. Droite : Détection attendue avec correction.

calculé le descripteur on peut utiliser sa norme avant qu'il ne soit normalisé. En effet par construction du descripteur cette norme est un indicateur du caractère texturé du voisinage du point d'intérêt puisque ses composantes sont des sommes de gradients. Ceci est confirmé visuellement comme on peut le constater sur la figure 1.9. Expérimentalement les zones les moins texturées sont celles où la norme de SURF est la plus faible et ces zones correspondent à celles où les artefacts de compression posent problème.

La pondération du descripteur est alors donnée par l'équation 1.33 :

$$D'_{SURF} = D_{SURF} \times f(\|D_{SURF}\|) \tag{1.32}$$

avec

$$f(x) = \begin{cases} 0 & \text{si } x < n_{min} \\ \dfrac{x - n_{min}}{n_{max} - n_{min}} & \text{si } n_{min} \leq x \leq n_{max}, \\ 1 & \text{si } n_{max} < x \end{cases} \tag{1.33}$$

où D_{SURF} est le descripteur SURF original, $\|D_{SURF}\|$ est la norme de D_{SURF} avant sa normalisation, f est la fonction de re-pondération du descripteur, n_{max} et n_{min} sont deux paramètres. Puisque les artefacts de compression ou le bruit engendrent un gradient faible, le descripteur associé sera re-pondéré et aura une norme proche de 0. Sans cette pondération la normalisation à 1 par défaut aurait entrainé une fausse alarme due à un changement minime de texture.

Pour les évaluations de cette méthode nous allons utiliser un post-traitement inspiré du calcul de densité de l'approche par point d'intérêt. Des gaussiennes

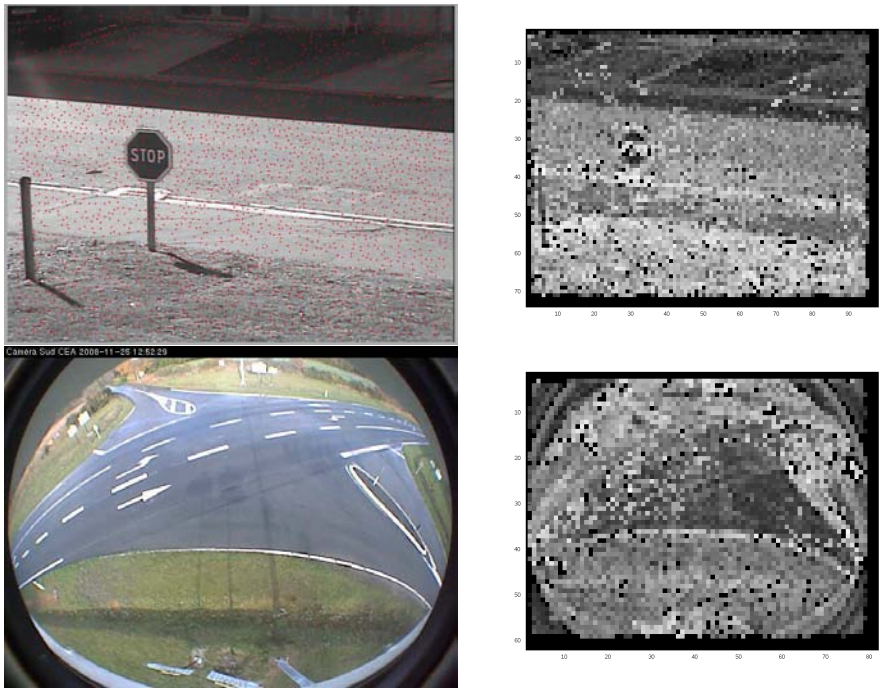


FIGURE 1.9 – Gauche : Image de la scène. Droite : image en niveaux de gris de la norme du descripteur SURF [7] avant normalisation. Noir pour les faibles normes, blanc pour les plus importantes

centrées sur les points de la grille sont sommées puis l'image de densité est seuillées. Cela permet d'avoir des frontières lisses, comme on peut le constater en figure 1.9. Cependant dans les chapitres suivants nous utiliserons comme post-traitement une fermeture morphologique sur les blocs de la grille. Ce dernier donne des résultats de qualité similaire mais à l'avantage d'être plus rapide en temps d'exécution.

1.2.3 Évaluation

La première partie de l'évaluation ne concerne que le descripteur SURF et SURF modifié. Nous avons pour cela effectué deux expériences pour tester la qualité du descripteur dans un contexte de soustraction de fond. Sur une séquence pour laquelle nous disposons d'une vérité terrain de soustraction de fond nous calculons des descripteurs répartis sur une grille régulière. Nous allons alors pouvoir comparer deux descripteurs provenant d'un même bloc mais à des instants différents. De cette manière les descripteurs sont bien testés en dehors du système de soustraction de fond mais tout en restant

dans son contexte d'utilisation. Pour ce faire nous calculons des scores de rappel et précision.

Le rappel et la précision sont définis comme suit :

$$rappel = \frac{VP}{VP + FN} \tag{1.34}$$

$$precision = \frac{VP}{VP + FP} \tag{1.35}$$

où VP, FN et FP signifient respectivement Vrai Positif, Faux Négatif et Faux Positif.

Dans un premier temps nous comparons les descripteurs issus de deux images consécutives. La figure 1.10 montre les courbes de précision et rappel obtenues avec les descripteurs Chen, SURF original et SURF modifié. Il apparaît clairement que le descripteur SURF est beaucoup plus robuste que le descripteur de Chen *et al.* . La différence notable entre les descripteurs SURF et SURF modifié vient du fait que la séquence pour laquelle la courbe a été calculée contient des zones homogènes qui peuvent générer des faux positifs importants en proportion, comme expliqué en section 1.2.2.3.

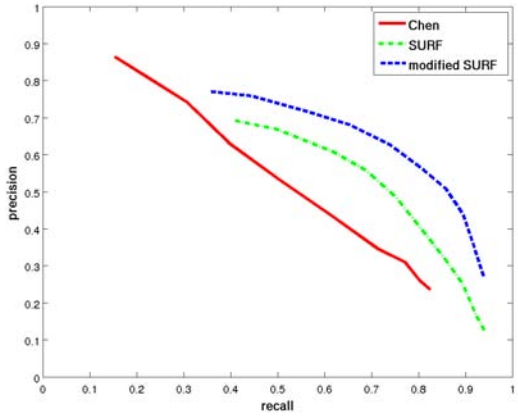


FIGURE 1.10 – Courbes de précision / rappel calculées sur une séquence. Les descripteurs de l'image courante sont comparés à ceux de l'image précédente.

La figure 1.11 montre des courbes précision / rappel calculées lors d'une expérience similaire à la précédente. Les descripteurs de référence sont calculés sur une image du début de la séquence et restent fixes pendant toute l'expérience. Là encore on observe la supériorité du descripteur SURF sur celui de Chen *et al.* . Les très faibles scores de précision obtenus sont dus aux

fortes variations de la scène au cours du temps. Cette expérience revient à avoir une soustraction de fond avec un modèle non mis à jour et sans post-traitement. Contrairement à l'expérience précédente les descripteurs SURF et SURF modifiés ont des courbes très similaires. Cela s'explique par le fait que la différence entre les deux est estompée par le nombre important de fausses alarmes.

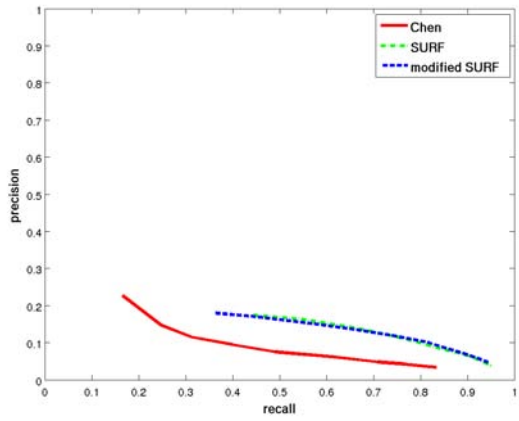


FIGURE 1.11 – Courbes de précision / rappel calculées sur une séquence. Les descripteurs de l'image courante sont comparés à ceux d'une image de référence. La faible précision est due au fait que l'image de référence, sélectionnée en début de séquence devient au cours du temps très différente de l'image courante à cause notamment des nombreux changements de luminosité.

La seconde partie de l'expérimentation concerne le système de soustraction de fond complet. Les résultats suivants sont issus de séquences acquises avec une caméra fixe. Pour se replacer dans notre cadre d'utilisation, celui de caméras PTZ effectuant un tour de garde et où chaque position de la caméra peut être assimilée à une caméra fixe avec un faible rafraichissement, nous ne traitons qu'une image sur 500. Cela correspond à une image toutes les 20 secondes. Parmi les diverses séquences sur lesquelles notre algorithme a été testé, la séquence *train* acquise avec une caméra montée dans un train est particulièrement intéressante en terme de changements soudains de luminosité. Les séquences *Pets* bien que tournées en intérieur permettent de tester sur des séquences montrant des zones moins texturées. La séquence *changement de luminosité1* montre des conditions extrêmement difficiles pour la soustraction de fond, parmi lesquelles des fortes ombres portées, des changements soudains de luminosité ou encore de la saturation due à une forte

luminosité. La séquence *changements de luminosité 2* présente des changements de luminosité et des zones homogènes. Des résultats qualitatifs sur ces séquences sont montrés en figure 1.12.

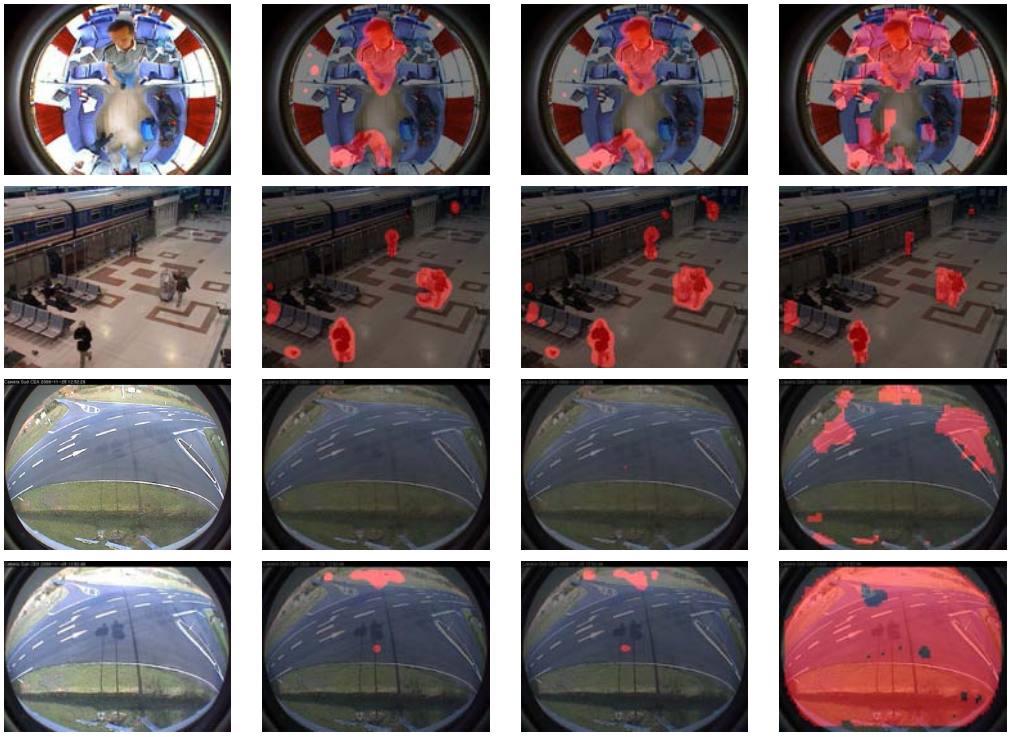


FIGURE 1.12 – Résultats qualitatifs obtenus sur diverses séquences. Première ligne : séquence *train*. Deuxième ligne : séquence *PETS 2006*. Les troisième et quatrième lignes correspondent à deux images consécutives de la séquence *changement de luminosité*. Première colonne : image originale. Deuxième colonne : grille de descripteurs SURF modifiés. Troisième colonne : densité de point non appariés. Quatrième colonne : Chen *et al.* .

La figure 1.13 donne des courbes de précision/rappel sur les séquences *changement de luminosité 2*, *changement de luminosité 1* et *train*. On peut constater que l'approche par grille de descripteurs donne des résultats au moins aussi bons que l'approche par points d'intérêt et largement supérieurs à ceux de Chen *et al.* .

On peut constater que les résultats de nos deux approches sont comparables ou supérieurs à ceux obtenus par l'approche de Chen *et al.* . Les différences de précision observées sont dues seulement aux fausses alarmes sur les zones les plus homogènes.

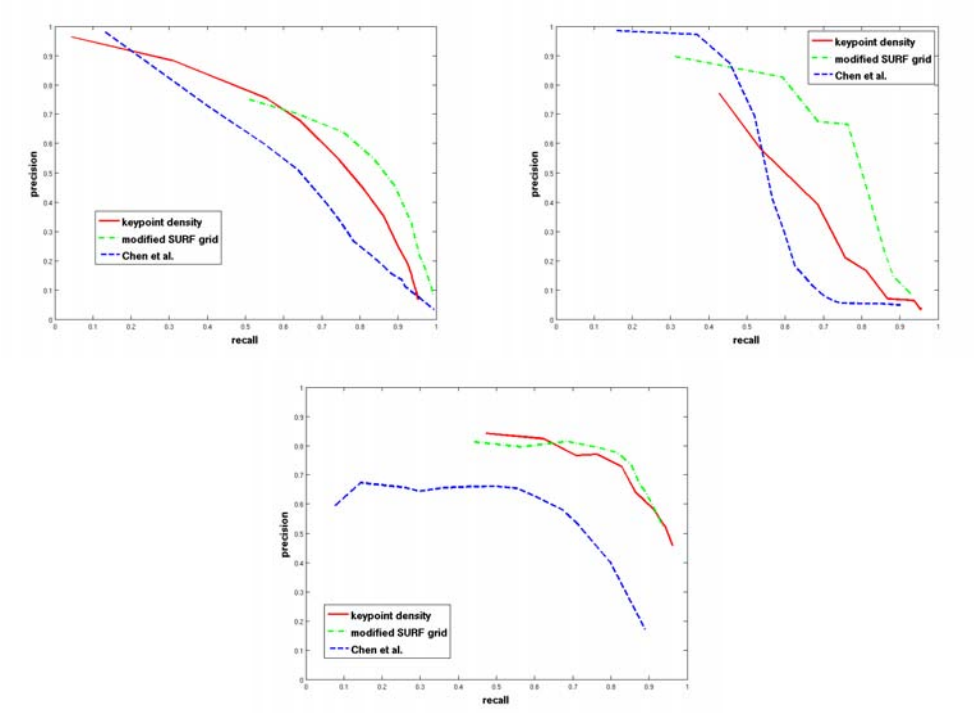


FIGURE 1.13 – Courbes de précision/rappel. Gauche : séquence *changement de luminosité 2*, Droite : séquence *changement de luminosité 1*. Bas : séquence *train*. Vert : SURF modifié, Rouge : Densité de SURF non appariés, Bleu : Chen *et al.* .

D'autres résultats qualitatifs obtenus sur des séquences acquises par une caméra PTZ en mode tour de garde sont montrés en figure 1.14

Finalement les figures 1.15 et 1.16 montrent des résultats sur des passages particulièrement difficiles de la séquence *changement de luminosité 1*. Si l'on peut remarquer la présence de quelques fausses alarmes, elles se trouvent principalement sur des zones de l'image où le brusque changement de luminosité a fait apparaître des nouveaux contours (ombre portée) ou une saturation importante. Là où la texture de l'image n'est pas altérée notre algorithme se comporte remarquablement bien.

1.2.4 Conclusion sur la soustraction de fond

Nous avons proposé une nouvelle approche de soustraction de fond et détection de région d'intérêt. Notre objectif était de répondre aux contraintes auxquelles sont soumises les caméras PTZ effectuant un tour de garde. Nous

FIGURE 1.14 – PTZ Sequence.

avons pour cela proposé d'utiliser un descripteur robuste (SURF) qui nous permet de gérer des fréquences d'acquisition faibles. Une comparaison avec des méthodes classiques de l'état de l'art ont montré l'apport de nos approches.

Nos deux algorithmes reposants sur des points d'intérêt ou une grille régulière montrent des résultats assez similaires dans les zones texturées de l'image. On peut cependant remarquer que l'approche par grille est par nature plus souple à utiliser. En effet son modèle de fond n'est pas une image mais un ensemble de descripteurs répartis sur une grille. Il est par conséquent très simple avec cette approche de gérer les fonds multi-modaux en autorisant plusieurs descripteurs de fond pour chaque blocs. De plus, de la même façon qu'est défini le modèle de fond, il est aisé de définir un modèle multi-modal du premier plan. Les principales caractéristiques de notre modèle de fond sont sa simplicité, sa robustesse aux changements de luminosité conférée par l'utilisation des descripteurs SURF et sa rapidité d'exécution par l'utilisation de blocs 8×8.

Pour ces raisons nous allons par la suite utiliser l'approche par grille de descripteurs SURF modifiés. On utilisera aussi un modèle du premier plan défini de manière similaire. Ce modèle du premier plan n'a pas d'influence sur le modèle de fond, il sera simplement utilisé en plus pour conserver un historique du premier plan.

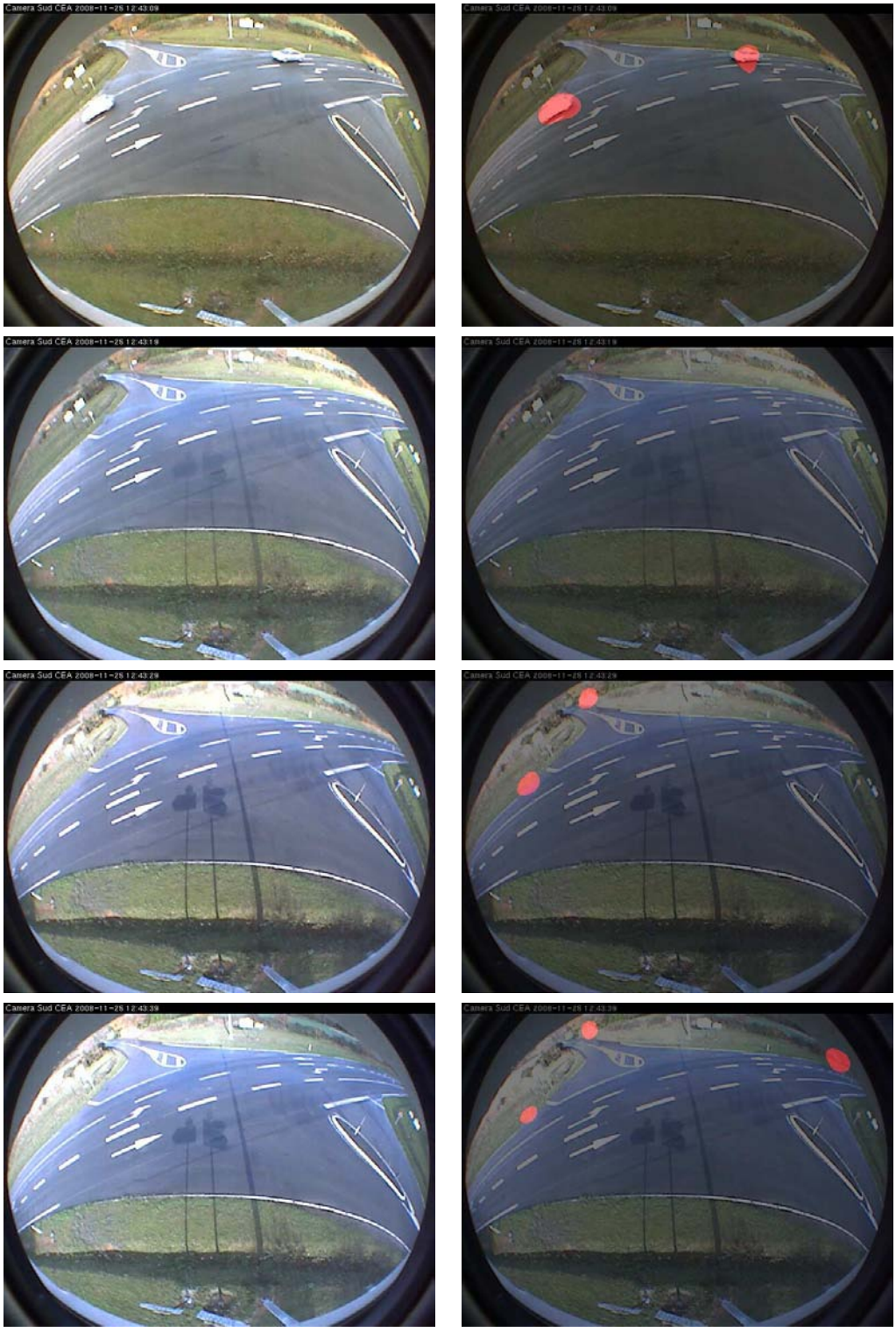


FIGURE 1.15 – Séquence *changements de luminosité 1*. Les lignes montrent des images consécutives, mais avec un faible taux de rafraîchissement : une image toutes les dix secondes. La colonne de gauche est l'image originale, la colonne de droite est la sortie notre algorithme par grille de descripteurs.

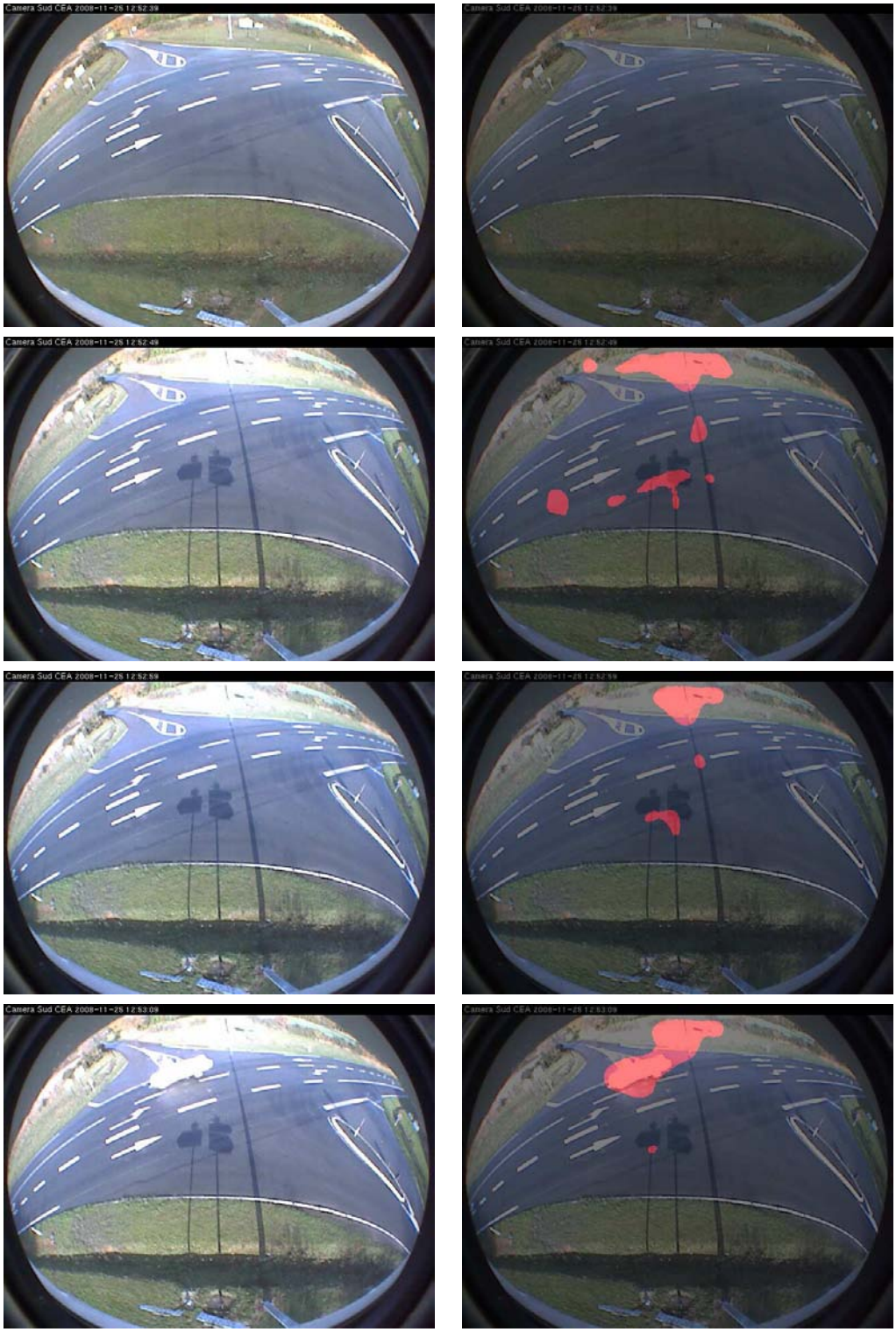


FIGURE 1.16 – Séquence *changements de luminosité 1.* Les lignes montrent des images consécutives, mais avec un faible taux de rafraîchissement : une image toutes les dix secondes. La colonne de gauche est l'image originale, la colonne de droite est la sortie notre algorithme par grille de descripteurs.

1.3 Détection d'objets stationnaires

Dans cette section nous allons présenter un algorithme de détection d'objets stationnaires reposant sur une caractéristique importante de notre modèle du premier plan qui est la possibilité de pouvoir réidentifier de manière robuste des blocs de l'image déjà observés.

Bien que nous parlions de détection *d'objet* stationnaire, notre algorithme, de même qu'une grande majorité de ceux de la littérature, ne cherche pas à réellement à détecter spécifiquement des *objets* mais plutôt des régions de l'image où le premier plan est stationnaire. Le terme *objet* est utilisé par abus car généralement seuls les objets sont effectivement stationnaires et l'application recherchée est bien leur détection.

1.3.1 État de l'art

Depuis quelques années beaucoup de méthodes ont été développées pour détecter des objets stationnaires. L'application visée est principalement la vidéo-surveillance pour la détection d'objets abandonnés, volés, ou encore la détection de véhicules mal garés. Une très grande majorité de ces méthodes repose sur des algorithmes de soustraction de fond. Bayona *et al.* [8] donnent une comparaison de certains de ces algorithmes et proposent de les séparer en deux catégories. La première catégorie d'algorithmes n'utilise qu'un seul modèle de fond et contient une phase d'analyse reposant sur l'accumulation de plusieurs masques binaires de premier plan, du tracking ou encore des caractéristiques intrinsèques au modèle de fond. La seconde catégorie de méthodes repose sur l'utilisation de plusieurs modèles de fond. Le principe général est de construire les différents modèles ayant des caractéristiques temporelles différentes et de les comparer.

Mathew *et al.* [71] utilisent le modèle de fond de Stauffer et Grimson [95] avec quatre gaussiennes par pixel. A chacune de ces gaussiennes est affecté l'un des trois états suivants : premier plan (FG), fond (BG) ou fond dominant (DBG). Ces états et les différentes transitions possibles sont montrés en figure 1.17.

L'idée générale est alors d'observer les transitions des états des gaussiennes. Les auteurs ont mis en évidence quatre conditions nécessaires pour qu'un objet soit considéré stationnaire :

1. Les distributions correspondant à de nouveaux objets stationnaires passent de l'état FG à DBG. Le temps τ_{BDG} de passage à l'état DBG est donc enregistré.

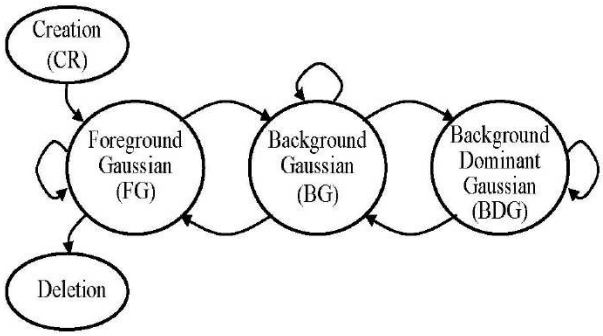


FIGURE 1.17 – Graphe des états et transitions possibles pour chaque gaussienne pour l'algorithme de Mathew *et al.* [71].

2. Les nouveaux objets stationnaires doivent correspondre à une nouvelle gaussienne, créée au temps τ_{CR}. Une gaussienne passant à l'état DBG ne sera considérée comme un nouvel objet que si $\tau_{DBG} - \tau_{CR} < \Gamma_{CR \to BDG}$.
3. La nouvelle gaussienne doit rester dans l'état DBG pendant au moins un certain intervalle de temps Γ_{BDG}.
4. Le poids de la gaussienne dans l'état BDG doit être supérieur à $0,5$. Cela permet de prévenir des fausses alarmes dans le cas par exemple des fonds dynamiques en s'assurant que la gaussienne principale est la même pour une période de temps suffisamment longue.

Les pixels affectés à une gaussienne dominante vérifiant les 4 conditions sont ceux qui appartiennent à un nouvel objet stationnaire.

Guler *et al.* [41] ont eux une approche reposant sur le suivi d'objets. Une étape préliminaire consiste à suivre les régions S_i en mouvement. Elles deviendront par la suite autant d'hypothétiques objets stationnaires. Quand une région S_i est créée une probabilité de stationnarité sP_i de la région est initialisée à 0. Cette probabilité est définie en supposant chaque pixel comme indépendant :

$$sP_i = \prod_{(x,y) \in S_i} sp_{x,y}, \text{ avec } sp_{x,y} \in [0,1] \tag{1.36}$$

A chaque nouvelle image traitée les probabilités des régions existantes sont mises à jour. Pour ce faire, chaque pixel est classé comme fond ou premier plan en regardant s'il est semblable au modèle de fond. Si c'est le cas sa probabilité de stationnarité est décrémentée :

$$sp_{x,y} = sp_{x,y} - n\frac{t - t_i}{t_{idle}} \tag{1.37}$$

Dans le cas où le pixel est classé comme du premier plan, sa probabilité de stationnarité est augmentée :

$$sp_{x,y} = sp_{x,y} + \frac{t - t_i}{t_{idle}} \text{ si } t < t_i + t_{idle} \tag{1.38}$$

Avec t le temps courant, t_i le temps de création de la région S_i, t_{idle} le paramètre spécifiant la vitesse à laquelle un objet est considéré stationnaire, $n > 1$ est un paramètre entier.

Pour chaque image, seules les régions S_i dont la probabilité sP_i dépasse un certain seuil seront considérées comme des objets stationnaires.

Ce type d'approche nécessite de pouvoir suivre les différents objets de la scène. Elle ne pourrait donc pas être appliquée dans notre contexte de caméra à très faible taux de rafraîchissement.

Un nombre plus important de méthodes repose sur l'accumulation de masques binaires du premier plan échantillonnés sur un intervalle de temps. Bayona *et al.* [8] ont montré expérimentalement que ce sont ces méthodes qui donnent les meilleurs résultats. Liao *et al.* [65] pour détecter les objets qui sont restés stationnaires pendant les 30 dernières secondes au moins retiennent 6 images $F_1, \ldots, F_6$ régulièrement espacées dans cet intervalle de temps. Un algorithme de soustraction de fond est alors appliqué à ces images pour générer 6 masques binaires $M_1, \ldots, M_6$ du premier plan.

$$M_k(i) = \begin{cases} 1 \text{ si } |F_k(i) - B(i)| > w_i\sigma_i \\ 0 \text{ sinon} \end{cases} \tag{1.39}$$

où B est l'image de fond, σ_i est l'écart type du pixel i, w_i est un poids fixé arbitrairement pour prendre en compte le changement de résolution entre la partie basse (proche de la caméra) et haute (loin de la caméra) de l'image. Le masque final des objets stationnaires S est alors le ET logique des 6 masques M_k :

$$S = \bigwedge_{k=1}^{6} M_k \tag{1.40}$$

Un filtre est alors appliqué au masque S pour supprimer une partie du bruit. Les auteurs estiment alors que S va "*très probablement*" correspondre aux objets stationnaires. On peut cependant remarquer qu'il suffit que des objets en mouvement soient présents au même endroit sur chacun des masques

M_k pour que ces zones soient considérées comme un objet stationnaire dans le masque final. Ceci peut facilement arriver si par exemple la scène comprend un flot continu de personnes se déplaçant. Bayona *et al.* [9] proposent d'améliorer cette méthode pour pallier ces problèmes. Un masque binaire des objets stationnaires S est défini de la même façon, mais pour diminuer le nombre de faux positifs dus à cette méthode, ils créent de manière similaire les masques des objets en mouvement $FD_1, \ldots, FD_6$ obtenus par le seuillage d'une simple différence entre deux images. Le masque IFD représentant les zones jamais en mouvement est alors défini comme suit :

$$IDF = \bigwedge_{k=1}^{6} \neg FD_K \tag{1.41}$$

A partir de S et IFD on peut définir le masque FM des objets stationnaires jamais occultés :

$$FM = S \wedge IFD \tag{1.42}$$

Enfin pour pouvoir détecter les objets stationnaires même s'ils ont été occultés par un objet en mouvement il faut détecter les occultations. Une liste de blobs est extraite du masque FM et est comparée à la liste des blobs présents au temps précédent. Si un blob est manquant cela signifie qu'un objet stationnaire n'est plus présent ou qu'il est occulté. Le cas d'occultation est facilement détecté en vérifiant si le blob a bien été détecté par la soustraction de fond et si le masque de mouvement FD est actif pour cette zone. En cas d'occultation avérée le blob manquant est ajouté au masque FM.

Porikli *et al.* [87] détectent les objets stationnaires en utilisant deux modèles de fond. Le premier modèle, B_S dit *court terme*, intègre assez rapidement les modifications de l'environnement alors que le second, B_L dit *long terme*, est beaucoup plus stable. L'idée est de considérer que les objets stationnaires sont les objets intégrés par le modèle court terme mais pas par le modèle long terme. Pour cela, à chaque nouvelle image, B_S et B_L sont utilisés pour calculer les deux masques binaires du premier plan F_S et F_L. Un objet temporairement stationnaire correspond au cas où $F_L \wedge \neg F_S$.

Les détections dans chaque image sont accumulées au cours du temps dans une image E de la manière suivante :

$$E(i) = \begin{cases} E(i) + 1 \text{ si } F_L(i) \wedge \neg F_S(i) \\ E(i) - k \text{ si } \neg F_L(i) \vee F_S(i) \\ max_e \text{ si } E(i) > max_e \\ 0 \text{ si } E(i) < 0 \end{cases} \tag{1.43}$$

où i est un pixel, max_e et k sont des nombres positifs. Un pixel est considéré comme abandonné lorsque $E(i) \geq max_e$. Accumuler les observations de cette manière dans une image permet notamment de limiter le bruit et de pouvoir facilement choisir le seuil temporel de détection.

Les approches reposant sur la combinaison de masques binaires, telles que celle de Bayona *et al.* [9] ne ré-identifient pas les objets au cours du temps. Elles ne permettent pas de détecter au plus tôt les objets stationnaires qui sont complètement occultés pendant un certain temps avant de pouvoir être considérés comme stationnaires. Cela est problématique, et ce particulièrement dans notre cadre applicatif, puisque avec un taux de rafraîchissement réduit les retards de détection se cumuleraient d'autant plus. Toujours au cause du taux réduit de rafraîchissement l'approche de Guler *et al.* [41] ne peut être utilisée dans notre cas puisque nous ne pouvons pas suivre les objets. Pour ces raisons nous allons proposer une approche originale reposant sur notre algorithme de soustraction de fond par grille de descripteurs, et qui permet de ré-identifier les objets du premier plan.

1.3.2 Notre approche

Plutôt que de travailler sur des masques de détection comme ont pu le faire Bayona *et al.* [9], nous allons utiliser une caractéristique du modèle de fond présenté en section 1.2.2.3. Les descripteurs SURF [7] utilisés pour représenter le fond peuvent aussi être utilisés pour représenter le premier plan. Si un descripteur d'un bloc de l'image courante n'a pas pu être apparié à un descripteur du modèle de fond alors il est comparé aux descripteurs du modèle d'objet du premier plan. Si un appariement est trouvé, alors le modèle objet est mis à jour, sinon un nouveau mode est créé et son temps de création $t_{création}$ est enregistré.

Les objets stationnaires visibles de l'image courante sont alors définis naturellement comme les blobs constitués des blocs du premier plan vérifiant :

$$t - t_{création} \geq t_{stationnaire} \tag{1.44}$$

où t est le temps courant et $t_{stationnaire}$ est une constante représentant la durée nécessaire pour considérer qu'un objet est stationnaire.

FIGURE 1.18 – Objet stationnaire détecté partiellement occulté. Un masque binaire permet de conserver en mémoire les blocs qui ont été détectés comme stationnaires et donc de préserver la forme des objets stationnaires même lorsqu'ils sont occultés. Un pixel du masque n'est remis à zéro que si le fond est de nouveau observé.

Pour gérer les occultations temporaires des objets stationnaires un masque binaire est utilisé. A chaque fois qu'un bloc de l'image est classé objet stationnaire la valeur du pixel correspondant du masque est mise à 1. Lorsque le fond sera de nouveau observé, cette valeur sera remise à 0 et le modèle des objets du premier plan sera supprimé. Comme illustré par la figure 1.18, ce masque permet de garder en mémoire les blocs de l'image qui ont été classés stationnaires mais dont on a pas encore observé la disparition.

1.3.3 Évaluation

Il n'y a pas dans la littérature de protocole commun pour évaluer les algorithmes de détection d'objets stationnaires.

La qualité d'un système de détection d'objets stationnaires est de segmenter les objets correctement. Une mesure d'évaluation d'un tel système pourrait donc être similaire à celle d'un algorithme de soustraction de fond. Le tableau 1.1 nous donne des valeurs de précision et rappel de notre algorithme calculées sur des séquences I-Lids pour AVSS 2007. Contrairement à certains auteurs nous avons fait le choix dans notre évaluation de ne pas négliger dans nos calculs les zones où se trouvent des personnes assises, même si elles génèrent des faux positifs. Certaines de ces personnes sont en effet immobiles ou partiellement immobiles. Ceci a pour effet de faire fortement baisser les valeurs de précision notamment sur la séquence AB Hard, comme

illustré en figure 1.19. Bien qu'appliqué à la détection d'objets stationnaires notre algorithme est en fait un détecteur des régions de l'image qui sont stationnaires.

Séquence	Précision	Rappel
AB Easy	0,70	0,89
AB Medium	0,77	0,61
AB Hard	0,46	0,77
PV Easy	0,73	0,94
PV Medium	0,61	0,79
PV Hard	0,8	0,5

TABLE 1.1 – Tableau de précision et rappel de notre algorithme calculés sur les séquences publiques I-Lids pour AVSS 2007. L'évaluation est ici au niveau pixélique. Le masque binaire de détection est comparé à celui de la vérité terrain.

FIGURE 1.19 – Séquence AVSSAB Hard. Cas où la détermination du nombre de vraies et fausses alarmes reste subjective. Le blob contenant la personne assise peut être considéré comme vrai positif puisqu'il englobe un objet stationnaire (le bagage à ses pieds). Il peut aussi être considéré comme un faux positif car la personne bien que bougeant peu n'est pas exactement stationnaire. Dans ce dernier cas la surface détectée est beaucoup plus importante que la surface du bagage seulement.

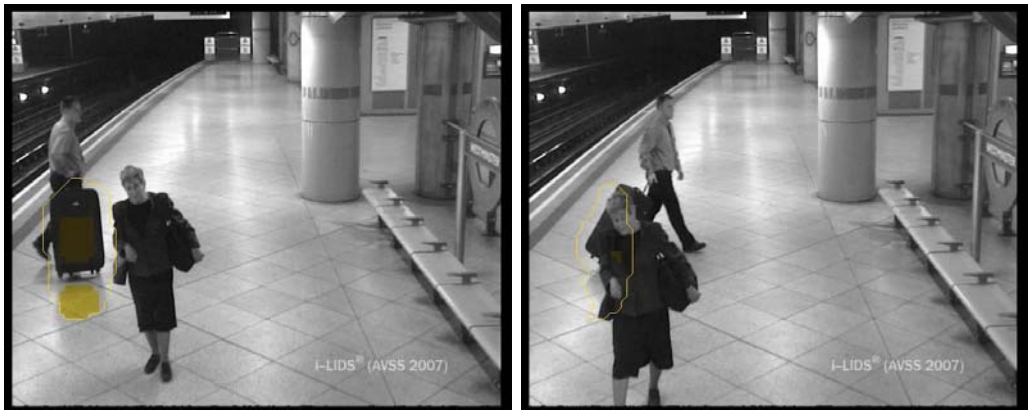


FIGURE 1.20 – Séquence AVSS AB Easy. Fausses alarmes : persistance de la détection suite à la mise en mouvement d'un objet stationnaire et à une occultation. Bien que l'objet ne soit plus stationnaire l'alarme est toujours levée car le fond n'a pas été re-observé.

On peut cependant penser que notre système est plus efficace que ce que laissent voir les chiffres du tableau 1.1. Une bonne partie de la faiblesse en précision est en effet due à une segmentation trop large des objets (figures 1.19 et 1.18). Lorsque l'on doit détecter de petits objets, qui ont une surface de 15 blocs par exemple dans le cas de la séquence PV Hard, une erreur de quelques blocs seulement fait baisser les scores de manière importante.

Face à la difficulté d'interprétation de ces chiffres il parait nécessaire de définir une mesure plus en phase avec l'application recherchée. D'un point de vue application la qualité de la segmentation n'est que "relativement" importante. Le principal pour un opérateur qui utiliserait un tel système serait que l'alarme se déclenche au bon moment et que la segmentation soit suffisamment bonne pour attirer son attention sur la partie intéressante de l'image. Le tableau 1.2 montre les temps de début et de fin d'alarme pour l'événement principal des séquences I-Lids. On peut constater que si le début des alarmes correspond à la vérité terrain (pire cas : une seconde de retard) le moment où l'alarme est levée peut atteindre neuf secondes après la fin réelle de l'alarme. Ceci s'explique facilement par le fait que lorsque un objet stationnaire redevient mobile, notre algorithme le considère comme une occultation quelconque puisque nous n'utilisons que des information locales de l'image (ie pas de suivi d'objet par exemple). Ce phénomène est illustré en figure 1.20.

Afin de mieux prendre en compte la qualité de la segmentation du point

Séquence	Début vérité terrain (s)	Début détecté (s)	Fin vérité terrain (s)	Fin détectée (s)
AB Easy	2 :20	2 :20	3 :14	3 :18
AB Medium	1 :58	1 :58	3 :02	3 :03
AB Hard	1 :51	1 :52	3 :07	3 :11
PV Easy	2 :48	2 :48	3 :15	3 :21
Guler [41]		2 :46		3 :18
Venetianer [103]		2 :52		3 :16
PV Medium	1 :28	1 :28	1 :47	1 :56
Guler [41]		1 :28		1 :54
Venetianer [103]		1 :43		1 :47
PV Hard	2 :12	2 :12	2 :33	2 :35
Guler [41]		2 :13		2 :36
Venetianer [103]		2 :19		2 :34

TABLE 1.2 – Temps des détections des événements principaux sur les séquences I-Lids [1]. Les fins alarmes sont détectées avec du retard sur la vérité terrain car lorsque l'objet d'intérêt redevient mobile il est considéré comme un objet mobile occultant potentiellement les objets stationnaires.

de vue de l'application et des difficultés d'interprétation (comme dans le cas de la figure 1.19) nous avons effectué un autre type d'évaluation. Nous considérons un événement positif (VP) lorsque dans une image le pourcentage de la surface d'un blob détecté stationnaire qui recouvre effectivement un blob de la vérité terrain est supérieur à un certain seuil $\tau_{Positif}$. Réciproquement nous considérons qu'il y a un événement négatif (FN) lorsqu'un blob de la vérité terrain n'a pas été recouvert au delà d'un certain seuil $\tau_{Négatif}$ par une de nos détections. Avec cette définition un même blob peut être considéré à la fois comme un Vrai Positif et un Faux Positif. Ce serait par exemple le cas si, alors qu'un objet stationnaire est effectivement présent dans la scène, une alarme constituée d'un blob comprenant l'image entière était levée. Les figures 1.21 et 1.22 montrent les taux de VP et de FP parmi les alarmes levées en fonction des seuils de recouvrement $\tau_{Positif}$ et $\tau_{Négatif}$ choisis.

Les courbes plutôt mauvaises de la séquence *AVSS PV Medium* s'expliquent par le fait que la durée de stationnarité du véhicule est relativement courte et que dans un premier temps (à cause d'une occultation partielle lors de son arrivée) elle n'est qu'à moitié détectée. Le taux de faux positifs s'explique quant à lui par la persistance de la détection (cf. table 1.2) due au passage de voitures occultant l'ancienne position de la voiture. Ceci est

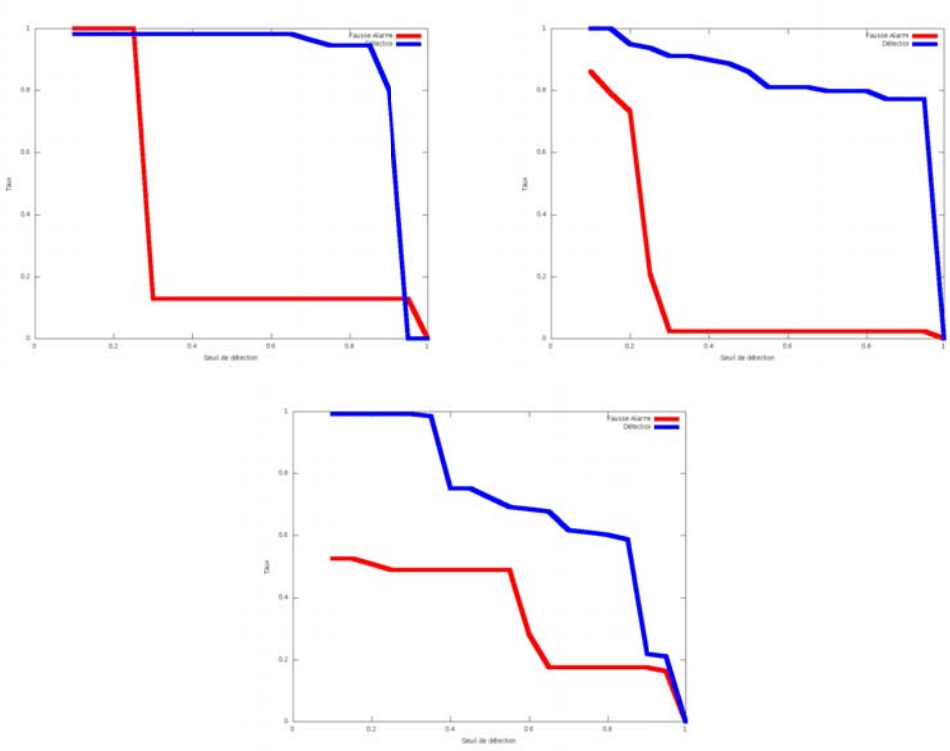


FIGURE 1.21 – De gauche à droite : AB Easy, AB Medium, AB Hard. Bleu : taux de VP parmi les alarmes levées. Rouge : taux de FP parmi les alarmes levées. Séquence AB Hard : la forte baisse du taux de FP est due à la petitesse d'un des objets stationnaires détectés, dans ce cas une sur-détection de quelques blocs fait qu'il n'est rapidement plus compté comme positif.

illustré en figure 1.23.

1.3.4 Conclusion sur la détection d'objets stationnaires

Nous avons proposé un algorithme de détection d'objets stationnaires reposant sur la ré-identification du premier plan modélisé par des descripteurs SURF. L'originalité de notre approche est l'exploitation des descripteurs associés aux blocs de l'image qui permettent la modélisation du premier plan et la ré-identification des blocs. Il en résulte que nous avons une estimation directe du temps de présence de chaque bloc de l'image, ce qui nous permet de les classifier facilement comme stationnaire ou non avec une grande robustesse aux occultations temporaires.

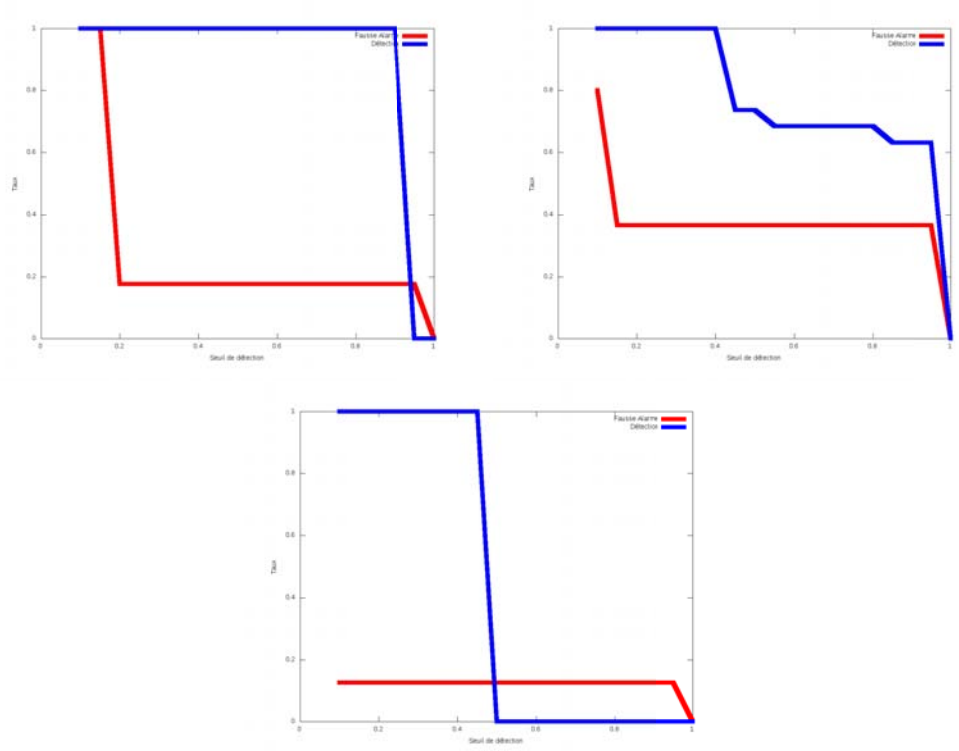


FIGURE 1.22 – De gauche à droite : PV Easy, PV Medium, PV Hard. Bleu : taux de VP parmi les alarmes levées. Rouge : taux de FP parmi les alarmes levées. Séquence PV Hard : la forte baisse du taux de FP est due à la petitesse du véhicule stationnaire détecté, dans ce cas une sur-détection de quelques blocs suffit pour qu'il ne soit plus comptabilisé parmi les positifs.

1.4 Conclusion

Nous avons dans ce chapitre introduit une méthode de soustraction de fond robuste aux changements soudains de luminosité. Nous avons proposé une approche par grille de descripteurs qui s'est avérée à la fois simple à mettre en oeuvre et efficace tant du point de vue temps de calcul que qualité de détection. Notre approche a montré des résultats convaincants face aux méthodes de l'état de l'art. Ses performances seront déterminantes dans le contexte d'une caméra PTZ effectuant un tour de garde ou de manière équivalente dans le cas des caméras fixes ayant un faible taux de rafraîchissement.

De plus l'utilisation d'un descripteur robuste dans le modèle du fond, mais aussi dans le modèle du premier plan, nous permet la ré-identification robuste du premier plan et ainsi la détection des régions stationnaires dans un

flux vidéo. Nous avons montré que notre système, simple et efficace, permet en particulier de gérer les cas d'occultation, fréquents dans le contexte de la détection d'objets stationnaires.

Afin d'affiner les capacités d'analyse de notre système, nous allons dans le chapitre suivant proposer une méthode permettant sous certaines conditions de séparer différents objets stationnaires contenus dans un unique blob. Nous proposerons ensuite une méthode originale d'appariement des blobs détectés dans une paire de caméras et qui permet plus d'informations et de robustesse que la vision monoculaire.

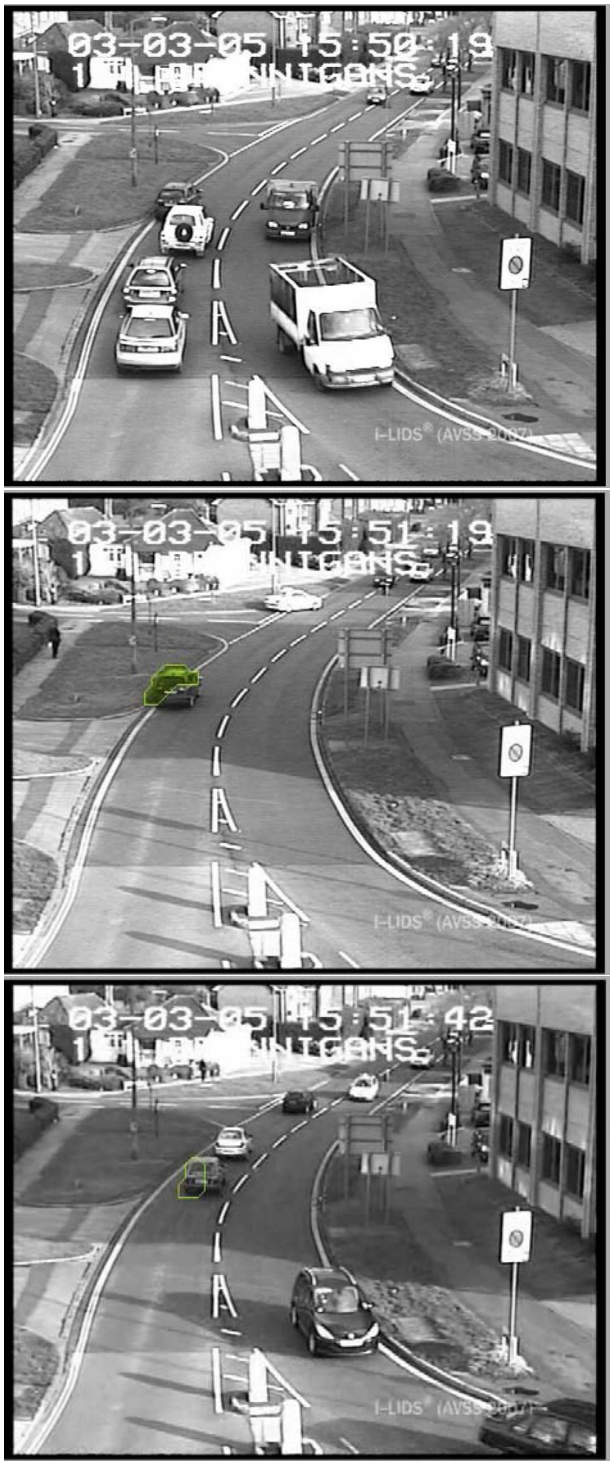


FIGURE 1.23 – Exemple de résultats sur la séquence AVSS PV Medium. Haut : Occultation partielle et prolongée du véhicule lors de son arrivée. Milieu : Détection partielle du véhicule due à l'occultation originale. Ceci explique le fort taux de FN. Bas : Persistance de la détection après le départ du véhicule à cause d'une occultation. Ceci explique le faible taux de FP sur cette séquence.

Chapitre 2

Étiquetage et appariement d'objets stationnaires

2.1 Introduction

Dans la partie précédente nous avons détecté de manière robuste des régions stationnaires du premier plan. Il est cependant possible que plusieurs objets se retrouvent dans un unique blob. L'algorithme de détection présenté au chapitre précédent ne permet pas de les distinguer.

Cette partie s'intéresse à la segmentation mono-caméra des objets stationnaires puis à leur mise en correspondance dans une paire de caméras. L'objectif de la méthode décrite dans le premier paragraphe est de tirer un maximum d'informations des détections de chaque caméra pour arriver, lorsque c'est possible, à segmenter les objets appartenant à un même blob de soustraction de fond. Cette nouvelle segmentation plus riche que la détection de régions stationnaires présentée au chapitre précédent pourra alors être utilisée dans un algorithme d'appariement pour mutualiser les détections de chaque caméra.

2.2 Segmentation mono-caméra

2.2.1 Objectifs de la segmentation mono-caméra

Dans l'idéal nous aimerions avoir une segmentation fine de tous les objets présents dans l'image même s'ils s'occultent partiellement, comme c'est le cas en figure 2.1.

Cependant, nous faisons le choix de ne pas utiliser de critère d'homogénéité d'apparence (couleur, texture) qui risque de provoquer une sur-segmentation des objets dont la texture n'est pas homogène. Nous choisissons d'utiliser des critères temporels pour séparer les objets apparus à des instants différents. Par conséquent si deux objets sont posés cote à cote simultanément notre critère ne pourra pas les séparer.

FIGURE 2.1 – Deux objets contenus dans un seul blob. Un unique masque binaire des objets stationnaires ne contiendrait qu'une composante connexe. Il ne permettra donc pas de distinguer les deux objets même s'ils sont apparus à des instants différents.

Ce problème parait pouvoir être facilement résolu en utilisant un seuil sur le temps d'apparition des objets, mais cela mènerait à une sur-segmentation en cas d'apparition d'un objet sous occultation partielle, comme illustré en figure 2.2. La valise, qui est un objet stationnaire, est toujours partiellement occultée par une personne. Certains blocs de la valise auront donc des âges très différents et un simple seuil sur le temps d'apparition des blocs ne peut pas être un critère suffisant pour résoudre notre problème.

Nous avons identifié trois cas que nous souhaitons pouvoir gérer. Ces trois cas sont illustrés par les tables 2.1, 2.2 et 2.3 dans le cas d'images à une seule dimension. Ces "images" dégénérées sont constituées de quatre pixels alignés dont la valeur est représentée par une lettre. Dans ces exemples un objet est considéré stationnaire à partir de trois unités de temps. La table 2.1 représente le cas simple où un objet est posé puis partiellement occulté.

La table 2.2 représente le cas où un objet est partiellement occulté lorsqu'il est visible pour la première fois. Il est ensuite entièrement visible. Malgré la différence de temps d'apparition des parties de l'objet on souhaite qu'il soit

FIGURE 2.2 – Objet stationnaire partiellement occulté.

Temps					scénario	sortie désirée
0	∅	∅	∅	∅	scène vide	
1	A	B	C	D	valise ABCD posée	
2	A	B	C	D		
3	A	B	E	F	valise partiellement occultée par EF	objet1 : AB
4	A	B	E	F	impossible de savoir si CD ou EF sont statiques	objet1 : AB
5	A	B	C	D	valise de nouveau entièrement visible	objet1 : ABCD

TABLE 2.1 – Exemple de problème de segmentation sur une image 1D. Cas simple avec occultation.

détecté comme un seul objet.

La table 2.3 illustre le cas où un objet est présent puis, plus tard, un autre objet est placé à coté de lui de sorte qu'ils forment une seule composante connexe dans l'image. Nous souhaitons que les deux objets soient correctement segmentés.

2.2.2 La segmentation d'objets

La segmentation d'objets est un vaste sujet de vision par ordinateur et est un problème qui a été très largement étudié. Nous allons présenter ici quelques unes des grandes familles d'approches.

Temps					scénario	sortie désirée
0	∅	∅	∅	∅	scène vide	
1	A	B	C	D	valise AB**EF** posée mais partiellement occultée	
2	A	B	C	D		
3	A	B	E	F	valise ABEF entièrement visible	objet1 : AB
4	A	B	E	F	EF pas encore classé stationnaire	objet1 : AB
5	A	B	E	F		objet1 : ABEF

TABLE 2.2 – Exemple de problème de segmentation sur une image 1D. Bien que la partie EF apparaisse plus tard que la partie AB, on veut pouvoir leur affecter une même étiquette car il est possible qu'elle est été occultée par CD aux temps 1 et 2. (voir aussi cas table 2.3)

Temps					scénario	sortie désirée
0	∅	∅	∅	∅	scène vide	
1	A	B	∅	∅	objet AB posé avec du fond ∅ à coté	
2	A	B	∅	∅		
3	A	B	E	F	objet EF posé	objet1 : AB
4	A	B	E	F		objet1 : AB
5	A	B	E	F		objet1 : AB, objet2 : EF

TABLE 2.3 – Exemple de problème de segmentation sur une image 1D. Bien que AB et EF soient côte à côte dans l'image, on veut leur affecter deux étiquettes distinctes car aux temps 1 et 2 l'observation du fond indique que l'objet EF est différent de l'objet AB.

Les contours actifs introduits par Kass *et al.* [52] sont une approche de segmentation reposant sur des critères caractérisant la frontière de l'objet à segmenter. Ce type d'approche a été très largement repris et amélioré, et est désormais utilisés pour de très nombreuses applications [75]. A chaque contour est associé une énergie qu'il conviendra de minimiser et qui se décompose en une énergie interne et une énergie externe.

$$E = E_{interne} + E_{externe} \tag{2.1}$$

Le contour est constitué d'une suite de points, mobiles au cours des itérations successives, et qui à la convergence épousent le contour de l'objet à

segmenter. L'énergie interne correspond aux caractéristiques de l'objet que l'on souhaite segmenter, tels que par exemple longueur du contour ou courbure. L'énergie externe repose sur les caractéristiques de l'image, elle sert à quantifier combien le contour dans sa configuration actuelle correspond à l'image. Un critère est par exemple le gradient puisque l'on peut s'attendre à ce que les points du contour soient sur des gradients de l'image.

Cette méthode présente cependant certaines limitations. Tout d'abord elle est sensible à l'initialisation, qui doit être suffisamment proche du contour réel à segmenter, et elle converge facilement vers des minima locaux. Leur principale limitation est cependant qu'il est difficile de gérer les changements de topologie. C'est à dire qu'il faut connaître à l'avance le nombre de composantes connexes de l'objet à segmenter.

Les ensembles de niveau permettent de pallier la limitation topologique des contours actifs. Introduits par Dervieux et Thomasset [27], puis Osher *et al.* [83], ils consistent à minimiser une énergie par l'intermédiaire d'une fonction ϕ appelée ensemble de niveau. Dans le cas d'une segmentation binaire en deux classes a et b, Malladi *et al.* [70], cette fonction est telle que $\phi(x) > 0$ si $x \in a$ et $\phi(x) < 0$ si $x \in b$. Le contour de niveau 0 correspond ainsi à la frontière entre les deux classes. Pour trouver la frontière optimale, ϕ est généralement initialisée à un ensemble de petits rectangles régulièrement espacés dans l'image, puis ϕ est modifiée au cours des itérations en effectuant une descente de gradient. Zhu et Yuille [112] ont défini une énergie pour généraliser à une segmentation en N régions, où N est connu à l'avance. Brox et Weickert [21] proposent une méthode qui détermine automatiquement le nombre N de régions.

Les méthodes de type *split and merge* sont des approches où la segmentation est effectuée sur un critère de type région. Ohlander *et al.* [81] proposent une méthode qui part de l'image globale puis la divise en sous-régions. Les auteurs calculent l'histogramme de l'image et trouvent le seuil qui sépare le mieux les pics de l'histogramme. Le processus est alors répété récursivement sur les nouvelles sous-régions jusqu'à ce qu'elles soient suffisamment uniformes ou d'une taille suffisamment petite. Brice et Fennema [20] proposent quant à eux une méthode qui utilise une grille pour représenter les frontières entre les régions et fusionnent celles qui satisfont certains critères sur la taille de leur frontière commune ainsi que l'intensité des gradients à cet endroit.

Shi et Malik [93] proposent une technique intitulée coupe normalisée, et qui prend en compte les similarités entre pixels voisins et tend à séparer les pixels ayant de faibles affinités. L'image est donc représentée par un graphe

dont l'ensemble V des sommets représente les pixels de l'image, et les arêtes sont les relations de voisinage entre ces pixels. Le poids w_{ij} de chaque arête quantifie la similarité entre deux pixels voisins i et j. Une coupe entre deux ensembles de sommets A et B a donc un coût :

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij} \tag{2.2}$$

Chercher la coupe de poids minimale n'est cependant pas satisfaisant puisque par construction cela revient à privilégier les petits clusters, éventuellement ne contenant qu'un pixel isolé.

Ils définissent donc une autre mesure de la segmentation, la coupe normalisée :

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{2.3}$$

où $assoc(A, V) = \sum_{i \in A, j \in V} w_{ij} = assoc(A, A) + cut(A, B)$ la somme des similarités des pixels de A avec leur voisins. Calculer la coupe minimale normalisée est cependant un problème NP-Complet. Les auteurs proposent donc une méthode approximée qui consiste à trouver la coupe minimale dans le cas continu.

Une importante catégorie des algorithmes de segmentation binaire repose sur la définition et minimisation d'une énergie. Boykov et Funka-Lea [17] décrivent en détail plusieurs de ces techniques. L'une des approches couramment utilisée est l'approche par champ de Markov aléatoire, combinée à une recherche de coupe minimale pour minimiser l'énergie. Les champs de Markov aléatoires permettent de définir une énergie qui se décompose en une vraisemblance et un a priori dépendant de la relation qu'un pixel entretient avec ses voisins. Boykov et Jolly [19] furent les premiers à utiliser cette approche en segmentation d'image. Ils proposent que l'utilisateur sélectionne à la main quelques pixels du *fond* et du *premier plan* qui vont devenir des *graines*. Ces graines vont permettre d'estimer des statistiques qui vont ensuite, pour chaque pixel de l'image, définir une vraisemblance d'appartenance au *fond* et au *premier plan*. L'a priori utilisé dans leur segmentation est que deux pixels voisins qui ont des valeurs similaires appartiennent probablement à la même classe. Cette approche à été grandement reprise et enrichie en particulier par Rother *et al.* [89]. Alahari *et al.* [4] proposent quant à eux un algorithme efficace pour le cas de la segmentation multi-étiquettes.

Parmi ces méthodes, c'est l'approche par champs de Markov aléatoires qui nous a semblé être la plus adaptée et simple à mettre en oeuvre pour

intégrer notamment des informations temporelles. Elle est présentée plus en détails dans la section qui suit, avant que ne soit définie l'énergie utilisée pour notre problème particulier.

2.2.3 Les champs de Markov aléatoires

Les champs de Markov aléatoires sont utilisés en segmentation d'images pour prendre en compte des relations entre un pixel et ses voisins. Ils sont utilisés depuis plusieurs dizaines d'années déjà (Geman et Geman [38]). L'objectif est d'affecter une étiquette à chacun des pixels tout en considérant des relations entre les pixels voisins. Pour cela une image est représentée par un graphe dont les sommets sont les pixels de l'image et les arêtes représentent les relations de voisinage. Nous allons nous restreindre à l'optimisation de l'étiquetage par une approche de recherche de coupe minimale. Ce type d'approche est très étudié et permet de trouver rapidement une bonne approximation du minimum global ([57], [16]). Soit le graphe $G = (V, E)$ défini par l'ensemble de ses sommets V et l'ensemble des arêtes $E \subset V^2$, et L l'ensemble des étiquettes. Soit un étiquetage $x = (x_1, \ldots, x_n) \in L^{|V|}$. L'affectation des étiquettes se fait par la minimisation d'une énergie. Nous allons nous intéresser aux énergies de la forme de l'équation 2.4.

$$E(x) = \sum_{i \in V} D_i(x_i) + \lambda \sum_{(i,j) \in E} V_{ij}(x_i, x_j) \tag{2.4}$$

D_i représente une attache aux données. Il correspond à la vraisemblance de l'attribution de l'étiquette x_i au pixel i. $V_{i,j}$ est un terme de régularité faisant intervenir les sommets voisins. Il permet de modéliser un a priori sur la relation qu'un pixel entretient avec ses voisins, ce qui fait que tous les étiquetages ne sont pas équiprobables. Pour cette raison on parle de recherche de *Maximum A Posteriori* (ou MAP). On peut remarquer que prendre $\lambda = 0$, c'est à dire ne pas avoir d'a priori sur la distribution des étiquetages, permet de retrouver le *Maximum de Vraisemblance*.

On dit que V_{ij} est régulière ou sous-modulaire si et seulement si elle vérifie 2.5.

$$V_{ij}(a, a) + V_{ij}(b, b) < V_{ij}(a, b) + V_{ij}(b, a) \tag{2.5}$$

Seules les énergies dont le terme de régularisation est sous-modulaire peuvent être minimisées par l'utilisation des graphcuts, c'est à dire par la recherche de coupe minimale dans un graphe [57], [84].

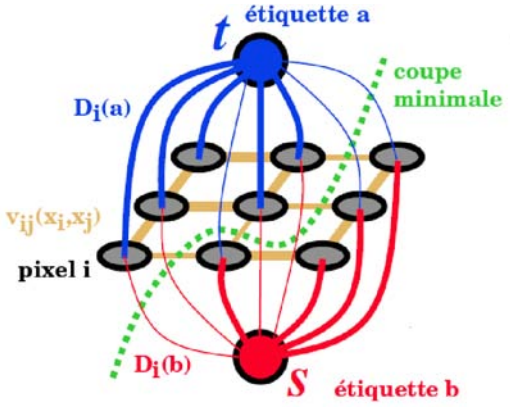


FIGURE 2.3 – Illustration du graphe utilisé pour une segmentation binaire. La coupe minimale constitue la frontière entre les deux étiquettes.

2.2.3.1 Cas de la segmentation binaire

Dans le cas où $|L = \{a, b\}| = 2$ il est possible de trouver un minimum global de l'énergie telle que définie par l'équation 2.4 en un temps polynomial ([39], [19]). Pour ce faire le problème de minimisation d'énergie est ramené à un problème de recherche de coupe minimale dans un graphe.

Comme illustré par la figure 2.3, deux sommets sont ajoutés S (*sink*) et T (*tank*) représentant chacune des deux étiquettes, respectivement b et a.

Soit $G_o = (V_o, E_o)$ le graphe original représentant l'image à segmenter. Le nouveau graphe G sur lequel la coupe minimale séparant S et T sera calculée est défini comme suit :

$$G = (V, E) \tag{2.6}$$
$$V = \{S, T\} \cup V_o \tag{2.7}$$
$$E = \{S\} \times V_o \cup V_o \times \{T\} \cup V_o \tag{2.8}$$

Les poids associés aux arêtes sont définis de la manière suivante :

$$w(e) = \begin{cases} D_i(a) & \text{si } e = (S, v_i) \in \{S\} \times V_o \\ D_i(b) & \text{si } e = (v_i, T) \in V_o \times \{T\} \\ V_{ij}(x_i, x_j) & \text{si } e = (x_i, x_j) \in E_o \end{cases} \tag{2.9}$$

Une fois la coupe minimale trouvée, les sommets de G qui se trouvent du même coté que le sommet S se verront affecter l'étiquette b. Réciproquement les sommets qui se trouvent du coté de T se verront affecter l'étiquette a.

Généralement la coupe minimale est trouvée en résolvant le problème dual : la recherche du flot maximal allant de S à T. On peut citer comme algorithme classique de recherche de flot maximal celui de Fort-Fulkerson [35].

2.2.3.2 Segmentation multi-étiquettes

Dans le cas multi-étiquettes la recherche d'un minimum global par graphcut est possible si il existe une relation d'ordre sur les étiquettes et que les termes V_{ij} vérifient des hypothèses de convexité.

Il existe néanmoins des heuristiques qui permettent de converger vers des minima locaux et se sont montrées très efficaces en pratique.

On rappelle qu'une fonction $d : X \times X \to \mathbb{R}$ est une semi-distance sur X si elle vérifie les équations 2.10, 2.11 et 2.12. Si d est une semi-distance et vérifie 2.13, alors c'est une distance.

$$d(x, y) \geq 0 \tag{2.10}$$
$$d(x, y) = 0 \Leftrightarrow x = y \tag{2.11}$$
$$d(x, y) = d(y, x) \tag{2.12}$$
$$d(x, z) \leq d(x, y) + d(y, z) \tag{2.13}$$

Boykov *et al.* [16] ont proposé deux algorithmes pour trouver rapidement une approximation du minimum global par graphcut. Si le terme de régularisation V_{ij} de l'énergie est une distance sur les étiquettes alors on peut trouver un minimum local de l'énergie en utilisant un algorithme d'α-expansion. Si ça n'est qu'une semi-distance on utilise un algorithme $\alpha\beta$-swap. Dans notre cas nous allons utiliser l'algorithme de Alahari *et al.* [3] dont une implémentation *C++* est disponible et qui propose une solution efficace pour accélérer l'algorithme de Boykov *et al.* .

2.2.3.3 Le mouvement d'α-expansion

Une α-expansion [19] consiste à autoriser un sommet du graphe soit à conserver son étiquette soit à prendre l'étiquette α. A chaque étape une étiquette α est choisie et on se ramène à un cas binaire en autorisant seulement deux étiquettes : α et $\bar{\alpha}$. Un nouveau graphe est construit de sorte que lors de la recherche de la coupe minimale un sommet étiqueté α conserve son étiquette α. C'est pour cela que l'on parle d'α-expansion. Un sommet étiqueté $\bar{\alpha}$ conservera son étiquette précédente.

L'algorithme d'α-expansion consiste donc à considérer toutes les étiquettes les unes à la suite des autres et à réaliser pour chacune d'elles le mouvement d'α-expansion optimal, jusqu'à ce qu'un état stable soit atteint.

L'algorithme d'α-expansion garantit une borne supérieure à l'énergie du minimum local trouvé. Cette borne est :

$$2 \times \max_{(p,q)\in E} \frac{\max_{\alpha\neq\beta} V_{p,q}(\alpha,\beta)}{\min_{\alpha\neq\beta} V_{p,q}(\alpha,\beta)} \times \arg\min_{x\in L^{|V|}} E(x) \tag{2.14}$$

Pour trouver l'α-expansion optimale d'un étiquetage x, Boykov *et al.* construisent un nouveau graphe avec une source α et un puits $\bar{\alpha}$ similaire au graphe original, mais avec des noeuds et arêtes supplémentaires dépendant de l'étiquetage initial x. Ils démontrent que calculer le flot maximal sur ce nouveau graphe est équivalent à trouver l'expansion optimale.

2.2.3.4 Algorithme Reduce, Reuse, Recyle

Pour notre application nous avons utilisé une implémentation de l'algorithme de Alahari *et al.* [3]. Cet algorithme est une combinaison de différentes méthodes. Il vise à donner une bonne approximation de l'étiquetage optimal de manière efficace, et peut se décomposer en trois parties. L'idée générale est de réutiliser au maximum les résultats intermédiaires calculés aux itérations précédentes de l'algorithme afin d'en accélérer la convergence (phase *reuse* et *recycle*). De plus les résultats de Kovtun [59] sont utilisés dans une phase d'initialisation *reduce* pour réduire la complexité du problème.

Reduce

Définition 1 *Un étiquetage $x \in (L \cup \{\epsilon\})^n$ (où ϵ est l'étiquette vide) est dit faiblement persistent s'il existe un minimum global x^* de l'énergie tel que $x_i \neq \epsilon \Rightarrow x_i = x_i^*$. Autrement dit si x est un étiquetage partiel d'un minimum global.*

Définition 2 *Un étiquetage $x \in (L \cup \{\epsilon\})^n$ est dit fortement persistent si pour tout minimum global x^* de l'énergie $x_i \neq \epsilon \Rightarrow x_i = x_i^*$. Autrement dit si x est un étiquetage partiel de tous les minima globaux.*

Le nombre de variables à optimiser va être réduit en s'inspirant de Kovtun [59] qui propose une solution pour obtenir des solutions non optimales pour des énergies non sous modulaires. Pour chaque étiquette $l_m \in L$ un problème auxiliaire P_m est construit de sorte que résoudre tous les sous-problèmes P_m nous permettent d'obtenir un étiquetage partiel fortement persistant.

Reuse Pour trouver les étiquettes les plus difficiles de façon efficace, celles a qui l'on a affecté l'étiquette vide lors de la phase de réduction de l'énergie (2.2.3.4), on va initialiser le dual de la première itération de l'algorithme d'expansion avec la solution partiellement optimale c'est à dire l'étiquetage partiel fortement persistant. Les étiquettes manquantes d'un sommet i sont fixées à $l_{i,min} = \arg\min_{l \in L} D_i(l)$.

Recycle A la première itération de l'algorithme d'expansion pour chaque étiquette, un graphe $G_i^1, i = 1 \dots |L|$ est construit. L'étiquetage optimal de l'expansion est trouvé en faisant une expansion classique. Aux itérations $u > 1$, G_i^u est trouvé en mettant à jour le graphe G_i^{u-1} [56]. L'idée est que plus les itérations (sur u) passent, plus les graphes G_i^u (i fixe) sont similaires et plus le calcul de mouvement optimal d'expansion est rapide si l'on se sert du flot optimal de l'itération précédente.

Les détails sur la mise à jour dynamique de l'étiquetage de graphes peuvent être trouvés dans [56].

2.2.4 Segmentation mono-caméra d'objets stationnaires

Nous avons présenté au chapitre précédent une méthode de détection d'objets stationnaires reposant sur le seuillage de l'âge des blocs du premier plan. Nous allons maintenant définir une énergie permettant de segmenter des objets stationnaires d'un même blob, comme expliqué en section 2.2.1.

2.2.4.1 Généralités

Le nombre d'étiquettes disponibles est fixé à l'avance, et l'une d'entre elles est réservée pour le fond et les objets non stationnaires. On a donc

$$L = \{l_{fond}, l_1, \dots, l_n\} \tag{2.15}$$

L'approche générale, sur laquelle l'énergie est définie, consiste à donner un coût nul à l'étiquette contenant le fond et les objets non stationnaires. Les autres étiquettes ont un coût par défaut qui est une constante positive, ce coût sera diminué en fonction de l'âge du descripteur observé. Un autre principe est que, lorsque l'on observe du premier plan, aucune étiquette ne peut a priori être privilégiée. En effet, l'énergie est définie au niveau du "bloc" et il est impossible à ce niveau là (avec notre approche) de déterminer si, à un instant donné, on observe réellement un objet statique, un objet en mouvement (occultant ou non un objet statique), ou encore un objet

stationnaire en occultant un autre. Nous verrons cependant que l'on peut trouver des critères permettant de considérer que certaines étiquettes ne sont pas compatibles avec un bloc.

2.2.4.2 Expression de l'énergie

Définition du terme d'attache aux données On rappelle ici que l'énergie est de la forme

$$E(x) = \sum_{i \in V} D_i(x_i) + \sum_{(i,j) \in E} V_{ij}(x_i, x_j) \tag{2.16}$$

où D_i est le terme d'attache aux données et V_{ij} est le terme de voisinage qui impose une régularité à la segmentation.

Le terme d'attache aux données permet de définir quand un bloc est considéré comme stationnaire. Pour chacun des blocs de l'image, un coût est affecté à chaque étiquette. Ce sera l'algorithme d'optimisation qui se chargera de trouver l'étiquetage optimal.

Pour chaque bloc i de l'image, un coût nul est affecté à l'étiquette représentant le fond et les objets en mouvement.

$$D_i(l_{fond}) = 0 \tag{2.17}$$

L'équation 2.17 nous permet de définir un coût par défaut pour étiqueter un bloc comme du fond. Étiqueter un bloc avec une étiquette objet coûte plus cher, à moins qu'il n'ait été observé un nombre suffisant de fois (équation 2.18). Nous définissons pour cela le coût des étiquettes "objet" de la façon suivante.

$$D_i(x_i \neq l_{fond}) = C - age_i(d_i) + penaliteTemps_i(x_i) + penaliteIncompatibilite_i(x_i) \tag{2.18}$$

Avec :

- C l'âge, en nombre d'images, nécessaire pour qu'un objet soit considéré stationnaire.
- age_i est l'age du descripteur d_i du pixel i. Cela signifie que ce descripteur a été observé pour la première fois il y a age_i images.
- $penaliteIncompatibilite_i(x_i) = max(t_{i,\emptyset} - t_{x_i} + C, 0)$ avec $t_{i,\emptyset}$ le temps où pour la dernière fois le fond a été observé au pixel i, et t_{x_i} est le temps où pour la première fois l'étiquette x_i a été affectée dans l'image. Ainsi ce terme représente la pénalité engendrée par le fait que au bloc i le fond a été observé postérieurement à la création de l'étiquette x_i.

– $penaliteTemps_i(x_i) = max(t_{x_i} - t_{d_i} + C, 0)$ avec t_{x_i} le temps où pour la première fois l'étiquette x_i a été affectée dans l'image et t_{d_i} le temps où pour la première fois le descripteur d_i a été observé au bloc i. Ce terme représente l'écart entre l'apparition de l'étiquette et la première observation de l'objet stationnaire.

Définition du terme de voisinage C'est le terme de régularisation qui va permettre d'homogénéiser l'étiquetage, et de séparer correctement deux objets stationnaires sur les zones de chevauchement. Soit $x \in L^{|V|}$.

$$V_{ij}(x_i, x_j) = \begin{cases} \lambda_1 + \lambda_2 e^{-|age_i - age_j|^2} \text{ si } x_i = x_j \\ 0 \text{ si } x_i \neq x_j \end{cases} \tag{2.19}$$

Avec cette expression du coût de voisinage il est plus coûteux de séparer deux blocs ayant des âges proches que des âges différents. Comme on peut le constater sur la figure 2.4, ceci permet d'obtenir la segmentation attendue d'un objet en occultant un autre. En effet le coût unaire ne favorise aucune des deux étiquettes, et au niveau de la zone de chevauchement elles sont toutes les deux compatibles (pas de pénalité). Comme on cherche à trouver une coupe minimale c'est la coupe la plus courte qui sera choisie. Le terme $\lambda_1 > 0$ permet de garantir l'existence du terme de régularisation. Le terme $\lambda_2 > 0$ permet une régularisation en fonction de l'age des blocs adjacents en augmentant le coût d'une coupe entre deux blocs ayant des ages similaires.

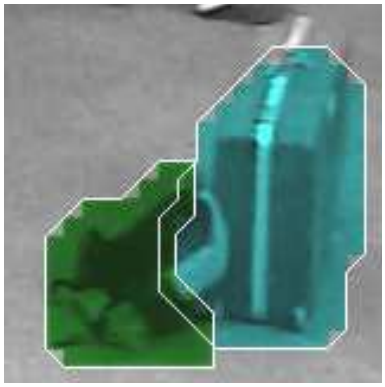
2.2.5 Évaluation de la segmentation mono-caméra

Nous présentons ici plusieurs résultats qualitatifs sur la segmentation mono-caméra des objets stationnaires.

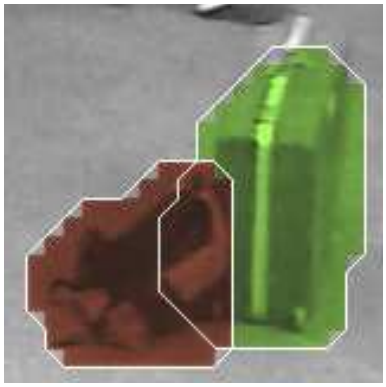
Les séquences sur lesquelles sont testées notre approche permettent de confirmer que nous pouvons traiter les cas introduits en section 2.2.1 et illustrés par les tables 2.1, 2.2 et 2.3. La table 2.4 montre quelles sont les figures qui correspondent aux différents cas. Les expériences montrent que notre approche permet effectivement de les traiter, cependant quelques erreurs de sur-segmentation peuvent apparaître comme c'est le cas dans les figures 2.5 et 2.10.

Nous allons maintenant analyser plus en détail les résultats présentés sur ces figures.

La figure 2.5 montre que nous sommes capables de gérer les situations décrites en section 2.2.1. On y observe, sur chacune des lignes, la détection d'une



(a) Terme binaire avec $\lambda_2 = 0$. La segmentation est arbitraire sur la zone de chevauchement des objets.



(b) Terme binaire avec $\lambda_2 > 0$. La segmentation prend en compte la similarité des âges.

FIGURE 2.4 – Cas de deux objets s'occultant partiellement et apparus à des instants différents. Il est donc possible de les segmenter avec l'algorithme que nous avons introduit.

Cas 1 Table 2.1 Apparition sous occultation	Cas 2 Table 2.2 Occultation par objet mobile	Cas 3 Table 2.3 Occultation par objet stationnaire
Figure 2.5, 2.6, 2.7, 2.10, 2.12	Figure 2.5, 2.6, 2.7, 2.10, 2.12	Figure 2.5, 2.8, 2.9, 2.11, 2.12

TABLE 2.4 – Index des figures illustrant les trois cas introduits en section 2.2.1.

valise qui apparaît sous occultation partielle, puis, sur la deuxième ligne, la segmentation de deux objets adjacents apparus à deux instants différents. Les deux objets sont correctement segmentés, mais on peut remarquer que la valise se voit affecter deux étiquettes. Cela est dû au fait que les deux composantes ne sont pas connexes et comme aucune étiquette n'est incompatible il est possible que deux étiquettes distinctes soient affectées.

Les figures 2.6 et 2.7 montrent un objet posé sous occultation partielle et filmé de deux points de vues différents. Dans les deux cas on observe bien qu'un unique objet est détecté, malgré les occultations par une personne mobile.

La figure 2.8 montre une limitation de notre segmentation. Un premier objet est amené dans la scène et est correctement détecté. Un second objet est adossé au premier et est bien détecté stationnaire. Cependant ce second

FIGURE 2.5 – Première ligne : détection d'un objet stationnaire qui apparaît sous occultation partielle. Deuxième ligne : détection de deux objets stationnaires adjacents.

objet glisse puis redevient stationnaire et est de nouveau détecté. Il est donc détecté deux fois puisque la disparition de la première détection n'est pas observée car l'objet est toujours présent. Les mêmes résultats filmés sous un second point de vue sont visibles en figure 2.9.

La figure 2.10 montre un cas où une erreur de ré-identification d'une partie d'un objet stationnaire avec un objet mobile engendre une sur-segmentation. Une personne pose une valise à l'instant t_{pose}, l'occulte partiellement et s'en va. Une région de la valise est cependant ré-identifiée par erreur avec la jambe de la personne présente à $t_{pose} - 1$ dans la scène, c'est à dire avant que la valise ne soit posée. A cet instant on observe le fond là où sera plus tard détectée l'ombre de la valise. La pénalité d'incompatibilité est donc active et deux étiquettes sont attribuées : une pour la partie ré-identifiée par erreur et les zones adjacentes où du fond n'était pas observé à $t_{pose} - 1$, et une pour les zones où le fond était visible à $t_{pose} - 1$ (l'ombre).

Dans la figure 2.11, un journal est détecté, puis une chaise est posée et l'occulte avant qu'il ne soit enlevé. La partie du journal occultée par la chaise engendre donc une fausse alarme. Cela est un effet attendu de notre algorithme car nous considérons que comme la disparition du journal n'est pas constatée (il est partiellement occulté), c'est qu'il est peut être encore présent. Dans la suite de cette séquence la chaise est détectée puis un sac est

FIGURE 2.6 – Détection d'un objet stationnaire qui apparaît sous occultation partielle. La figure 2.7 montre la même scène filmée sous un autre point de vue.

FIGURE 2.7 – Détection d'un objet stationnaire qui apparaît sous occultation partielle. La figure 2.6 montre la même scène filmée sous un autre point de vue.

posé dessus. Bien qu'une partie du sac soit détectée plus tardivement que le reste, le terme de voisinage permet bien d'avoir une segmentation correcte des deux objets.

Finalement la figure 2.12 montre le cas idéal où une valise est déposée sous occultation partielle et est bien détectée. Deux autres objets sont ensuite ajoutés et, bien qu'ils recouvrent partiellement la valise, chacun des objets est correctement segmenté.

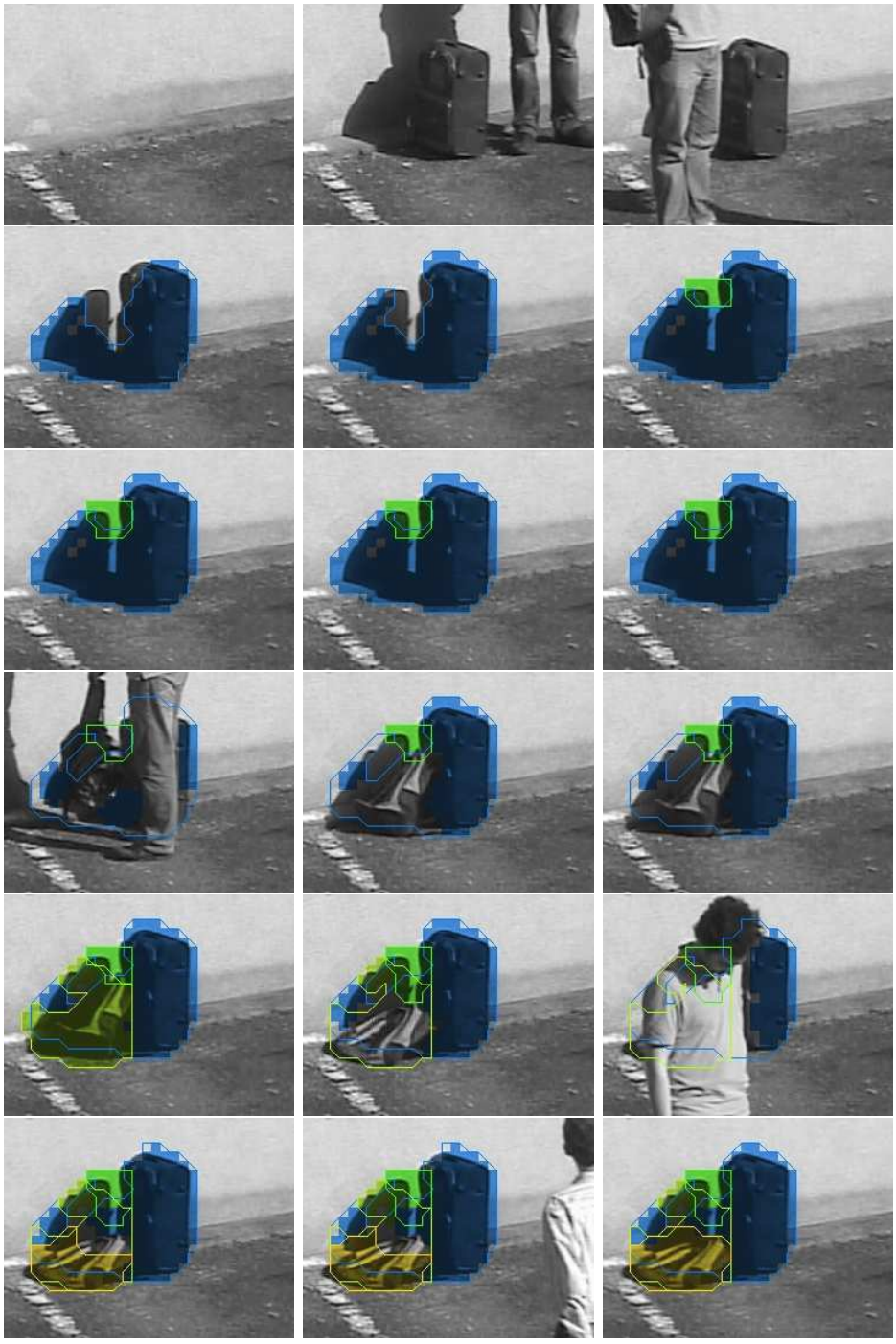


FIGURE 2.8 – Une valise et son ombre sont détectées stationnaires bien qu'elles apparaissent partiellement occultées. Un sac est ajouté à coté de la valise et est détecté stationnaire. Il s'affaisse alors et est de nouveau détecté stationnaire. Ceci conduit à une sur-segmentation. Un second point de vue est donné en figure 2.9.

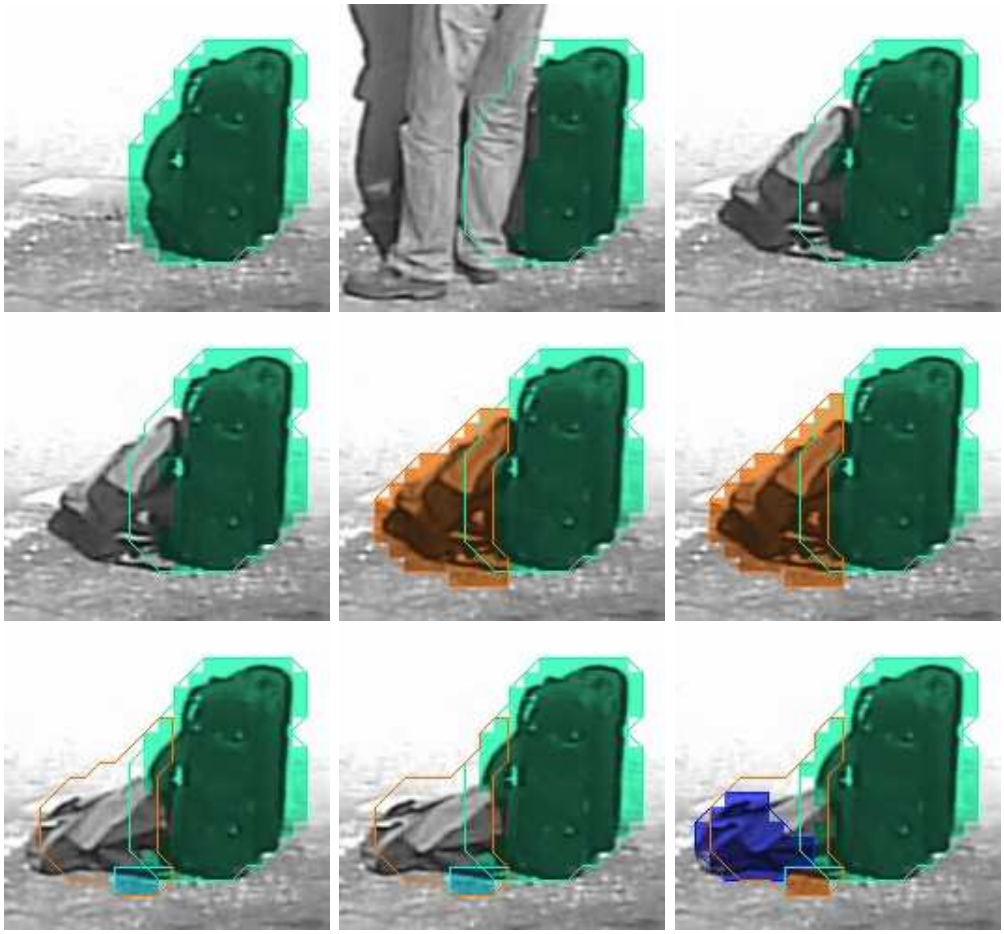


FIGURE 2.9 – Une valise et son ombre sont détectées stationnaires bien qu'elles apparaissent partiellement occultées. Un sac est ajouté à coté de la valise et est détecté stationnaire. Il s'affaisse alors et est de nouveau détecté stationnaire. Ceci conduit à une sur-segmentation. Un second point de vue est donné en figure 2.8.

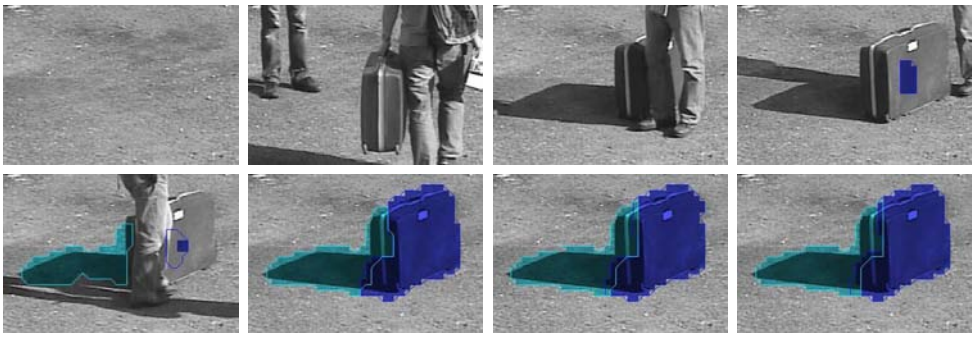


FIGURE 2.10 – Une partie de la valise (image 4) est réidentifiée avec la jambe (image 2) et est en conséquence détectée stationnaire trop tôt. Comme sur l'image 2 le fond était observée là où il y a de l'ombre sur l'image 4, la pénalité d'incompatibilité est active et impose l'utilisation de deux étiquettes distinctes.

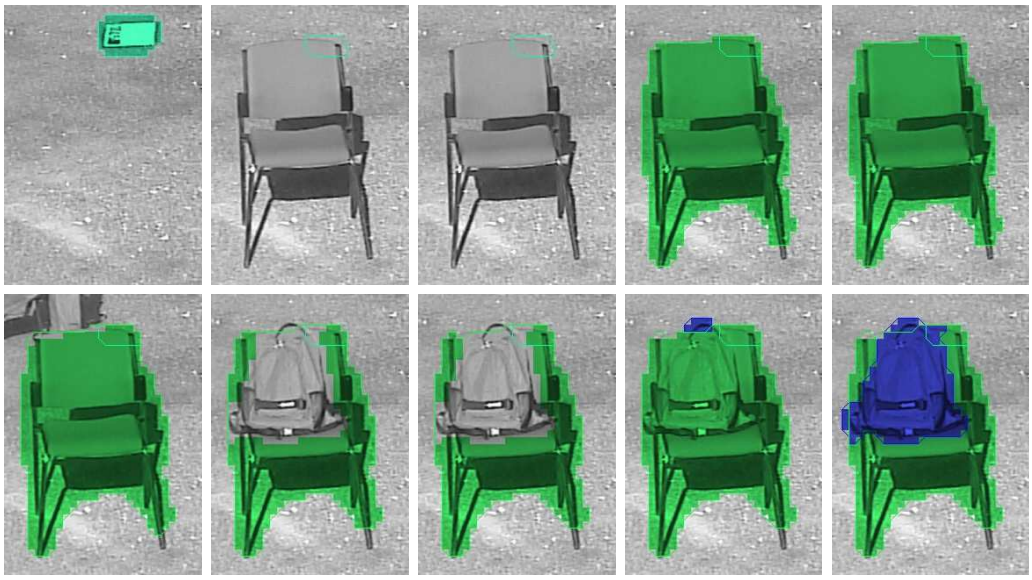


FIGURE 2.11 – Illustration d'un cas où une fausse alarme est due à l'occultation d'un objet (ici le journal) par un autre (la chaise) : une fois le journal enlevée la partie de la silhouette occultée par la chaise reste, car le fond n'est pas observé à cet endroit. Un troisième objet (le sac) est alors ajouté sur la chaise et est correctement segmenté.

FIGURE 2.12 – Cas idéal de segmentation de trois objets. On peut remarquer que la valise est déposée sous occultation partielle.

2.2.6 Conclusion sur la segmentation monocaméra d'objets stationnaires

Nous avons présenté une méthode qui permet de segmenter certains objets stationnaires contenus dans une unique silhouette. Un critère d'incompatibilité permet de conserver l'unité des objets apparaissant sous occultation partielle tout en séparant ceux qui sont posés à des instants différents. Notre détection n'est donc pas un simple masque binaire contenant tout les objets stationnaires, mais un ensemble de masques pour les objets apparus à différents instants.

2.3 Mise en correspondance dans une paire de caméras

Dans cette partie on suppose les caméras calibrées intrinsèquement et extrinsèquement et dans une configuration stéréo. Contrairement à la configuration stéréo classique, où l'écart entre les deux caméras n'est pas trop important par rapport à la distance de la scène observée, l'écart entre les deux caméras est suffisamment grand pour empêcher un appariement stéréo reposant sur des critères d'apparence. Une telle disposition des caméras a notamment pour avantage d'augmenter les chances d'observer les objets de la scène en cas d'occultation.

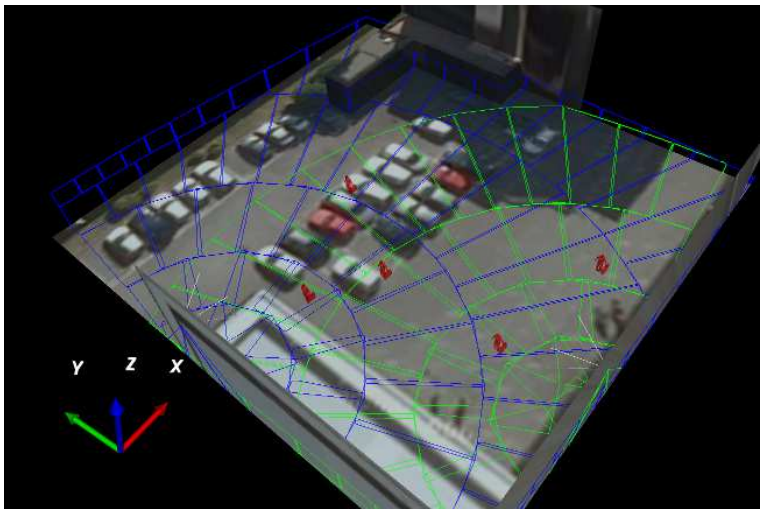


FIGURE 2.13 – Exemple de tour de garde de deux caméras PTZ surveillant un parking.

2.3.1 État de l'art

Nous présentons ici quelques méthodes de la littérature permettant la mise en correspondance des observations de plusieurs caméras ayant un champ de vue commun afin de retrouver le nombre et la position d'objets ou de personnes. Ces méthodes ont été développées pour des applications de détection d'objet stationnaire, mais aussi de suivi de personnes. Elles peuvent être classées en deux catégories. Il y a d'une part les méthodes directes qui partent des observations des caméras pour en inférer la position des objets, et d'autre part les méthodes inverses qui émettent des hypothèses sur les positions des objets puis sélectionnent les meilleures par un processus d'optimisation.

Beynon *et al.* [12], dans un contexte de tracking et détection d'objets abandonnés, font l'hypothèse que le monde est plan et que tous les objets du premier plan sont en contact avec le sol. De cette façon les coordonnées d'un blob dans le monde 3D sont simplement retrouvées en projetant le centre de l'arête basse de la boite englobante du blob. Le principe de leur méthode est ensuite d'utiliser un algorithme de résolution de problèmes linéaires d'affectation pour associer simultanément et de manière optimale tous les blobs observés aux objets réels. Pour ce faire, un coût d'association entre une observation et un objet est défini par l'équation 2.20.

$$cost = w_{pos}cost_{pos} + w_{size}cost_{size} + w_{color}cost_{color} \tag{2.20}$$

Ce coût repose sur trois critères qui sont la position dans le monde 3D, la taille dans chaque caméra et la couleur dans chaque caméra. Ainsi seule la position est un critère partagé par l'ensemble des caméras.

Chaque itération de l'algorithme d'association peut résulter en la création d'un nouvel objet du monde 3D, la mise à jour d'un objet existant, ou encore pas de mise à jour pour un objet. Un nouvel objet est créé si le coût d'association d'une observation à tous les objets existant est supérieur à un certain seuil. Un objet est supprimé s'il n'a pas été observé depuis un certain temps.

Il y aura cependant toujours des situations pour lesquelles l'étape d'association fera des erreurs. Les auteurs proposent donc après la phase d'association une phase de correction des erreurs. Ces problèmes sont de deux types :

1. Un objet réel peut être représenté par deux objets.
2. Lorsque un objet en occulte un autre, il peut n'y avoir qu'une seule observation pour deux objets réels.

Pour le cas 1 où un objet réel est représenté par deux objets, les auteurs proposent de fusionner les objets sous certaines conditions. Si les deux objets ont été observés dans une même caméra alors ils ne sont pas fusionnés. Par contre si leurs observations sont toutes récemment disjointes sur l'ensemble des caméras alors ils sont fusionnés.

Pour le cas 2 où deux objets réels forment une seule observation, les auteurs n'essayent pas de segmenter les objets. L'occultation est détectée si la reprojection des objets dans une caméra se fait dans un unique blob. Dans ce cas afin de ne pas faire une mise à jour erronée les objets sont marqués comme occultés et ne sont pas mis à jour.

Miezianko *et al.* [73], pour apparier et localiser les objets détectés dans un réseau de caméras, supposent eux aussi que le monde est plan. Ils initialisent une image vide représentant le plan du sol et qui va servir à compter le nombre d'observations de chaque pixel. Ils projettent simplement sur cette image les silhouettes détectées et incrémentent la valeur des pixels concernés. Finalement les objets sont les maxima locaux d'accumulation des observations projetées sur le plan du sol.

Utasi *et al.* [100] [101] proposent une méthode pour détecter et localiser efficacement les piétons dans un réseau de caméras à champ joint. Leur approche doit pouvoir gérer les occultations importantes. Pour cela ils utilisent les masques binaires de soustraction de fond de chaque caméra et procèdent en trois étapes principales. Tout d'abord les silhouettes sont projetées sur plusieurs plans parallèles au plan du sol à différentes altitudes (figure 2.14), puis des caractéristiques donnant des indications sur la taille et la localisation des personnes sont extraites, enfin un algorithme d'échantillonnage est utilisé pour trouver la configuration optimale.

Pour une position p dans le plan 3D, et h la taille supposée d'une personne on définit pour la caméra i deux mesures illustrées figure 2.14 et caractéristiques de la présence d'une personne. La mesure f_0^i mesure la présence d'un objet en contact avec le plan du sol au dessus du sol, et f_z^i mesure la présence d'un objet sous le plan d'altitude z mais en contact avec celui-ci. Ces deux mesures combinées sont donc un indicateur de la présence d'un objet de taille z sur le plan du sol.

$$f_0^i(p) = \frac{|A_0^i \bigcap S^i(p)| - \alpha |A_0^i \bigcap \bar{S}^i(p)|}{|\bar{S}^i(p)|} \ , \tag{2.21}$$

$$f_z^i(p) = \frac{|A_z^i \bigcap S^i(p)| - \alpha |A_z^i \bigcap \bar{S}^i(p)|}{|S^i(p)|} \ , \tag{2.22}$$

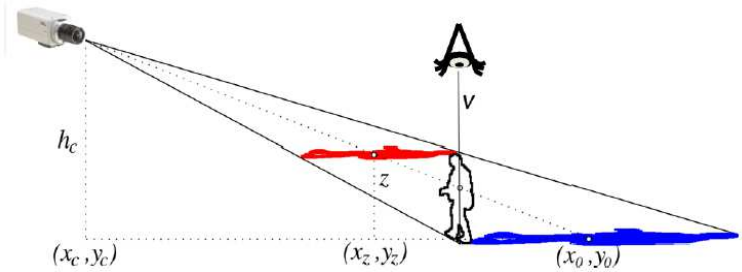


FIGURE 2.14 – La projection sur des plans parallèles au plan du sol permet de retrouver la taille d'une personne.

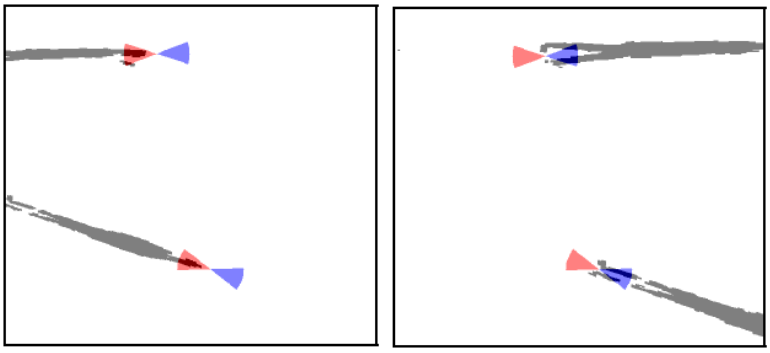


FIGURE 2.15 – Utasi *et al.* [100] [101] : extraction des caractéristiques. Gauche : au niveau de la tête. Droite : sur le plan du sol. En bleu : secteur angulaire $S^i(p)$. En rouge : secteur angulaire $\bar{S}^i(p)$.

où A_z^i est la projection du masque du premier plan de la caméra i sur le plan d'altitude z, $S^i(p)$ est une section angulaire de sommet p de largeur et rayon fixés a priori et de direction opposée à la caméra. $\bar{S}^i(p)$ est dans la direction opposée à $S^i(p)$. Ces caractéristiques sont illustrées en figure 2.15.

Si les mesures f_0^i et f_z^i sont des caractéristiques faibles, elles peuvent cependant être utilisées pour construire une caractéristique robuste, si l'on considère simultanément les observations provenant de toutes les caméras. On construit alors en équation 2.23 la caractéristique $f(p,z)$ mesurant la présence d'un objet de taille z à la position p reposant sur le plan du sol.

$$f(p,z) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} f_0^i(p) \times \frac{1}{N}\sum_{i=1}^{N} f_z^i(p)} \tag{2.23}$$

Le nombre et la configuration optimale des personnes dans la scène sont calculés en minimisant une énergie. Elle est de la forme :

$$\Phi(\omega) = \sum_{u \in \omega} J(u) + \gamma \sum_{(u,v) \in \omega^2} I(u,v) \ . \tag{2.24}$$

où ω est une configuration (*ie.* ensemble de personnes), I permet de modéliser les interactions entre les personnes et d'éviter les recouvrements trop importants, J permet de caractériser la vraisemblance d'un objet par rapport aux observations. L'optimisation est effectuée en utilisant la méthode d'échantillonnage de Descombes *et al.* [28] : *Multiple Birth and Death Dynamics*. Cet algorithme a l'avantage d'être plus rapide que d'autres techniques couramment utilisées, telle que les RJMCMC.

Fleuret *et al.* [34] proposent de même que Utasi *et al.* une méthode inverse. Ils discrétisent le plan du sol en une grille régulière. En chacune des positions de la grille un rectangle modélisant la silhouette d'un piéton est projetée sur les différentes caméras. Une carte des probabilités d'occupation est alors calculée et un algorithme de programmation dynamique permet de calculer les positions des personnes.

Toujours pour le suivi de personnes, Khan *et al.* [53] proposent de projeter sur le plan du sol les silhouettes détectées. Ils calculent une contrainte d'occupation homographique sur le plan du sol, qui leur permet de fusionner l'information provenant des différentes caméras. Cette contrainte traduit le fait que seul les pixels du premier plan qui sont réellement sur le plan du sol sont garantis d'être reprojetés sur des pixels du premier plan pour toutes les autres caméras. Afin de pouvoir gérer les cas où les personnes courent ou sautent, les auteurs considèrent plusieurs plans parallèles au plan du sol et à des altitudes différentes.

Toutes ces méthodes de l'état de l'art supposent que la scène se résume à un monde planaire. Cette hypothèse permet de simplifier l'approche directe, ou de grandement diminuer l'espace de recherche des méthodes inverses. Nous proposons une approche reposant sur la mise en correspondance des silhouettes des objets observés et qui n'impose aucune contrainte sur le monde 3D.

2.3.2 Rappels de stéréovision

L'approche que nous allons proposer consiste à associer des ensembles de silhouettes correspondant à un même objet entre les deux caméras. Le critère de mise en correspondance repose sur des contraintes géométriques liées à la paire de caméras. Pour cette raison, nous présentons ici quelques rappels de stéréovision.

On suppose que l'on dispose de deux caméras C_g et C_d étalonnées et de centres optiques respectifs O_g et O_d. L'objectif est de retrouver la position 3D de points à partir de leurs observations par la paire de caméras. Il y a pour cela deux étapes clefs :

1. Mise en correspondance de points de chaque image.
2. Triangulation des points.

Nous rappelons les termes suivants :
- **épipole** : projection du centre optique d'une caméra sur le plan image de l'autre caméra (ici e_g et e_d).
- **plan épipolaire** : plan contenant les centres optiques O_g et O_d.
- **droite épipolaire** : intersection d'un plan épipolaire avec le plan image d'une caméra.

Comme illustré par la figure 2.16, à chaque pixel d'une image est associée une droite épipolaire dans l'autre image. Cette droite forme une contrainte pour l'appariement des points en restreignant l'espace de recherche des pixels candidats. Dans les situations où les deux caméras ont à peu près le même point de vue de la scène, un critère d'apparence peut par exemple être utilisé pour sélectionner parmi les points de la droite épipolaire celui qui est l'image du même point monde.

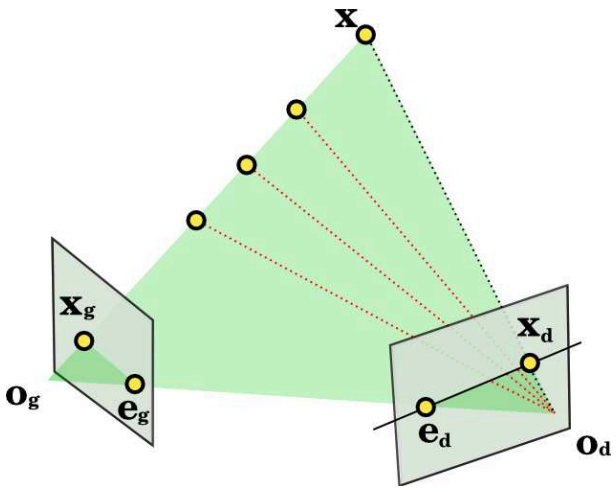


FIGURE 2.16 – Les candidats à l'appariement du point x_g de la caméra gauche sont situé sur une droite épipolaire. Cette droite est la projection de la droite $(O_g x_g)$ dans la seconde caméra. On remarquera que les droites épipolaires d'une caméra sont concourantes. Leur point d'intersection, appelé épipole, est le point d'intersection de la droite $(O_g O_d)$ avec le plan image de la caméra.

Si l'on connait une paire de pixels (x_g, x_d) de deux caméras qui sont l'image d'un même point du monde x alors on peut retrouver les coordonnées de x par une triangulation. En effet x est le point d'intersection des droites $(O_g x_g)$ et $(O_d x_d)$. Cependant en pratique, comme illustré en figure 2.17 les deux droites ne se croisent pas réellement. Nous allons alors chercher à minimiser la distance D entre les deux droites.

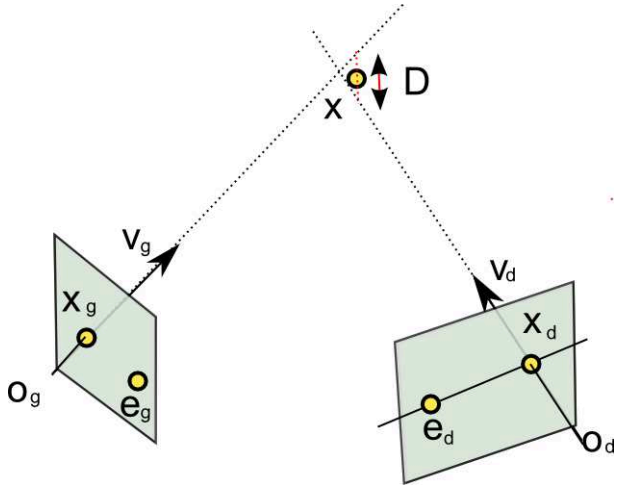


FIGURE 2.17 – Triangulation de deux droites non sécantes.

Notons δ le vecteur dont on va chercher à minimiser la norme.

$$\delta = O_g + \lambda_g v_g - (O_d + \lambda_d v_d) \tag{2.25}$$

Notre problème revient alors à chercher

$$(\hat{\lambda_g}, \hat{\lambda_d}) = \underset{\lambda_g, \lambda_d}{\arg\min} \|\delta\| \tag{2.26}$$

Or on sait que le δ de norme minimale est orthogonal à v_g et v_d. On a donc :

$$\begin{cases} \delta \cdot v_g = 0 \\ \delta \cdot v_d = 0 \end{cases} \tag{2.27}$$

$$\begin{cases} (O_g - O_d + \lambda_g v_g - \lambda_d v_d) \cdot v_g = 0 \\ (O_g - O_d + \lambda_g v_g - \lambda_d v_d) \cdot v_d = 0 \end{cases} \tag{2.28}$$

Ce qui nous permet de trouver λ_g et λ_d en résolvant un système linéaire. En effet on obtient finalement :

$$\begin{pmatrix} \|v_g\|^2 & -v_d \cdot v_g \\ -v_d \cdot v_g & \|v_d\|^2 \end{pmatrix} \begin{pmatrix} \lambda_g \\ \lambda_d \end{pmatrix} = \begin{pmatrix} (O_g - O_d) \cdot v_g \\ (O_d - O_g) \cdot v_d \end{pmatrix}. \tag{2.29}$$

Enfin le point P d'intersection estimé des deux droites est

$$P = \frac{(O_g + \lambda_g v_g) + (O_d + \lambda_d v_d)}{2} \tag{2.30}$$

Pour notre application nous traitons des blobs, ou silhouettes, plutôt que des points. Comme nous ne voulons pas imposer que les objets soient vus avec des angles de vues similaires, nous ne pouvons pas nous reposer sur des critères d'apparence pour la phase de mise en correspondance. Cependant pour tout objet il existe au moins deux points visibles à la fois dans les deux caméras [23]. En effet, en particulier si l'objet est concave, les points d'un objet pour lesquels les plans épipolaires associés sont tangents à la surface de l'objet ne sont pas victime d'auto-occultation par l'objet, comme illustré en figure 2.18.

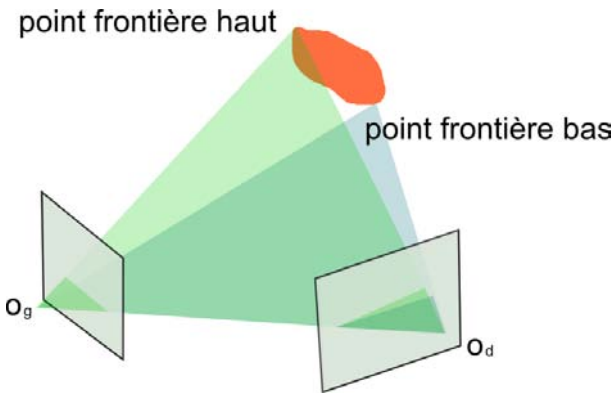


FIGURE 2.18 – Illustration des points frontières haut et bas d'un objet vu par une paire de caméras.

En pratique, avec des images rectifiées dont les droites épipolaires sont horizontales, les points haut et bas (dans l'image) d'un objet sont des points frontières. Ce sont donc ces points frontières haut et bas que nous utiliserons par la suite dans un critère d'appariement. Ces points facilement identifiables peuvent directement être mis en correspondance d'une caméra à l'autre si les silhouettes sont appariées, et ceci sans avoir recours à un critère d'apparence.

2.3.3 Mise en correspondance par recherche de couverture en cycles de coût minimal

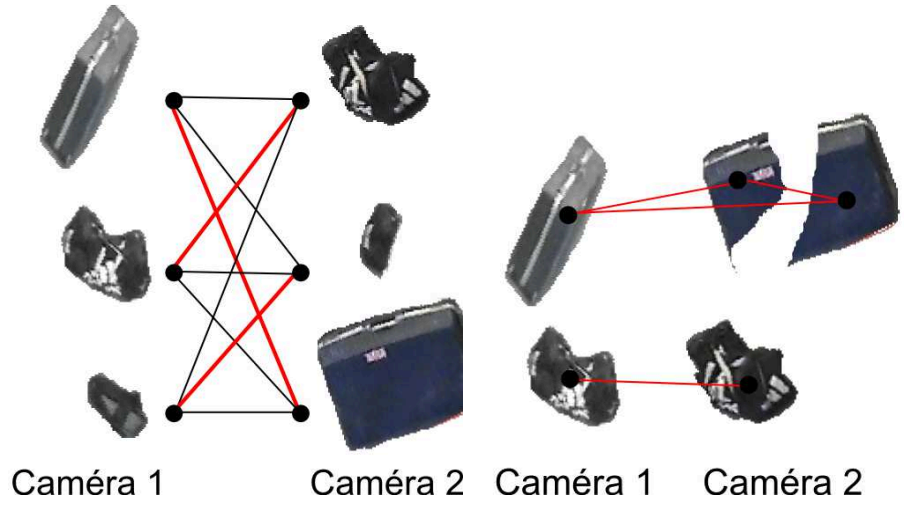
Nous partons de l'hypothèse que notre paire de caméras est entièrement calibrée et que les objets stationnaires sont détectés et segmentés comme présenté en section 2.2.4. On suppose que les images sont rectifiées de sorte que les droites épipolaires sont horizontales.

Si nous avions à notre disposition une segmentation parfaite de tous les objets stationnaires il serait possible de trouver les meilleures associations entre les silhouettes provenant des deux caméras en construisant un graphe biparti (figure 2.19(a)). Le coût de chaque arc pourrait être simplement la somme des angles entre les plans épipolaires des points frontières haut de chaque silhouettes et des plans épipolaires des points frontières bas de chaque silhouette. L'intérêt d'une telle modélisation est que le problème de recherche de couplage optimal dans un graphe biparti peut être résolu en temps polynomial [60]. Cependant, dans les cas réels, un objet peut être mal segmenté et apparaitre comme un ensemble de plusieurs silhouettes au sein d'une même caméra (figure 2.19(b)). Pour cette raison le modèle de graphe biparti n'est pas suffisant pour représenter notre problème.

Faire directement des associations de silhouettes n'est pas suffisant nous proposons d'associer des points frontières. Le principe de base est d'associer les points frontières haut (respectivement bas) d'une caméra aux points frontières haut (respectivement bas) de la seconde caméra. Le coût d'une association entre deux points frontières est l'angle entre les plans épipolaires qui les supportent. Dans le cas idéal d'une bonne association les deux points frontières appartiennent au même plan épipolaire et le coût est donc nul. Dans le cas d'une mauvaise association où deux points frontières appartiendraient à des plans épipolaires différents, le coût d'association serait strictement positif.

Afin de représenter correctement les objets nous définissons un graphe orienté pondéré tel qu'illustré en figure 2.20. Ce graphe présente quatre types d'arcs dont nous allons détailler les poids et significations. Les cycles de ce graphe représentent les différentes associations possibles de points frontières. A chaque cycle on peut associer un coût, celui de la somme des poids des arcs qui le constitue. Notre problème d'appariement revient alors à un problème de partition de coût minimal du graphe en cycles disjoints.

Dans la suite considérons deux silhouettes s_1 et s_2 ainsi que leur deux points frontières sortants o_1, o_2 et entrants i_1 et i_2. Nous noterons $|.|$ la norme dans l'image rectifiée de la projection d'un point sur l'axe vertical. De



(a) Cas où la segmentation des objets est parfaite. Le problème peut être représenté avec un graphe biparti. Chaque arête représente une possible association de silhouettes. Si les arêtes sont pondérées on peut chercher le couplage de coût minimal en temps polynomial [60].

(b) Cas réel avec segmentation imparfaite. Un objet du monde peut être représenté par plusieurs silhouettes dans une caméra.

FIGURE 2.19 – Illustration du problème d'appariement de silhouettes observées dans une paire de caméras. Les arcs rouges sont les associations désirées.

cette façon, à titre d'exemple, $|o_1 - o_2|$ est aussi la valeur absolue de l'angle entre les plans épipolaires supportant les point o_1 et o_2. Dans nos images rectifiées cela peut être assimilé à une différence de hauteur dans l'image.

Les arcs d'*association* relient deux points frontières hauts (respectivement bas) de deux caméras. Le coût associé est défini naturellement comme l'angle entre les plans épipolaires portant chaque point. Un coût positif peut signifier que l'association est mauvaise ou encore qu'une silhouette est partiellement occultée dans l'une des caméras (voir figure 2.21). Le coût d'association de s_1 vers s_2 est :

$$c_{association} = |o_i - i_j|. \tag{2.31}$$

Une contrainte sur les dimensions de la scène réelle peut être ajoutée lors de la définition des coûts d'association. En effet si deux points frontières sont appariés, il est possible de les trianguler pour retrouver le point 3D duquel ils sont l'image. Si l'appariement est mauvais il est possible que le point 3D reconstruit soit à une position aberrante, par exemple sous le niveau du sol,

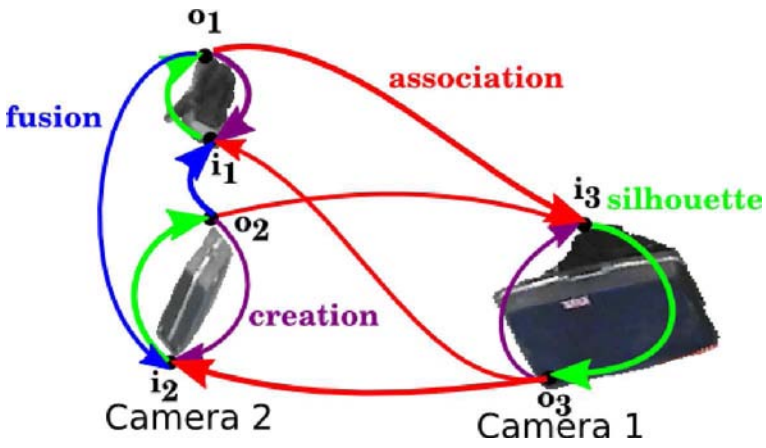


FIGURE 2.20 – Graphe orienté complet montrant les quatre types d'arcs autorisés. Les associations possibles de points frontières forment des cycles dans le graphe. Chaque cycle correspond alors à une association de silhouettes. L'orientation du graphe définit des points entrants et sortant des silhouettes, notés i et o. On peut remarquer que suivant la caméra que l'on considère les points entrants sont les points hauts ou bas des silhouettes.

ou encore à une altitude improbable. Ainsi il est possible de contraindre les associations des points frontières en donnant un coût infini aux arcs d'association de points frontières bas dont le point 3D reconstruit serait en dessous d'une certaine altitude, et de même pour les arcs d'association de points frontières hauts dont les points 3D reconstruits serait au dessus une altitude fixée à l'avance. Ce seuillage va permettre de limiter certaines ambiguïtés d'appariement en filtrant des mauvaises associations de points se trouvant sur un même plan épipolaire mais appartenant à des objets différents.

Les arcs de *silhouette* relient le point frontière entrant d'une silhouette au point frontière sortant. Ils ont un coût nul, leur rôle est de préserver l'unité de l'objet.

$$c_{silhouette} = 0 \tag{2.32}$$

Les arcs de *création* relient le point frontière sortant d'une silhouette au point frontière entrant. Ils permettent d'autoriser le fait qu'une silhouette ne soit associée à aucune autre. Ce cas peut arriver si la silhouette est une fausse alarme qui ne serait détectée que dans une caméra, ou encore si l'objet réel auquel elle correspond n'a été détecté que dans une caméra. Cela revient à considérer que l'objet est complètement occulté dans la seconde caméra. Par conséquent son coût est :

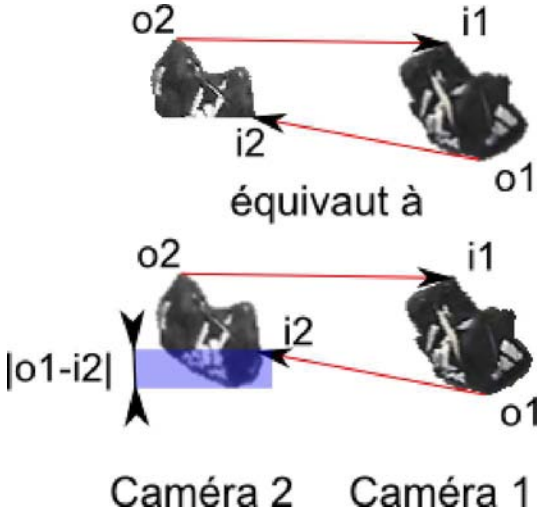


FIGURE 2.21 – Illustration du coût d'association. Un coût non nul peut être vu comme révélateur d'une occultation partielle.

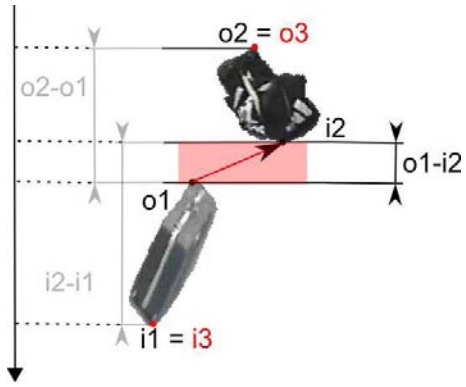
$$c_{creation} = |o_i - i_i|. \tag{2.33}$$

Les arcs de *fusion* relient le point frontière sortant au point frontière entrant de deux silhouettes d'une même caméra. Ils permettent d'expliquer les parties de silhouettes en trop (figure 2.22(b)) ou manquantes (figure 2.22(a)). De plus afin de prévenir la fusion de deux silhouettes trop éloignées leur coût intègre la distance angulaire entre les deux silhouettes selon l'axe horizontal. En effet les coûts des arcs reposent principalement sur des angles entre plans épipolaires, ce qui fait qu'ils ne dépendent pas de la position des points dans ces plans. Prendre en compte cet angle permet de ne pas fusionner des silhouettes reposant sur le même plan épipolaire mais qui sont trop éloignées. Enfin les arcs de fusion entre deux silhouettes qui se sont vues affecter deux étiquettes différentes ont eux aussi un coût infini. De cette façon si des silhouettes sont détectées comme incompatibles lors de l'étape de segmentation mono-caméra elles ne pourront pas être fusionnées à l'étape d'appariement. Finalement le coût d'un arc de fusion est :

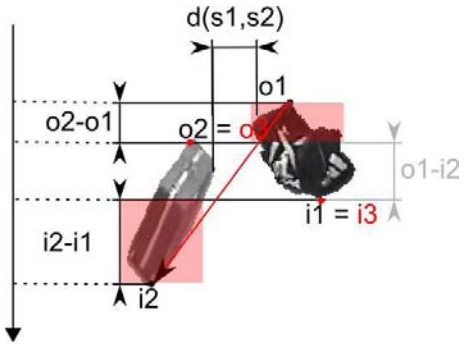
$$c_{fusion} = (o_2 - o_1)^+ + (i_2 - i_1)^+ + (o_1 - i_2)^+ + d(s_1, s_2) \tag{2.34}$$

avec $(.)^+ = \max(0, .)$ et $d(s_1, s_2)$ est la distance entre s_1 et s_2 dans l'image selon l'axe horizontal. Pour une paire de silhouettes il y a deux arcs de fusion possibles. On peut remarquer que le coût des arcs n'est pas symétrique,

ceci est visible figure 2.22. Les deux coûts sont en fait en quelque sorte complémentaires de sorte que si l'un est faible alors l'autre est important. Cela permet d'éviter une situation où les deux coûts de fusion seraient faibles voir nuls, ce qui pourrait favoriser les créations de cycles indésirables contenant deux silhouettes d'une même caméra.



(a) Sélectionner l'arc de fusion signifierait qu'un obstacle a empêché la détection de la zone en rouge. Dans cette configuration on a de plus $d(s_1, s_2) = 0$.



(b) Sélectionner l'arc de fusion signifierait que les zones en rouge sont de trop.

FIGURE 2.22 – Illustration du coût de deux arcs de fusion. L'arc de fusion pour lequel le coût est illustré est en rouge. Fusionner deux silhouettes revient à en considérer l'union. i_3 et o_3 représentent les nouveaux points frontières entrant et sortant si l'arc considéré est sélectionné. Les rectangles rouges illustrent des occultations ou sur-détections correspondant au coût de fusion. Les coûts grisés sont nuls, à cause de l'orientation du repère.

Un récapitulatif des différents arcs avec leur fonction est donné en table 2.5

2.3.3.1 Recherche de partition optimale

Nous avons défini un graphe orienté et pondéré dont les cycles correspondent à des associations de silhouettes. Par construction, on s'attend à ce que les associations ayant les poids les plus faibles correspondent aux observations des objets 3D réels. Par conséquent chercher les bonnes associations de silhouettes revient à chercher la partition de coût minimal en cycles disjoints de notre graphe. Comme le nombre possible de telles partitions croît exponentiellement avec le nombre des silhouettes dans le graphe, nous proposons une heuristique efficace. Un noeud du graphe est sélectionné aléatoirement

Appellation	Points concernés	Rôle
Silhouette	points entrant vers point sortant d'une même silhouette	Préserver l'unité de la silhouette.
Création	points sortant vers point entrant d'une même silhouette	Traduire la non observation (équivalent à une occultation totale) dans la seconde caméra.
Fusion	point sortant vers point entrant de deux silhouettes d'une caméra	Pallier la segmentation d'un objet en plusieurs silhouettes.
Association	point sortant vers point entrant de deux silhouettes de deux caméras différentes	Apparier les points frontières entre les deux caméras.

TABLE 2.5 – Récapitulatif des différents arcs.

puis le cycle de poids minimal passant par ce noeud est trouvé avec un algorithme de Dijkstra modifié. Cette modification consiste à contraindre le nombre d'arcs d'association à ne pas dépasser deux pour un unique cycle. La figure 2.23 illustre le type de configuration que nous souhaitons éviter et qui peut survenir lorsque les silhouettes ne sont pas parfaitement alignées dans les deux caméras. Dans un tel cas le cycle correspond à plusieurs objets mais l'on ne sait pas précisément comment apparier les silhouettes entre elles. Comme on suppose qu'une silhouette ne peut appartenir qu'à un seul objet, les noeuds du cycle ainsi trouvés sont supprimés du graphe. Le processus est alors répété sur le sous graphe restant jusqu'à obtention d'un graphe vide. Puisque cette approche dépend de l'ordre dans lequel les noeuds sont sélectionnés, elle est répétée plusieurs fois. Finalement c'est la partition de coût minimal qui sera conservée. Cette phase d'optimisation est résumée par l'algorithme 2.

2.3.3.2 Estimation de la taille et position 3D des objets

Une fois la phase d'optimisation effectuée, nous avons à notre disposition un ensemble d'associations de silhouettes dont il est possible d'extraire des informations. On peut tout d'abord avoir une estimation du nombre d'objets dans la scène que l'on peut espérer plus précise que si elle était effectuée avec une seule caméra. En effet un certain nombre de fausses alarmes ne

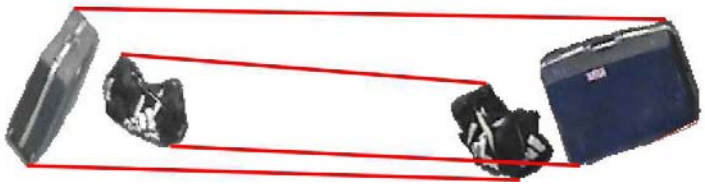


FIGURE 2.23 – Illustration d'un cycle ayant quatre arcs d'association. Pour éviter ce type de situation le nombre d'arc d'association est limité à 2 par cycle.

Entrées:graphe orienté pondéré G, nombre d'itérations nb
p : ensemble vide de partitions
pour $i = 1$ to nb **faire**
  $H = G$
  p_i = partition vide
  tantque H non vide **faire**
    n = noeud aléatoire de H
    c = plus court cycle passant par n utilisant Dijkstra contraint
    $p_i = p_i \cup \{c\}$
    supprimer c de H
  fin tantque
  $p = p \cup \{p_i\}$
fin pour
p_{best} = élément de p de coût minimal
Sorties:p_{best}

Algorithme 2: Phase d'optimisation

sont détecté que dans une seule caméra et n'ont donc pas de bon candidat à l'appariement dans l'autre caméra. Le cycle correspondant trouvé par notre algorithme ne contiendra pas d'arc d'association (mais un arc de création). Le nombre d'objets de la scène visible (au moins en partie) par les deux caméras est donc égal au nombre de cycles contenant des arcs d'association.

Pour chacun de ces cycles, qui correspond à un objet 3D, il est possible de calculer la position 3D des points frontières haut et bas en triangulant les points frontières haut (resp. bas) les plus hauts (resp. bas) des deux caméras. Cela nous permet d'obtenir une estimation de la position et de la taille des objets (estimation car la position des points frontière dépend du point de vue des caméras).

Comme illustré en figure 2.24, il est possible de détecter et partiellement

compenser les erreurs de position et taille dues aux occultations. Si l'angle entre les plans épipolaires de deux points frontières associés est trop important c'est qu'il y a une occultation. Celle ci peut être partiellement compensée dans l'estimation de la position et taille de l'objet en introduisant des points frontières virtuels. Pour le cas de deux points frontières bas (resp. hauts) le point frontière le plus haut (resp. bas) est remplacé par un point frontière virtuel trouvé en rabaissant (resp. remontant) celui ci au niveau de son homologue. Ceci est illustré figure 2.24.

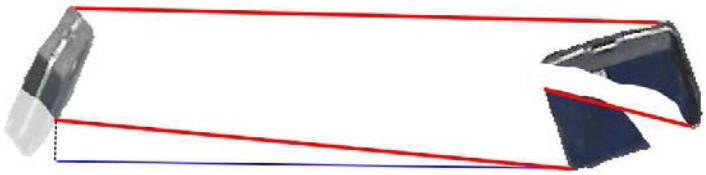


FIGURE 2.24 – Une différence de hauteur de point frontière (ici les points bas) permet de détecter les occultations. Un point frontière virtuel (ici en bleu) permet d'améliorer l'estimation de la position et taille de l'objet 3D.

2.3.3.3 Conclusion sur l'appariement de silhouettes

Nous avons présenté un algorithme de mise en correspondance de silhouettes d'objets stationnaires reposant sur des contraintes purement géométriques et qui ne demande pas de connaissance a priori de la scène 3D. Les occultations même très importantes sont prisent en compte au travers des coûts de création et d'association. Finalement, comme c'est un algorithme direct, il permet de traiter efficacement le cas de scènes non planaires.

2.3.4 Évaluation

Nous allons maintenant présenter une évaluation de notre système. Nous disposons pour cela de deux séquences tournées en intérieur et qui contiennent des situations particulièrement intéressantes en terme d'occultation. On y trouve des objets stationnaires partiellement occultés par des personnes en mouvement, des objets occultés par des parties de la scène, et des objets occultés par d'autres objets.

2.3.4.1 Évaluation qualitative

Nous allons dans un premier temps commencer par une analyse qualitative des résultats. Les images présentées sont rectifiées de sorte que les droites épipolaires sont horizontales. Le contour des silhouettes des objets détectés comme stationnaires sont mis en évidence par une ligne de couleur et un identifiant leur est affecté. Les cycles correspondant aux associations de points frontières sont représentées par des segments d'une même couleur.

On peut tout d'abord remarquer que les caméras ont un point de vue très différent de la scène, ce qui rend difficile la mise en correspondance inter caméra avec un critère reposant sur l'apparence des silhouettes et montre l'intérêt de notre approche purement géométrique. Certains objets qui sont presque entièrement occultés dans une caméra sont complètement visibles dans l'autre caméra, ce qui confirme l'intérêt d'une telle disposition.

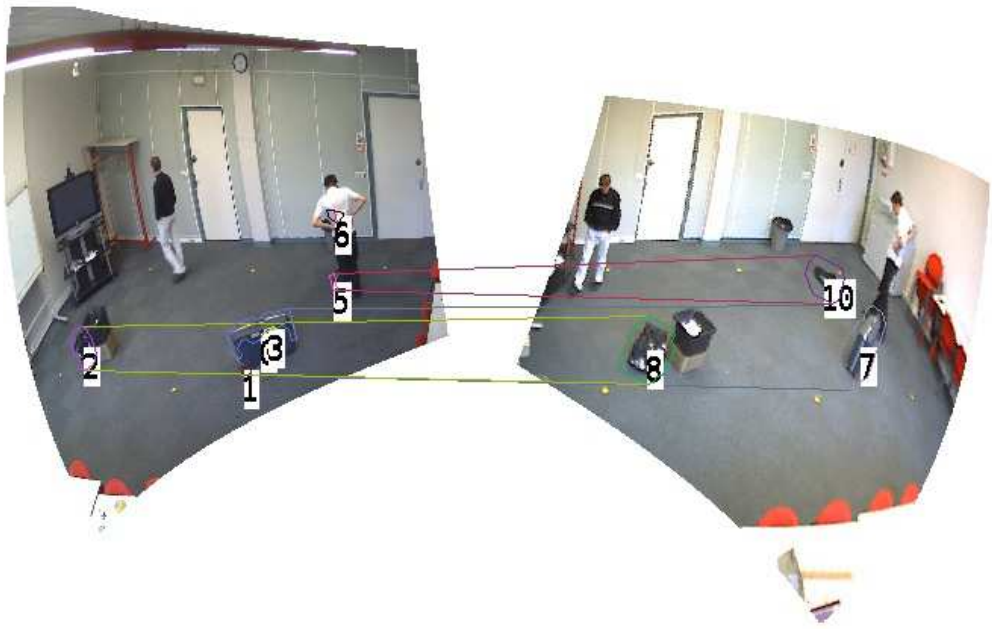


FIGURE 2.25 – Exemple d'association entre les deux caméras. Onze silhouettes au total ont été détectées. L'appariement entre les deux caméras permet de trouver qu'elles correspondent à trois objets seulement.

La figure 2.25 montre onze silhouettes détectées par les deux caméras résultant de fausses alarmes ou d'une sur-segmentation des objets. La valise ayant une texture très similaire à celle de la moquette, celle-ci est détectée

de manière fragmentée. L'algorithme d'appariement, en mutualisant les informations provenant des deux caméras, permet de retrouver le bon nombre d'objets présents dans la scène.

Les figures 2.26 et 2.27 montrent la robustesse de notre algorithme aux occultations. Dans la figure 2.26, la valise, partiellement occultée, n'est qu'en partie détectée dans l'une des caméras. Malgré l'importante différence entre les deux silhouettes l'appariement est effectué car le coût engendré par l'occultation (traduit en pratique par le coût des arcs d'appariement) est inférieur au coût de création des deux silhouettes. La figure 2.27 illustre le fait que comme notre détection d'objets stationnaires est robuste aux occultations temporaires et qu'en conséquence le système complet l'est aussi.

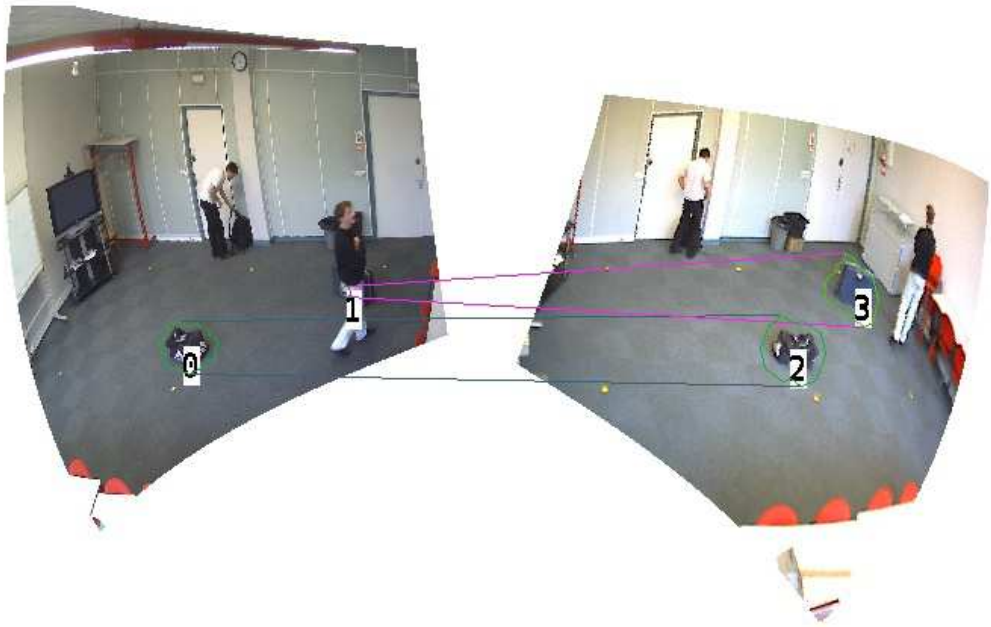


FIGURE 2.26 – La valise est occultée dans une caméra ce qui fait qu'elle n'est, au début, que partiellement détectée. Malgré cela notre algorithme fait l'association correcte.

La figure 2.28 montre l'une des limites de notre approche purement géométrique. Il peut arriver que deux objets réels aient leurs points frontières approximativement sur les mêmes plans épipolaires. Avec les imprécisions de l'acquisition (étalonnage des caméras, erreur de soustraction de fond, ou comme c'est le cas ici en figure 2.28(a) des occultations) un mauvais appariement peut avoir un coût moins élevé que l'appariement souhaité. Si l'on ne

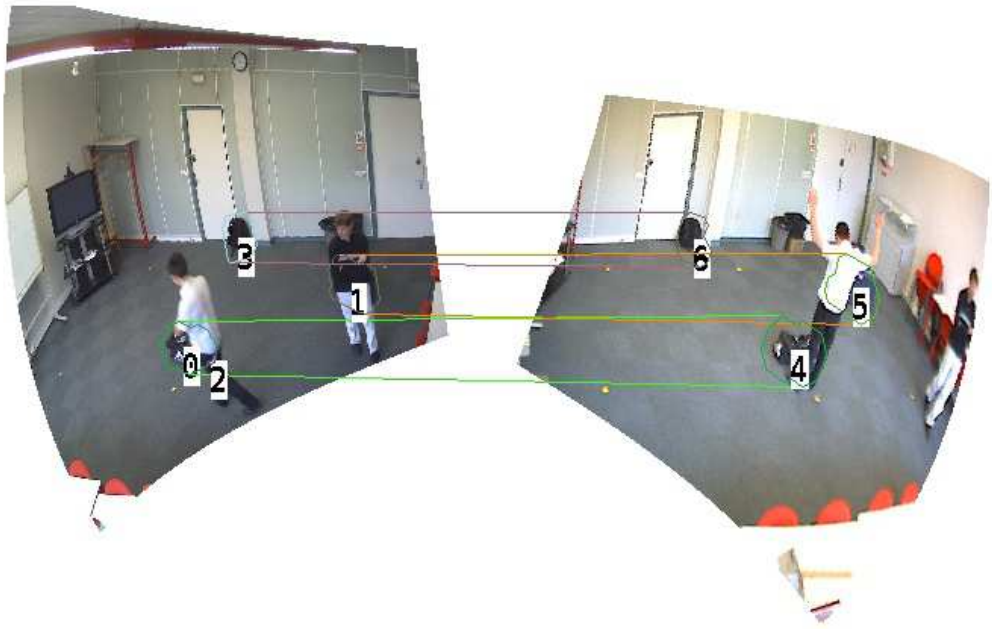


FIGURE 2.27 – Notre détection d'objet stationnaire étant robuste aux occultations temporaires, la phase d'appariement l'est aussi.

peut éviter ce genre de situation, l'appariement redevient cependant correct si chaque objet a un correspondant dans la caméra opposée pour la phase de mise en correspondance, comme on peut le constater en figure 2.28(b).

La figure 2.29 montre que certaines ambiguïtés telles que celles présentées en figure 2.28(a) peuvent être résolues en utilisant une connaissance a priori de la scène 3D. Dans cet exemple on a utilisé le fait que tous les objets doivent se trouver au niveau du sol pour interdire certains mauvais appariements. Un coût infini est donné aux arcs d'associations entre deux points frontières bas dont le point 3D reconstruit serait à plus de vingt centimètres du sol. Cette marge permet de conserver une tolérance à l'imprécision de l'étalonnage des caméras, à la segmentation des objets (on a tendance à sur-segmenter), mais aussi aux petites occultations partielles d'objets dans l'une des caméras, comme on peut l'observer en figure 2.28(b) par exemple.

La figure 2.30 montre le cas d'une valise dont la texture est homogène et très similaire à celle du sol. En conséquence celle-ci est détectée de manière fragmentée, en trois blobs distincts, dans l'une des caméras. On peut constater que les arcs permettent de rassembler ces différents blobs puisqu'ils sont finalement tous contenus dans le cycle correspondant à l'objet 3D.

2.3.4.2 Évaluation quantitative

Pour l'analyse quantitative de notre système, nous avons procédé de la manière suivante. Deux types de vérités terrain sont constitués à la main. Pour chaque caméra, les boites englobantes des objets stationnaires sont enregistrées. Elles vont permettre pour chaque objet détecté dans une caméra de voir s'il correspond à un objet stationnaire ou s'il s'agit d'une fausse alarme. Ainsi par exemple, si un objet stationnaire est détecté en trois parties, comme c'est le cas figure 2.30, il est compté comme trois vrais positifs.

Pour évaluer l'appariement la vérité terrain est constituée, pour chaque image, des associations de boites englobantes des objets appariés désirés. Une association de silhouettes est donc comptabilisée positive lorsque l'union des boites englobantes des objets de chaque caméra est inclus dans la boite englobante de la vérité terrain correspondante. Une fois de plus un unique objet réel peut engendrer plusieurs détections positives. Ce serait par exemple le cas s'il était détecté dans les deux caméras en deux silhouettes, qui ne sont pas fusionnées mais engendrent deux cycles. Nous avons choisi ce type de vérité terrain pour limiter l'impact de la soustraction de fond dans l'évaluation, qui est un paramètre d'entrée de notre système d'appariement. Si un objet est détecté dans les deux caméras en plusieurs composantes non connexes, il est possible que plusieurs cycles soient créés. Chaque cycle correspondrait alors à différentes parties de l'objet. Dans une telle situation chaque détection est considérée comme un vrai positif.

Dans la première séquence, il y a 696 objets à détecter dans les deux caméras, pour 226 appariements. Notre algorithme a détecté 515 VP pour 50 FP dans les deux caméras. L'algorithme de mise à correspondance détecte 226 VP pour 17 FP. Une part importante de l'intérêt du système multi-vues réside donc dans l'importante diminution du nombre d'alarmes à gérer.

Dans la deuxième séquence il y a 1464 objets à détecter dans les deux caméras, pour 508 appariements. Les caméras ont détecté 1275 vrais positifs pour 178 faux positifs. Sans contrainte sur les points frontières triangulés notre algorithme de mise en correspondance détecte 433 VP pour 53 FP. Les scores de rappel et précision peuvent être trouvés dans la table 2.6 pour différentes valeurs de la contrainte sur l'altitude du point frontière bas. En effet dans cette séquence le monde est planaire. Cette connaissance peut être utilisée pour interdire les appariements qui correspondraient à des objets à une altitude trop élevée. Appliquer une telle contrainte peut faire gagner à la fois en rappel et en précision. En effet un mauvais appariement est une fausse alarme et fait donc baisser la précision. De plus, comme chaque sil-

houette n'appartient qu'à un seul cycle, un mauvais appariement interdit par la suite de trouver le bon appariement. Cela engendre alors une diminution du rappel. Cependant une contrainte trop importante, même si elle donne la meilleure précision, génère un rappel limité car elle diminue la tolérance aux occultations, qui peuvent être très importantes comme dans le cas de la figure 2.28(b) page 90. En effet, lors d'une occultation la triangulation des points frontières qui ne correspondent pas réellement au point le plus bas de l'objet 3D reconstruit un point 3D qui n'est pas forcément là où se situe réellement l'objet. La diminution du score de rappel visible en table 2.6 entre l'approche mono-caméra et multi-vues n'est pas significative d'une faiblesse de l'algorithme d'appariement. En effet, en momo-caméra, un objet peut être détecté dans une caméra à un instant donné mais pas dans l'autre, il y a donc dans ce cas un VP pour un FN. Dans l'approche multi-vues, si un objet est détecté dans une seule caméra il ne peut y avoir de bon appariement, on a alors nécessairement zéro VP pour un FN. Il est donc normal que le rappel soit plus important en mono-caméra.

	Rappel	Précision
Mono Caméra	$0,943$	$0,877$
Appariement non contraint	$0,852$	$0,890$
Appariement contraint à $+0,1$m	$0,830$	$0,956$
Appariement contraint à $+0,2$m	$0,877$	$0,941$
Appariement contraint à $+0,3$m	$0,877$	$0,941$
Appariement contraint à $+0,4$m	$0,877$	$0,941$
Appariement contraint à $+0,5$m	$0,889$	$0,940$
Appariement contraint à $+0,6$m	$0,889$	$0,940$

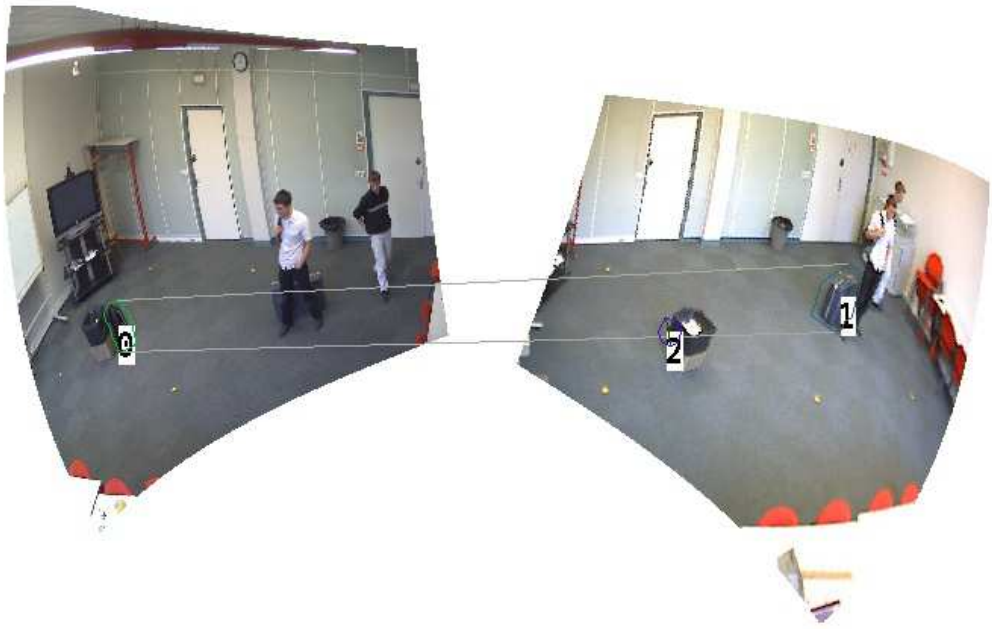
TABLE 2.6 – Statistiques de détection sur la séquence 2. Comparaison de l'approche mono-caméra à l'approche par mise en correspondance dans la paire de caméras. La contrainte d'appariement ici mise en évidence porte sur la hauteur maximale du point frontière bas de l'objet. Plus la contrainte est forte moins l'algorithme est sensible aux ambiguïtés d'appariement, mais moins l'algorithme est robuste aux occultations.

2.4 Conclusion

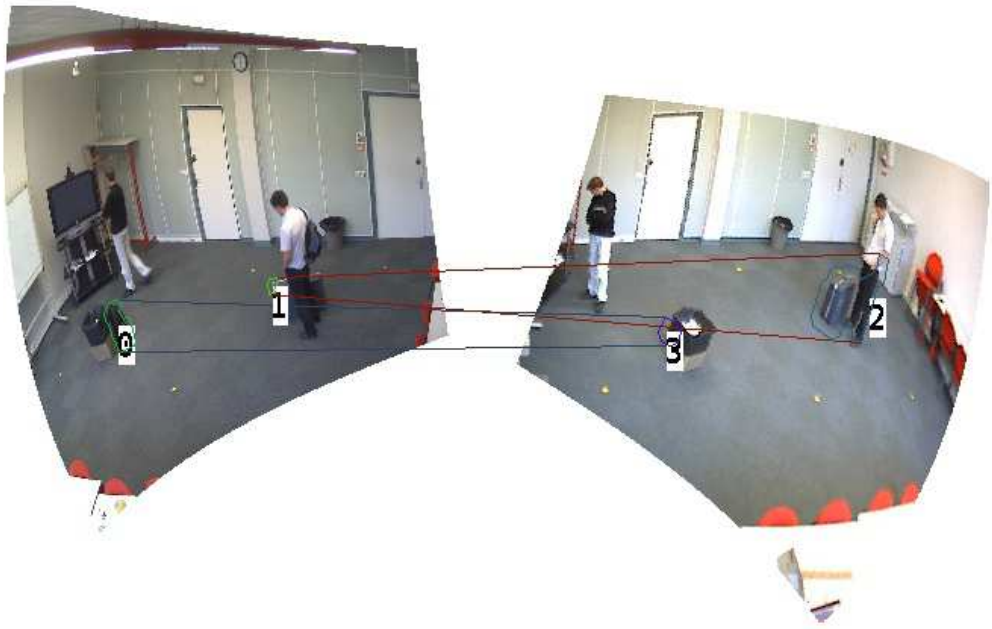
Nous avons présenté et évalué un système de détection d'objets stationnaires dans une paire de caméra qui repose seulement sur des contraintes

géométriques. Une phase de segmentation robuste aux occultations permet sous certaines conditions de segmenter en plusieurs silhouettes les blobs correspondant à plusieurs objets stationnaires. La phase d'appariement permet l'association de plusieurs silhouettes entre les deux caméras. Elle se fait en associant des points frontières par la recherche d'une partition de coût minimal en cycles dans un graphe orienté. Des contraintes sur ces associations font que chaque cycle représente un seul objet 3D, même si celui ci peut être composé de plusieurs silhouettes par caméra. L'évaluation montre que l'approche multi-vues a deux intérêts majeurs. D'une part elle permet de faire diminuer le nombre global d'alarmes. Pour un objet 3D il n'y a plus une alarme levée par caméra mais une alarme pour la paire de caméra. De plus, la phase d'appariement permet d'augmenter significativement la précision du système en filtrant certains faux positifs qui ne sont détectés que dans une caméra.

Nous allons dans le chapitre suivant appliquer ce système au cas particulier d'une paire de caméras PTZ effectuant un tour de garde.



(a) Certains objets peuvent avoir leurs points frontières situés dans des plans épipolaires assez proches. Cela fait que dans certaines configurations l'appariement de coût minimal n'est pas celui souhaité. Ici l'occultation du sac, et la non détection de la valise participent à ce mauvais appariement.



(b) L'ambiguïté de la situation en figure 2.28(a) est ici levée par le début de détection de la valise dans la deuxième caméra.

FIGURE 2.28 – La contrainte épipolaire laisse toujours place à des ambiguïtés pouvant engendrer des mauvais appariements.

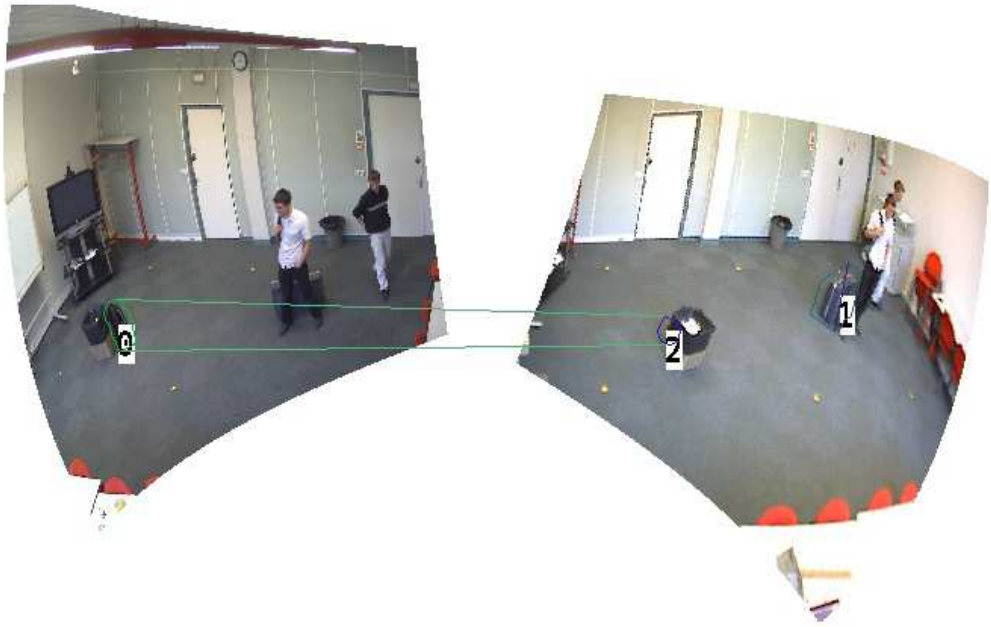


FIGURE 2.29 – Un a priori sur le monde 3D peut permettre d'éviter certains mauvais appariements. Ici, dans la même situation que celle présentée en figure 2.28(a), une contrainte forçant les objets appariés à être proches du sol permet de trouver le bon appariement.

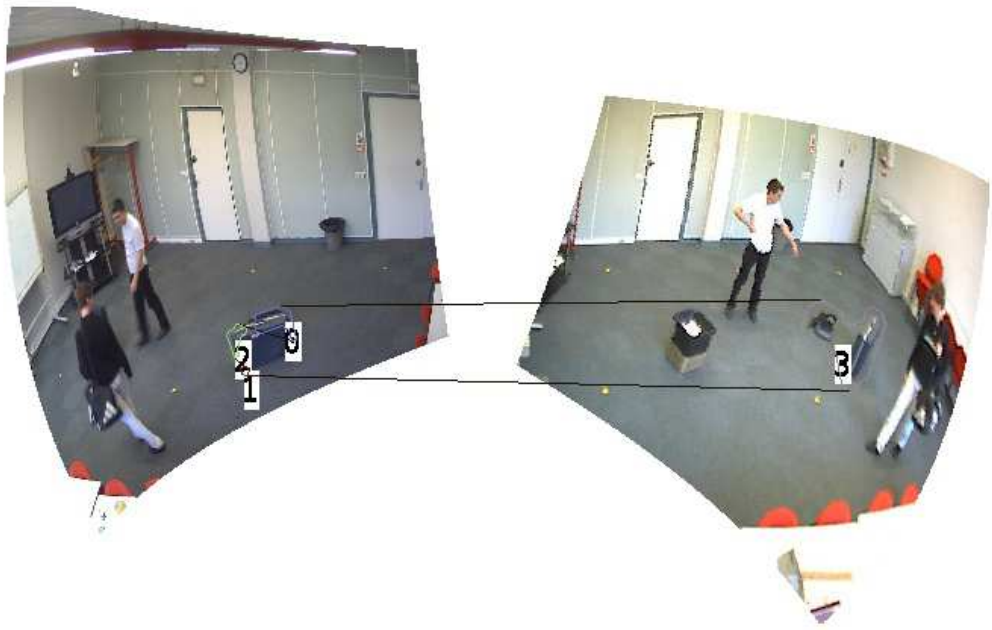


FIGURE 2.30 – Illustration de l'utilité des arcs de fusion. La valise qui a la même texture que la moquette est mal détectée, trois silhouettes sont détectées au lieu d'une seule. Notre algorithme est capable de les fusionner.

Chapitre 3

Détection d'objets stationnaires par une paire de caméras PTZ

3.1 Introduction

On rappelle que contrairement aux caméras classiques, les caméras PTZ sont commandables en orientation et en zoom. Nous proposons d'exploiter ces capteurs pour gérer de larges zones. Pour cela la caméra réalise un tour de garde et, pour chaque position, la focale va être adaptée de façon à disposer de la résolution adéquate pour détecter des objets d'une taille similaire à celle d'une valise. En contrepartie seulement une partie de la scène est visible à chaque instant. La caméra PTZ parcourt donc indéfiniment un ensemble de positions *pan, tilt, zoom* qui définissent ce que l'on appellera aussi des *vues*. Les paramètres du tour de garde sont choisis pour que la scène observée soit entièrement traitée. Pour assurer une continuité dans la détection, les vues adjacentes du tour de garde se chevauchent.

Dans ce chapitre, nous allons mettre en oeuvre un système de détection d'objets stationnaires s'appuyant sur un couple de caméras PTZ. Ces caméras, parce-que leurs paramètres internes et externes peuvent varier, ont besoin d'une procédure d'étalonnage spécifique. De plus, comme chacune des vues de la caméra est mise à jour de manière totalement désynchronisée, une adaptation de l'algorithme d'appariement est nécessaire.

3.2 Étalonnage d'une paire de caméras PTZ

3.2.1 Introduction

Il existe dans la littérature de nombreuses méthodes d'étalonnage spécifiques aux caméras PTZ. On peut notamment citer les travaux de Jain *et al.* [48] qui donne une méthode d'étalonnage très complète, ou encore Sinha *et al.* [94] et Trajkovic [98] pour l'estimation des paramètres internes de la caméra, ou de Micheloni *et al.* [72] et Wan *et al.* [106, 105] pour le cas d'une paire de caméras PTZ. Gardel *et al.* [37] se sont intéressés en détail à l'influence du zoom sur les paramètres intrinsèques d'une caméra PTZ.

Nous allons présenter ici une méthode d'étalonnage de caméra PTZ qui a été développée au laboratoire. Tout d'abord nous présentons une méthode pour estimer la focale, puis pour obtenir la matrice des paramètres externes de la paire de caméras.

Une caméra est définie par un centre optique, généralement noté O, un axe optique et un *plan image*, comme illustré en figure 3.1. L'intersection entre l'axe optique et le plan image est le point principal P. La distance f entre le centre optique O et le point principal P est la distance focale.

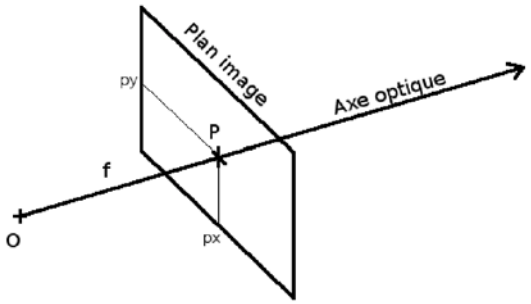


FIGURE 3.1 – Représentation des paramètres intrinsèques d'une caméra : le centre optique O, l'axe principal, le plan image et la focale f.

On considère que les caméras sont de type appareil à sténopé simple. Un point du monde (X, Y, Z) se projette en un point image (x, y) par la relation

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = KR[I_3|-C] \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{3.1}$$

où $K \in \mathcal{M}_{3\times3}(\mathbb{R})$ est la matrice des paramètres internes de la caméra, $R \in \mathcal{M}_{3\times3}(\mathbb{R})$ est une matrice de rotation, $[I_3|-C] \in \mathcal{M}_{3\times4}(\mathbb{R})$, avec C les

coordonnées du centre optique. On suppose que les pixels sont carrés, de cette façon,

$$K = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

où (x_0, y_0) est la position du point principal. De plus on néglige les distortions et on suppose que les axes de rotation pour le *pan* et *tilt* sont concourants et que leur point d'intersection coïncide avec le centre optique. Ce sont des hypothèses fortes mais qui se justifient en pratique, notamment de part le niveau de zoom que nous utilisons.

3.2.2 Estimation de la focale

Si les caméras PTZ sont commandables en angles (*pan* et *tilt*) et en zoom, la valeur de la focale n'est cependant pas directement contrôlable. Dans le cas particulier de la caméra Axis 233 D que nous avons utilisée, le constructeur permet le contrôle du zoom par l'intermédiaire d'une variable sans unité $z \in [1, 9999]$. Il est donc nécessaire d'estimer la fonction permettant de passer d'une valeur de zoom à une distance focale.

La méthode présentée est très similaire à celle de Trajkovic [98] dont le principe pour estimer la focale est de capturer deux images à un *pan* fixe, mais en faisant varier de φ l'angle du *tilt*. Il calcule alors le déplacement vertical d en pixels du point principal, et obtient la focale f par la relation

$$f = \frac{-d}{\tan \varphi} \tag{3.3}$$

La méthode proposée au laboratoire est plus générique que celle de Trajkovic car elle ne repose pas sur le seul déplacement du point principal mais de n'importe quel point de l'image.

Considérons une caméra définie par ses paramètres de positionnement extrinsèque $R[I_3| - C]$ et sa matrice de calibration interne K. Avec ces définitions la matrice P est donnée par :

$$P = KR[I_3| - C] \tag{3.4}$$

P est la matrice de projection d'un point 3D sur le plan image de la caméra.

On considère maintenant deux états de la caméra caractérisés par les matrices de rotation R_1 et R_2. Soit un point du monde X qui se projette dans l'image en les points x_1 et x_2 pour les deux états respectifs.

On a donc :

$$x_2 = KR_2R[I_3|-C]X \tag{3.5}$$
$$= KR_2R(KR_1R)^{-1}(KR_1R)[I_3|-C]X \tag{3.6}$$
$$= KR_2R_1^{-1}K^{-1}x_1 \tag{3.7}$$
$$= KR_{1\to2}K^{-1}x_1 \tag{3.8}$$
$$= P_{1\to2}x_1 \tag{3.9}$$

On ne fait varier d'une valeur φ que le paramètre *tilt* entre les deux prises de vue. La relation entre deux points $\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}$ devient alors :

$$P_{1\to2}\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}\begin{bmatrix} \frac{1}{f} & 0 & \frac{-x_0}{f} \\ 0 & \frac{1}{f} & \frac{-y_0}{f} \\ 0 & 0 & 1 \end{bmatrix}\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \tag{3.10}$$
$$= \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{bmatrix}\begin{pmatrix} \frac{x_1-x_0}{f} \\ \frac{y_1-y_0}{f} \\ 1 \end{pmatrix} \tag{3.11}$$
$$= \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{pmatrix} \frac{x_1-x_0}{f} \\ \frac{y_1-y_0}{f}\cos\varphi - \sin\varphi \\ \frac{y_1-y_0}{f}\sin\varphi + \cos\varphi \end{pmatrix} \tag{3.12}$$
$$= \begin{pmatrix} (x_1-x_0) + x_0\left(\frac{y_1-y_0}{f}\sin\varphi + \cos\varphi\right) \\ (y_1-y_0)\cos\varphi - f\sin\varphi + y_0\left(\frac{y_1-y_0}{f}\sin\varphi + \cos\varphi\right) \\ \frac{y_1-y_0}{f}\sin\varphi + \cos\varphi \end{pmatrix} \tag{3.13}$$
$$= \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \tag{3.14}$$

En injectant la troisième ligne des équations 3.13 et 3.14 dans les deux premières lignes, on obtient les deux équations suivantes :

$$0 = (x_1 - x_0) - (x_2 - x_0)\left(\frac{y_1 - y_0}{f}\sin\varphi + \cos\varphi\right) \tag{3.15}$$
$$0 = (y_1 - y_0)\cos\varphi - f\sin\varphi - (y_2 - y_0)\left(\frac{y_1 - y_0}{f}\sin\varphi + \cos\varphi\right) \tag{3.16}$$

Les équations 3.15 et 3.16 permettent toutes deux de retrouver la valeur f de la focale, mais une étude théorique des erreurs incite à utiliser l'équation 3.16.

Finalement on obtient

$$f = \frac{y_1 - y_2}{2\tan\varphi}\left(1 + \sqrt{1 - 4\frac{(y_1 - y_0)(y_2 - y_0)}{(y_1 - y_2)^2}\tan^2\varphi}\right) \tag{3.17}$$

L'algorithme 3 résume la procédure d'estimation de la focale.

- Capturer deux images avec recouvrement à focale fixée en faisant varier le *tilt*.
- Calculer les points d'intérêt et descripteurs sur chaque image.
- Apparier les points d'intérêt entre les deux images.
- Calculer la focale pour chaque paire de points en utilisant l'équation 3.17.
- Estimer la focale à partir des valeurs précédemment obtenues.
- Répéter le processus avec plusieurs paires d'images.

Algorithme 3: Estimation de la focale d'une caméra PTZ.

3.2.3 Étalonnage extrinsèque

On suppose les paramètres internes des caméras connus. Il reste donc à estimer les paramètres extrinsèques des caméras pour pouvoir les positionner dans le repère du monde. Puisque notre algorithme de mise en correspondance repose sur la géométrie épipolaire des deux caméras, on privilégie la précision sur la position relative des deux caméras à leur position absolue dans le repère du monde. Pour cette raison nous allons calculer la matrice essentielle E de la paire de caméras puis les positionner dans le repère du monde.

3.2.3.1 Cas d'une paire de caméras fixes

Considérons tout d'abord le cas d'une paire de caméras stationnaires. L'algorithme des cinq points proposé par Nistér [78] permet de calculer la matrice essentielle E entre ces deux caméras. Cependant cet algorithme fournit jusqu'à dix solutions, il est donc nécessaire de connaître des appariements supplémentaires. La matrice E choisie sera celle qui minimise l'angle entre les plans épipolaires des couples de points appariés. Afin que des mauvais appariements ne faussent pas l'estimation, la matrice E est estimée avec un algorithme d'estimation robuste de type RANSAC.

A partir de cette matrice essentielle E il est possible de trouver la matrice de rotation R et de translation t entre les deux caméras [45]. Quatre configurations de la paire de caméras sont possibles, comme illustré en figure 3.2. Celle qui correspond à la configuration réelle est celle où tous les points de la scène sont situés en face des deux caméras.

Pour connaître la position d'une caméra dans le repère du monde, il suffit de connaître la position de trois points visibles par la paire de caméras. Ces trois points peuvent être triangulés et il suffit alors d'estimer la transformation rigide permettant de passer des coordonnées dans le repère monde aux

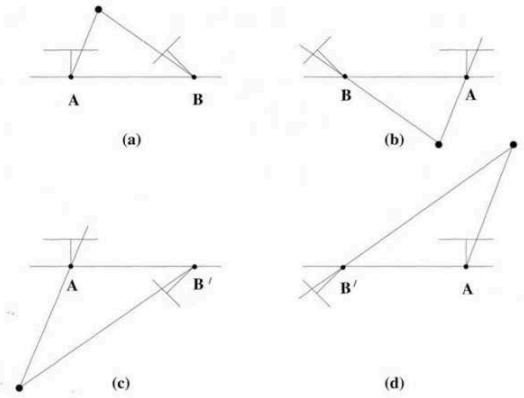


FIGURE 3.2 – Pour une matrice essentielle quatre configurations de rotations et translations sont possibles. Les configurations des caméras et du point monde correspondant sont illustrées dans les sous images (a), (b), (c) et (d). Figure tirée de [45].

coordonnées dans le repère d'une des caméras [47], puis d'en extraire la rotation R_0 et translation t_0 correspondante. On obtient finalement les matrices de projection pour les caméras 1 et 2 :

$$P_{C_1} = K_1[R_0, t_0] \tag{3.18}$$

$$P_{C_2} = K_2[R, t][R_0, t_0] \tag{3.19}$$

3.2.3.2 Cas d'une paire de caméras PTZ

Dans le cas précédent, la méthode d'estimation de la matrice essentielle repose sur la mise en correspondance de points projetés sur les plans image de chaque caméra. Dans le cas des caméras PTZ le plan image de la caméra n'est pas unique, il y a en un différent pour chaque paire de paramètres $(pan, tilt)$. Plutôt que de se restreindre à un unique couple de paramètres on choisi de projeter les points du monde observés avec différentes valeurs de *pan* et *tilt* sur un plan fixe unique P choisi arbitrairement, comme illustré en figure 3.3. De cette façon il est possible de couvrir tout le champ de vue de la caméra PTZ tout en profitant de sa capacité de zoom élevée pour se focaliser précisément sur des points très espacés de la scène.

Ce sont finalement les points du plan P qui sont utilisés et appariés avec les points de la seconde caméra pour l'estimation de la matrice essentielle E.

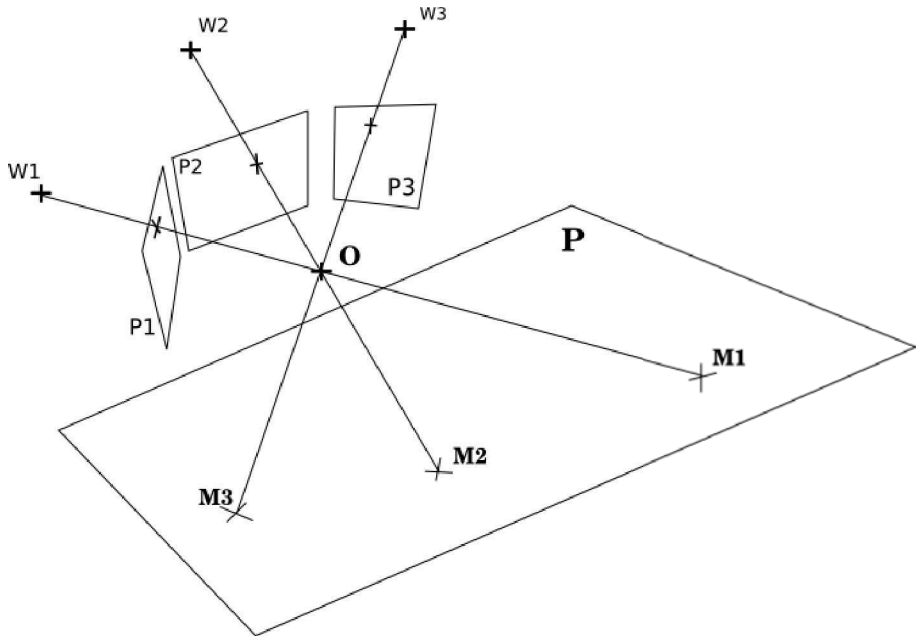


FIGURE 3.3 – Les points du monde W_1, W_2, W_3 sont projetés sur le plan P unique à la caméra en M_1, M_2, M_3.

Pour calibrer une paire de caméras PTZ on focalise donc les caméras sur un ensemble de points 3D de la scène. L'image de chacun des points est projetée sur les plan P_1 et P_2 respectifs à chaque caméra. Se sont finalement les couples de points sur ces deux plans qui sont utilisé pour estimer la matrice essentielle E.

La géométrie des systèmes d'acquisition étant sphérique, nous travaillons en coordonnées sphériques. Cela a de plus pour avantage de simplifier la rectification des images. Pour afficher les paires de panoramas rectifiés, nous procédons donc de la manière suivante, illustrée figure 3.4. Comme les caméras sont étalonnées extrinsèquement on effectue un changement de repère. Chacune est placée dans un nouveau repère (en rouge dans la figure). Les nouveaux repères sont choisis de sorte que l'un est la translation de l'autre, et que l'un de leurs axes soit porté par la droite O_gO_d. Dans ces nouveaux repères, en coordonnées sphériques, si φ est l'angle de rotation autour de O_gO_d alors les plans épipolaires sont les plans d'équation $\varphi = C$, avec C une constante. Les coordonnées d'un point dans le panorama sont ses coordonnées sphériques. Dans l'image panoramique un décalage horizontal (resp. vertical) d'un pixel correspond donc à un décalage angulaire constant.

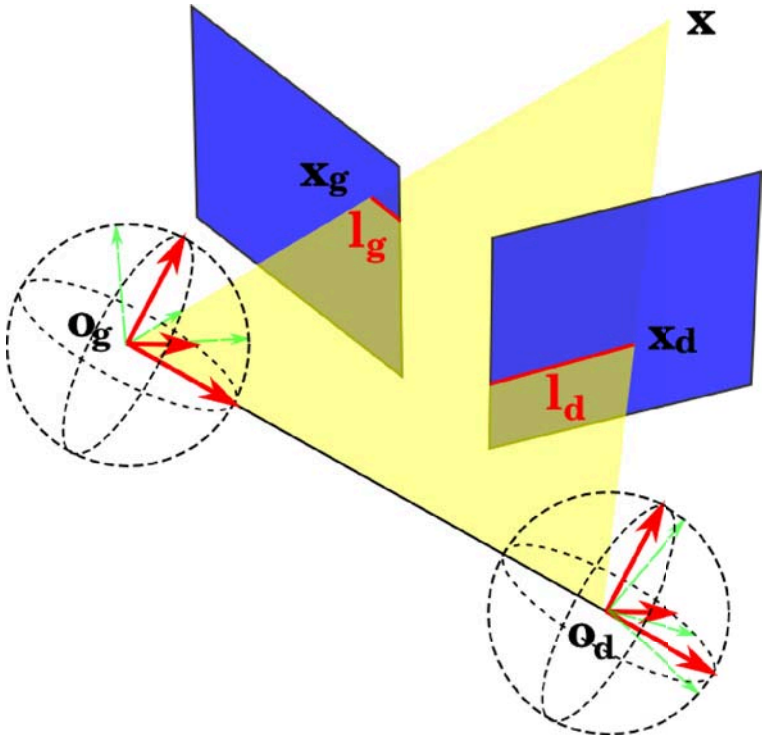


FIGURE 3.4 – Pour construire les panoramas et les rectifier, on travaille en coordonnées sphériques dans un nouveau repère (en rouge). Ces repères, dans lesquels on utilise les coordonnées sphériques, ont un axe porté par la droite O_gO_d reliant les deux centres optiques. O_g et O_d sont les centre optiques des caméras. l_g et l_d sont les droites épipolaires associées aux points x_g et x_d.

3.2.4 Conclusion sur l'étalonnage

Bien que la qualité de cette méthode d'étalonnage de caméra PTZ ne soit pas évaluée quantitativement on pourra constater en section 3.4 que les images panoramiques générées sont de qualité grandement suffisante pour notre application. Comme elles sont générées sans effectuer de quelconque recalage, ces images panoramiques sont bien un indicateur de la précision de notre étalonnage.

En pratique une quinzaine de points seulement sont nécessaires pour le calibrage. Les appariements sont effectués en focalisant à la main les PTZ sur des points du monde.

3.3 Adaptation de l'algorithme d'appariement au cas PTZ

Dans le cas des caméras PTZ, chacune des vues du tour de garde (qui correspond à un jeu de paramètres *pan*, *tilt* et *zoom*) est considérée comme une caméra fixe indépendante. Comme deux vues adjacentes n'ont pas un champ de vision disjoint il est possible qu'un objet soit détecté dans plusieurs vues, comme illustré en figure 3.5.

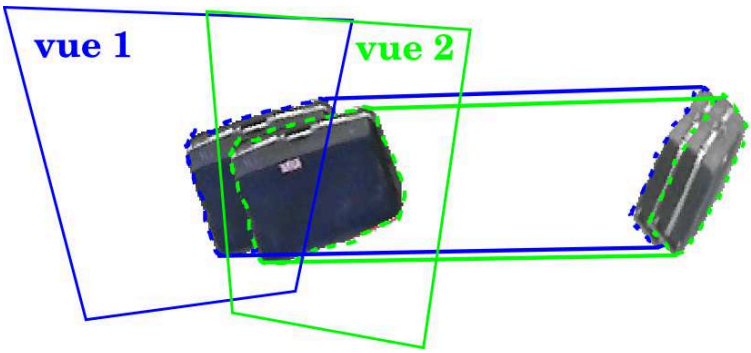


FIGURE 3.5 – Illustration du problème lié aux champs de vision joints des vues adjacentes. La valise est détectée une fois dans chaque vue de chaque caméra. Il en résulte que deux cycles sont créés et la valise est donc détectée deux fois. Pour une meilleure compréhension les silhouettes de la valise détectées sur les deux vues sont volontairement décalées.

Pour pallier ce problème nous fusionnons les silhouettes provenant de deux vues différentes et d'intersection non nulle. Cela a pour effet de diminuer le nombre de silhouettes à apparier et donc de limiter les sur-détections d'objets, comme illustré en figure 3.6.

3.4 Évaluation du système pour une paire de caméras PTZ

Nous allons maintenant présenter une évaluation de notre système de détection d'objets stationnaires par une paire de caméras PTZ. Nous avons pour cela quatre séquences. Les deux premières sont acquises en intérieur en parallèle de celles présentées au chapitre précédent avec des caméras stationnaires. Les deux autres sont tournées en extérieur.

FIGURE 3.6 – La fusion intra-caméra des silhouettes provenant de vues différentes permet de garantir l'unicité de la détection d'un objet. Pour une meilleure compréhension, les silhouettes de la valise détectée sur les deux vues sont volontairement décalées.

3.4.1 Évaluation qualitative

Nous allons dans un premier temps commencer par une analyse qualitative des résultats. De même qu'au chapitre précédent, les résultats présentés sont sous forme d'images rectifiées de sorte que les droites épipolaires sont horizontales. Les images panoramiques présentées sont générées sans recalage et sont donc représentatives de la qualité de l'étalonnage des caméras. L'image acquise la plus récente est toujours celle affichée au dessus.

Dans les séquences tournées en intérieur, chaque caméra parcourt un tour de garde constitué de trois positions avec des paramètres *pan*, *tilt*, et *zoom* différents. La figure 3.7 montre la scène considérée. On peut y observer les trois vues du tour de garde des caméras PTZ. A titre indicatif les caméras fixes utilisées au chapitre précédent sont représentées. On peut constater que les caméras PTZ permettent d'avoir un champ de vue plus large pour une résolution toujours adaptée. On peut aussi observer le chevauchement des vues adjacentes des caméras PTZ.

Les figures 3.8 et 3.9 montrent des phénomènes déjà observés au chapitre précédent. La figure 3.8 montre que du fait du point de vue très différent des caméras sur la scène, un objet peut être entièrement visible dans une caméra mais presque complètement occulté dans l'autre. L'appariement est effectué car la petite portion d'objet observée dans une caméra explique une partie de l'observation dans l'autre caméra.

La figure 3.9 montre l'intérêt de notre segmentation des objets stationnaires. Bien que les blobs des objets soient connexes dans les deux images, nous avons pu les séparer, ce qui permet de trouver le bon nombre d'objets dans l'image.

La figure 3.10 montre un inconvénient de notre méthode. Bien que dans

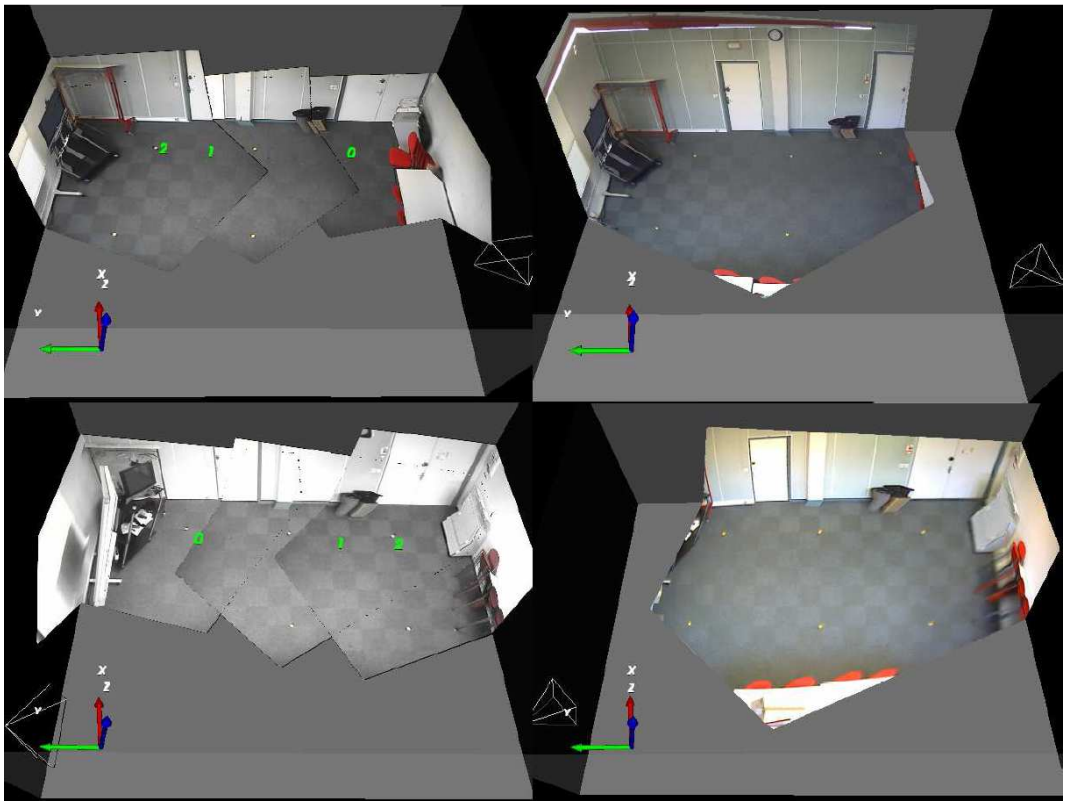


FIGURE 3.7 – Modélisation de la scène et des positions des caméras pour les acquisitions en intérieur. Les images acquises par les caméras sont projetées sur le modèle de la scène. Colonne gauche : pour les caméras PTZ. On peut constater le chevauchement entre les vues adjacentes d'une caméra. Colonne droite : pour les caméras fixes (séquences utilisées au chapitre précédents).

la vue la plus récente le fond soit clairement visible cette information n'est pas utilisée. L'objet qui était visible ou occulté dans une vue adjacente est toujours supposé présent dans la scène. Pour qu'un objet ne soit plus détecté il faut que sa disparition soit observée dans toutes les vues où il a été visible. Cela a pour effet de générer des faux positifs, même si d'un point de vue applicatif ils ne sont pas trop gênants puisqu'il ne s'agit pas d'une nouvelle détection mais du prolongement d'une détection déjà existante.

Pour les séquences tournées en extérieur, chaque caméra parcourt un tour de garde de huit vues et deux valeurs de zoom différentes. La durée de chaque tour de garde est d'environ une quinzaine de secondes. Les deux caméras sont espacées de treize mètres et sont à $4,70$ mètres de haut. Les objets d'intérêt sont situés à une distance de quinze à vingt mètres des caméras.

FIGURE 3.8 – Le sac (silhouettes 0 et 4) est fortement occulté par une poubelle dans l'une des caméra, mais les silhouettes sont correctement appariées.

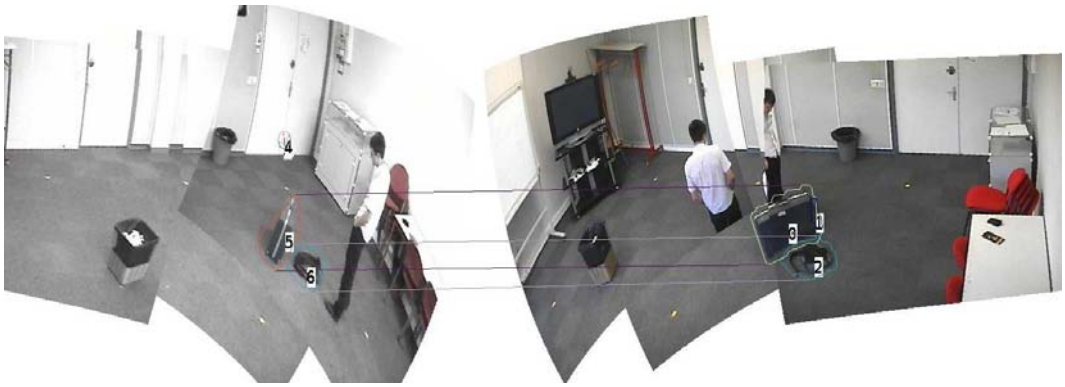


FIGURE 3.9 – Deux objets stationnaires adjacents et arrivés à des instants différents sont correctement segmentés. L'appariement permet de trouver qu'il y a bien deux objets.

FIGURE 3.10 – Limitation de notre approche. Bien que la disparition de l'objet ait été constatée dans la vue la plus récente de la caméra il est encore présent dans une autre vue mise à jour moins récemment. Ceci engendre un faux positif le temps que la vue incriminée soit mise à jour.

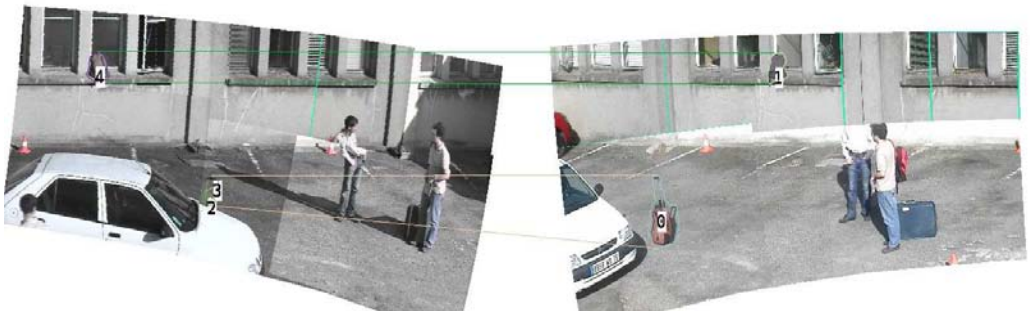


FIGURE 3.11 – Séquence *extérieur 2*. Détection d'un objet fortement occulté. Notre méthode permet de détecter l'occultation et de corriger l'estimation de l'altitude et la taille de l'objet.

La figure 3.11 montre un exemple de situation où une approche par projection des silhouettes sur le plan du sol ne pourrait pas aboutir à une détection de l'objet. Comme l'essentiel de la partie basse de l'objet est occultée, dont le point de contact avec le sol, il est peut probable que les projections des silhouettes s'intersectent. Dans un premier temps la valise (association des silhouettes $0 \leftrightarrow (2,3)$) est détectée à une altitude de $0,31m$ pour une taille de $0,62m$. La correction de cette estimation en ramenant les points frontières sur les plans épipolaires les plus éloignés permet d'obtenir une altitude estimée à $0,02m$, une taille de $0,91m$, ce qui correspond à un taux d'occultation de 31%. La taille réelle de cette valise est de $1,05m$.

La figure 3.12 montre la sensibilité de notre approche à la qualité de

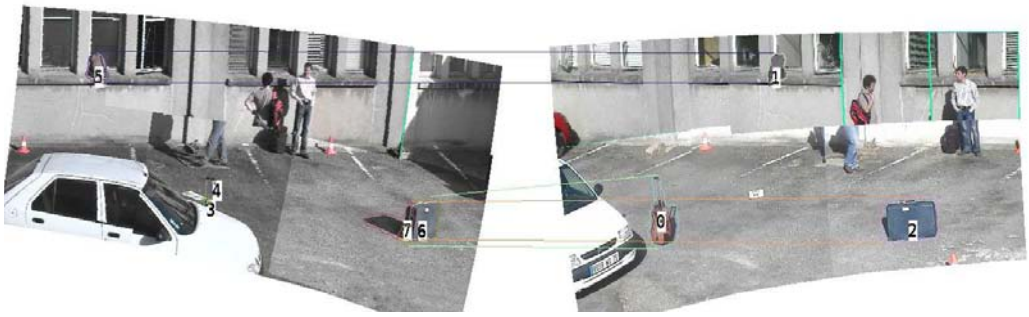


FIGURE 3.12 – Séquence *extérieur 2*. La sur-segmentation de la valise (silhouettes 6 et 7, caméra gauche) engendre un mauvais appariement car le coût de l'arc de création de la silhouette qui est de trop est supérieur au coût de création des silhouettes de la valise occultée (silhouettes 3 et 4). Une contrainte sur l'altitude des objets n'est pas adaptée pour résoudre le problème si l'on souhaite aussi pouvoir détecter les objets sur le rebord de mur.

la segmentation. La valise est sur-segmentée dans la caméra gauche, ce qui engendre un mauvais appariement. Comme dans le cas des séquences en intérieur cette erreur pourrait être évitée en imposant comme contrainte que les objets soient au niveau du plan du sol. Cependant avec cette contrainte on ne pourrait plus détecter les objets posés sur le rebord de la fenêtre. En effet ce mauvais appariement correspond à un objet posé à $1,02m$ du sol et d'une taille de $0,69m$, alors que le sac sur le rebord de la fenêtre est à $1,18m$ pour une taille de $0,52m$.

La figure 3.13 montre six objets stationnaires de différentes tailles. Elle permet de vérifier que la qualité de l'estimation de la taille des objets. Les valeurs obtenues sont données en table 3.1. On constate que l'estimation est correcte sauf pour deux objets. La taille du journal (silhouettes 9 et 2) est surestimée, mais c'est un résultat prévisible et qui résulte de plusieurs facteurs : erreurs d'étalonnage, de segmentation, de précision sur la localisation des points frontière. La seconde mauvaise estimation (silhouettes 10 et 3) est quant à elle due à une occultation de l'objet, puisque il n'est que partiellement visible dans l'image. Cette occultation peut être détectée et l'estimation de la taille de l'objet peut être corrigée comme expliqué au chapitre précédent en ramenant les points-frontière sur un même plan épipolaire.

La figure 3.15 page 108 illustre une scène particulièrement difficile en termes d'occultations. L'une des caméras montre une importante sur-segmentation des objets d'intérêt. L'appariement inter-caméras permet cependant de limi-

Association	Altitude (m)	Taille estimée (m)	Taille réelle (m)
$8 \leftrightarrow 0$	$-0,01$	$0,57$	$0,53$
$7 \leftrightarrow 1$	$-0,07$	$0,51$	$0,51$
$9 \leftrightarrow 2$	$-0,03$	$0,09$	$0,01$
$10 \leftrightarrow 3$	$0,16$	$0,13$ (corrigée $0,21$)	$0,20$
$12 \leftrightarrow 4$	$-0,01$	$0,15$	$0,15$
$11 \leftrightarrow 5$	$0,01$	$0,07$	$0,13$

TABLE 3.1 – Altitudes et tailles estimées des objets détectés figure 3.13. La colonne *Association* fait référence aux numéros des silhouettes dans la figure 3.13, chaque association correspond donc à un objet.

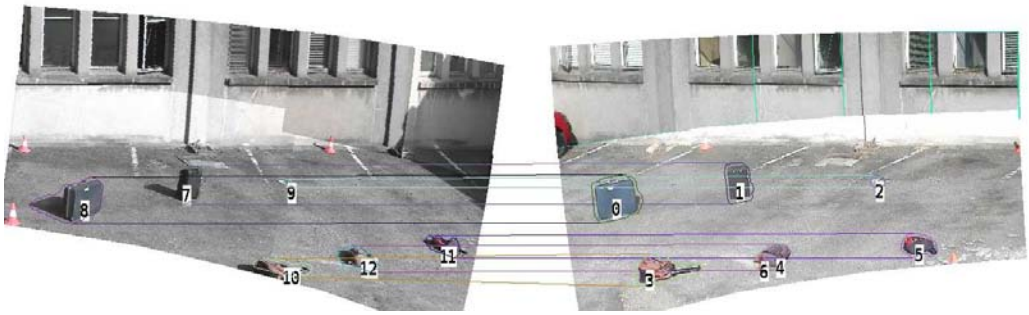


FIGURE 3.13 – Séquence *extérieur 1*. Six objets stationnaires détectés. Leurs tailles estimées sont données table 3.1.

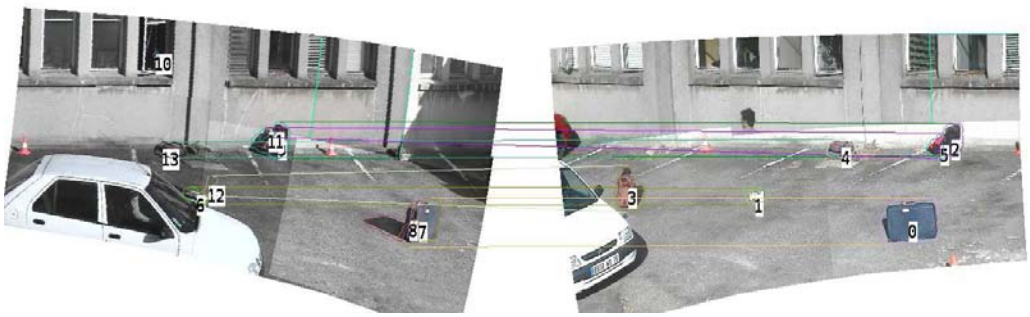


FIGURE 3.14 – Séquence *extérieur 2*. Un sac visible par une seule des caméras (silhouette 3) est apparié avec un faux positifs de la seconde caméra (silhouette 12).

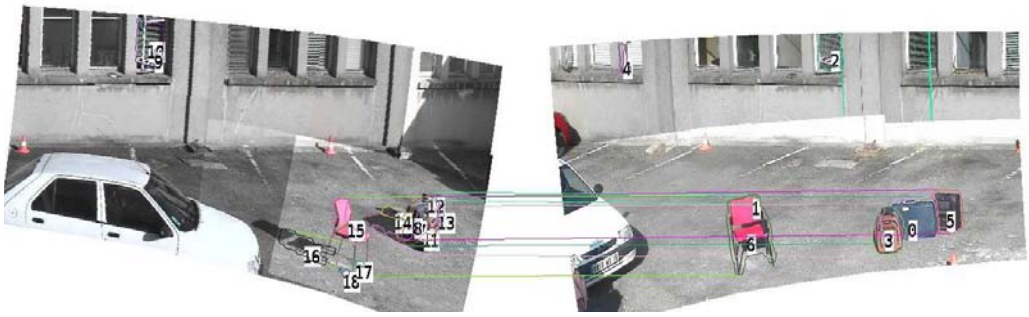


FIGURE 3.15 – Séquence *extérieur 2*. Situation difficile en terme d'occultation. La chaise dont les pieds sont très fins est détectées en plusieurs silhouettes, qui seront finalement correctement fusionnées.

ter le nombre de faux positifs puisque les objets ont été bien détectés dans l'autre caméra. L'algorithme a effectué cinq appariements, pour quatre objets stationnaires réellement présents dans la scène. Le faux positif détecté est du à la silhouette numérotée 1 et qui est un reste d'objet stationnaire précédemment détecté puis partiellement occulté par la chaise avant qu'il ne soit enlevé de la scène. On peut cependant remarquer que la chaise, très mal détectée dans la caméra gauche est bien appariée (association des silhouettes $6 \leftrightarrow (15, 16, 18)$). Il en résulte qu'elle est estimée à une altitude de $0,00m$ pour une taille de $0,82m$, ce qui correspond à sa taille réelle. Ce cas est un bon exemple de l'utilité des arcs de fusion introduits pour associer des groupes de silhouettes.

La figure 3.14 montre une limitation de notre système. Pour qu'un objet soit détecté, nous supposons qu'il est visible dans les deux caméras. Il y a bien entendu des situations où cela n'est pas le cas. Notre système permet donc d'augmenter la précision de la détection (moins de faux positifs), mais pas d'en augmenter le rappel.

3.4.2 Évaluation quantitative

Pour l'analyse quantitative des résultats nous procédons de la même façon qu'au chapitre précédent. Deux types de vérité terrain sont constituées à la main. Pour chaque caméra les boîtes englobantes des objets stationnaires sont enregistrées. Elles vont permettre d'évaluer la détection mono-caméra. La vérité terrain des associations de boites englobantes est aussi enregistrée et permet l'évaluation du système complet, jusqu'à la phase d'appariement. Dans le cas PTZ, la constitution des vérités terrain est plus délicate que dans

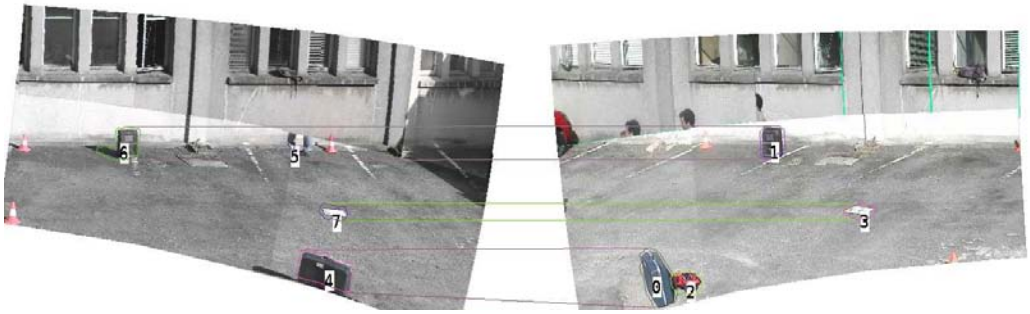


FIGURE 3.16 – Séquence *extérieur 1*. Les objets visibles dans une seule caméra ne peuvent être détectés par notre système. Ici : cas de la silhouette 2.

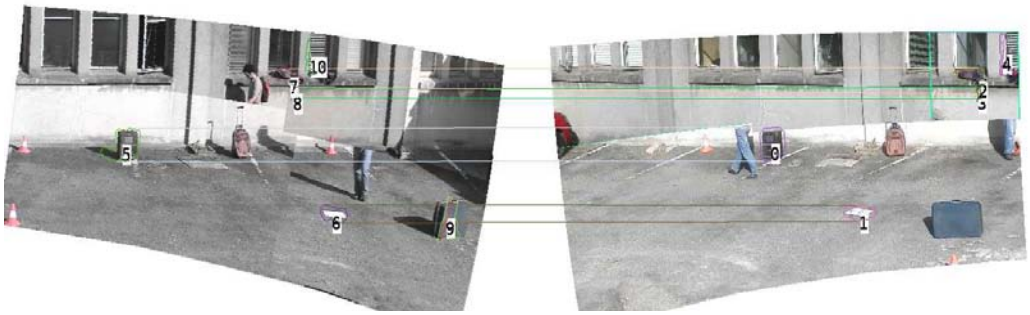


FIGURE 3.17 – Séquence *extérieur 1*. Notre approche permet de détecter des objets en hauteur, sans connaissance a priori de la scène (silhouettes 2 et 7).

le cas de caméras fixes. Comme un objet peut être visible dans plusieurs vues de la PTZ et que ces vues ne sont pas mises à jour au même instant il est possible que sur une vue mise à jour récemment sa disparition ait été constatée, bien qu'il soit toujours supposé présent sur une vue moins récente. Ce phénomène a déjà été constaté figure 3.10. Dans ce type de situation, dans les vérités terrain, nous considérons que l'objet n'est plus présent à partir du moment où sa disparition est observée dans une vue.

Les tables 3.2 et 3.3 montrent des statistiques calculées sur les différentes séquences avec une approche mono-caméra, puis pour l'appariement dans une paire de caméras. Le nombre important d'objets à détecter valide la robustesse de notre approche sur ces séquences. On peut remarquer sur la séquence *intérieur 1* la nette diminution du nombre de faux positifs par l'utilisation du seuil sur l'altitude des objets. On rappelle en effet que dans cette séquence tous les objets sont sur le plan du sol. Contraindre leurs altitudes permet donc de limiter les ambiguïtés d'appariement.

Séquence	Nombre d'objets à détecter	Vrais positifs	Faux positifs
Intérieur 1	1034	1034	591
Intérieur 2	1888	1886	1085
Extérieur 1	14693	14100	2219
Extérieur 2	18645	17861	3978

TABLE 3.2 – Statistiques des séquences en mono-caméra. Les statistiques sont calculées pour chaque caméra puis sommées.

La table 3.4 montre les scores de rappel et précision sur les différentes séquences, tout d'abord en considérant les deux caméras comme indépendantes

Séquence	Nombre d'objets à détecter	Vrais positifs	Faux positifs
Intérieur 1	456	454	63
Intérieur 1 (contraint à $+0,1m$)	456	440	39
Intérieur 2	803	759	187
Extérieur 1	6297	5847	297
Extérieur 2	6097	5526	1271

TABLE 3.3 – Statistiques des séquences calculées pour les appariements dans la paire de caméras. Les statistiques sont calculées pour chaque caméra puis sommées.

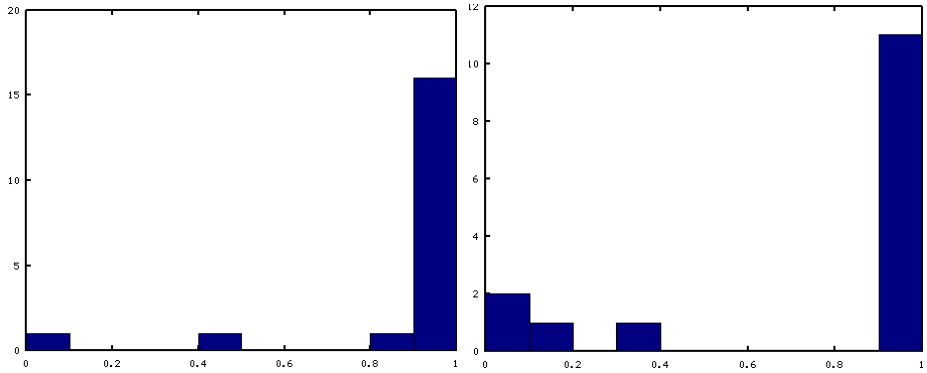
puis avec la mise en correspondances inter-caméra des silhouettes. Excepté pour la séquence *Extérieur 2* on remarque une nette augmentation de la précision au détriment d'une légère baisse du rappel. Cette baisse du rappel peut s'expliquer par le fait qu'il faut qu'un objet soit détecté dans les deux caméras pour qu'il puisse y avoir un appariement correct. Il ne peut donc y avoir une amélioration du rappel par notre approche. La non amélioration de la précision sur la séquence *extérieur 2* s'explique par la difficulté de celle ci (figure 3.12 et 3.15). Nous avions en effet constaté pendant l'analyse qualitative qu'une tendance à la sur-segmentation était la cause d'un certain nombre de mauvais appariements (figure 3.12).

Séquence	Mono-caméra		Appriement multi-caméra	
	Rappel	Précision	Rappel	Précision
Intérieur 1	0,99	0,63	0,99	0,88
Intérieur 1 (contraint à $+0,1m$)	0,99	0,63	0,99	0,92
Intérieur 2	1	0,63	0,93	0,80
Extérieur 1	0,95	0,86	0,95	0,95
Extérieur 2	0,95	0,81	0,91	0,81

TABLE 3.4 – Comparaison des statistiques calculées avec une approche mono-caméra à celles de l'approche multi-caméras.

La figure 3.18 représente les histogrammes des taux de détection des objets stationnaires des séquences *extérieur 1* et *extérieur 2*. Le taux de détection d'un objet est défini comme le nombre d'images où il a été détecté divisé par le nombre d'images où il était effectivement présent. On peut constater que sur les deux séquences trois objets n'ont pas été détectés par notre système. Cependant ceci n'est pas le résultat d'erreur d'appariement ou de détection, mais vient du fait que ces objets ne sont visibles que dans une caméra, comme c'est par exemple le cas de la silhouette 2 figure 3.16. Hormis ces trois cas tous les objets sont détectés au moins une fois. On peut remarquer que 27, parmi les 34 objets, l'ont été avec un taux de détection supérieur à $0,9$. Seulement 4 objets ont donc été détectés avec des taux compris entre 0 et $0,9$. Les non détections dans le cas des objets détectés avec un faible taux sont dues aux ambiguïtés causées par des erreurs de segmentation mono-caméra. Ces erreurs créent des silhouettes supplémentaires qui si elles sont appariées empêchent un objet d'être détecté. Le cas le plus emblématique est celui présenté en figure 3.12 et explique en partie la faible précision

constatée sur la séquence *extérieur 2*. Le deuxième objet de cette séquence à être mal détecté est la valise correspondant à la silhouette 5 de la figure 3.15. Ceci est dû à la sur-segmentation au niveau des trois bagages. Dans la séquence *extérieur 1* les deux objets dont le taux de détection est inférieur à $0,9$ sont montrés en figure 3.19 et 3.20. Dans le premier cas il s'agit d'un journal qui au bout d'un certain temps est entré dans le modèle de fond d'une des deux caméras. Il ne peut donc plus être détecté par notre système. Le second cas est celui d'une valise qui est mal segmentée, de sorte que la boite englobante de sa silhouette n'est pas incluse dans la boite englobante de la vérité terrain. Cette silhouette est donc considérée comme une fausse détection.



(a) Séquence *extérieur 1* (b) Séquence *extérieur 2*

FIGURE 3.18 – Histogramme des *taux de détection* des objets à détecter. Le *taux de détection* d'un objet est le nombre d'images où il a été détecté divisé par le nombre d'images où il est effectivement présent. La raison pour laquelle certains objets ne sont pas du tout détectés est qu'ils sont visibles dans une caméra seulement.

3.5 Conclusion

Dans ce chapitre nous avons présenté un système de détection d'objets stationnaires exploitant une paire de caméras PTZ. La stratégie retenue consiste à faire réaliser un tour de garde à chaque caméra afin de couvrir toute la surface à surveiller avec une résolution adaptée.

Dans ce contexte, nous utilisons notre méthode de détection d'objets stationnaires appliquée à chaque position du tour de garde, qui peut être considérée comme équivalente à une caméra fixe ayant une vitesse de rafraîchis-

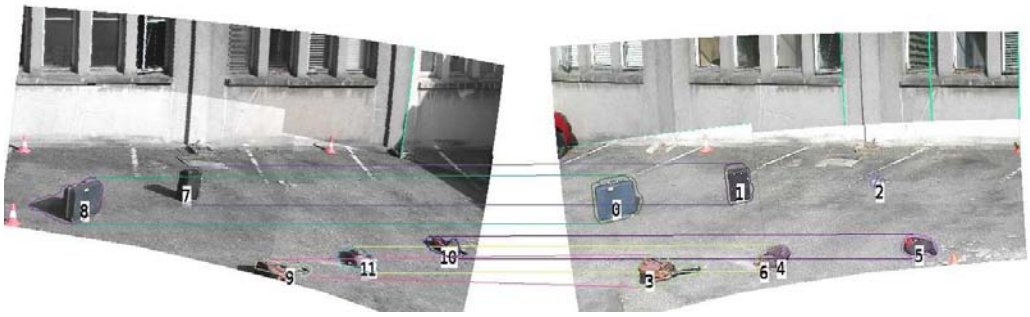


FIGURE 3.19 – Séquence *extérieur 1.* Le journal, silhouette 2, est entré par erreur dans le modèle de fond pour la caméra gauche. Il en résulte que notre système ne peut plus le détecter.

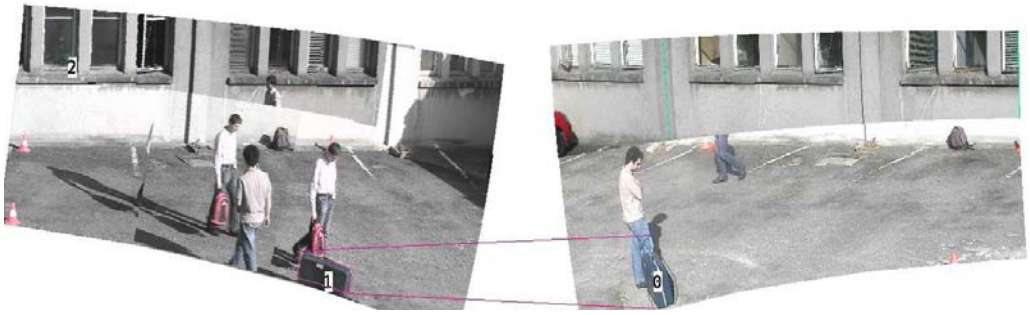


FIGURE 3.20 – Séquence *extérieur 1.* La valise est bien détectée mais sa silhouette (n° 0, caméra droite) est trop haute. D'après notre vérité terrain il s'agit donc d'une fausse détection.

sement très faible (correspondant au temps nécessaire pour effectuer un tour de garde complet et revenir à la dite position). Ensuite notre solution d'appariement d'objets inter-image est exploitée sur des images panoramiques pour assurer l'appariement des silhouettes entre les deux caméras. Notre système permet aussi de donner une estimation de la position 3D des objets, de leur taille, et de l'occultation qu'ils subissent.

Les expérimentations menées ont montré la pertinence d'un tel système ainsi que son apport en précision. Les scènes traitées sont relativement complexes puisqu'elles comprennent un important nombre d'objets ainsi que plusieurs personnes se déplaçant et générant des occultations.

Conclusion et perspectives

Travaux réalisés

Les travaux présentés dans ce manuscrit traitent de la détection d'objets stationnaires. L'objectif est d'augmenter la robustesse d'un système de détection en utilisant les spécificités des caméras PTZ et en mutualisant les informations provenant d'une paire de caméra. Pour ce faire, les caméras PTZ parcourent indépendamment un tour de garde, ce qui leur permet de surveiller une zone même très large à une résolution adaptée. En contrepartie seule une partie restreinte de la scène est visible à chaque instant.

Dans une première partie nous avons proposé deux approches de soustraction de fond, que nous avons testées et comparées à des algorithmes de la littérature sur des séquences particulièrement interessantes en terme de variations de luminosité. L'approche par grille de descripteurs SURF à été retenue et nous à permis de proposer un algorithme très simple, mais néanmoins efficace, de détection d'objets stationnaires.

Dans une seconde partie, nous avons amélioré notre détection d'objets stationnaires en définissant des critères temporels d'incompatibilité, ce qui nous a permis de segmenter dans certains cas les objets d'un même blob apparus à des instants différents. La difficulté de cette tâche réside dans la robustesse aux occultations, qui interdisent d'effectuer un simple seuillage sur l'âge des blocs de l'image. Les silhouettes ainsi trouvées sont ensuite mises en correspondance entre les deux caméras, ce qui permet de filtrer certaines fausses alarmes. L'appariement est effectué de manière à être robuste aux erreurs de segmentation. Pour cela un graphe orienté pondéré est construit, de sorte que ses cycles correspondent aux associations possibles, et que le poids des arcs soit significatif de la qualité des deux points frontières correspondants.

Finalement, dans une dernière partie nous avons appliqué notre système au cas particulier d'une paire de caméras PTZ effectuant un tour de garde. Ce contexte est particulièrement délicat puisque les scènes observées par les deux caméras sont mises à jour seulement partiellement, et de manière totalement désynchronisée. Nous avons cependant montré sur des séquences présentant des scénarios d'occultations difficiles que notre système est capable de détecter les objets stationnaires avec de très bon scores de précision et rappel.

Perspectives

En ce qui concerne la soustraction de fond on pourrait essayer différentes variations sur le descripteur SURF. Ce descripteur a en effet été développé pour être très spécifique et réidentifier avec précision deux textures identiques. Dans le contexte de la soustraction de fond on veut cependant autoriser les légères variations de texture, pour tolérer des petits changements dans une texture d'herbe ou de feuillage par exemple. On peut donc penser qu'un descripteur de plus faible dimension puisse être suffisamment discriminant pour l'application tout en réduisant le nombre de fausses alarmes.

La phase de segmentation mono-caméra est un point sensible de l'algorithme puisque une sur-segmentation augmente les possibilités d'erreurs d'appariements. Limiter au maximum la sur-segmentation permettrait donc de rendre encore plus robuste notre approche. Une première approche serait de jouer sur la pénalité d'incompatibilité pour la rendre un petit peu plus tolérante. On pourrait par exemple retarder légèrement son déclenchement. Il pourrait de plus être intéressant d'intégrer un critère d'uniformité de texture pour limiter certains cas de sur-segmentation d'objets.

Pour le moment la mise en correspondance des silhouettes n'est possible que dans une paire de caméras. Il serait cependant intéressant de généraliser l'algorithme pour pouvoir gérer un nombre plus élevé de caméras. Une approche possible serait de faire des appariements dans toutes les paires de caméras puis de regarder qu'elles sont les objets ont des tailles et positions similaires dans le monde 3D. Une autre amélioration possible concerne le gestion des objets 3D reconstruits. Pour le moment notre algorithme se contente

à chaque étape de calculer les objets 3D d'après les observations des caméras. On pourrait cependant avoir une gestion plus haut niveau des objets 3D, de sorte que par exemple si la disparition de l'objet est constatée dans une vue (mais pas nécessairement dans toutes) alors l'objet entier ne soit plus considéré comme une alarme.

Bibliographie

[1] i-lids dataset for avss 2007. http ://www.eecs.qmul.ac.uk/˜ andrea/avss2007_d.html.

[2] Lourdes Agapito, E. Hayman, et I. Reid. Self-calibration of rotating and zooming cameras. *Int. J. Comput. Vision*, 45 :107–127, November 2001.

[3] K. Alahari, P. Kohli, et P. H. S. Torr. Reduce, reuse & recycle : Efficiently solving multi-label MRFs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[4] K. Alahari, P. Kohli, et P. H. S. Torr. Dynamic hybrid algorithms for map inference in discrete mrfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(10) :1846–1857, 2010.

[5] S. Alpert, M. Galun, R. Basri, et A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007.

[6] P. Azzari, L. Di Stefano, et A. Bevilacqua. An effective real-time mosaicing algorithm apt to detect motion through background subtraction using a ptz camera. In *AVSS*, 2005.

[7] Herbert Bay, Andreas Ess, Tinne Tuytelaars, et Luc Van Gool. Surf : Speeded up robust features. In *CVIU*, 2008.

[8] A. Bayona, J.C. SanMiguel, et J.M. Martinez. Comparative evaluation of stationary foreground object detection algorithms based on background subtraction techniques. In *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pages 25 –30, 2009.

[9] A. Bayona, J.C. SanMiguel, et J.M. Martinez. Stationary foreground detection using background subtraction and temporal difference in video surveillance. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 4657 –4660, 2010.

[10] A. Bevilacqua et P. Azzari. High-quality real time motion detection using ptz cameras. In *Proc. IEEE International Conference on Video and Signal Based Surveillance AVSS '06*, pages 23–23, Nov. 2006.

[11] A. Bevilacqua, L. Di Stefano, et A. Lanza. An efficient change detection algorithm based on a statistical nonparametric camera noise model. In *Proc. International Conference on Image Processing ICIP '04*, volume 4, pages 2347–2350, 24–27 Oct. 2004.

[12] Michael D. Beynon, Daniel J. Van Hook, Michael Seibert, Alen Peacock, et Dan Dudgeon. Detecting abandoned packages in a multi-camera video surveillance system. In *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, AVSS '03, pages 221–, Washington, DC, USA, 2003. IEEE Computer Society.

[13] M. Bhargava, Chia-Chih Chen, M.S. Ryoo, et J.K. Aggarwal. Detection of abandoned objects in crowded environments. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, 2007.

[14] Kiran Bhat, Mahesh Saptharishi, et Pradeep K. Khosla. Motion detection and segmentation using image mosaics. In *ICME*, 2000.

[15] Thierry Bouwmans, Fida El Baf, et Bertrand Vachon. Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey. *Recent Patents on Computer Science*, 2008.

[16] Y. Boykov, O. Veksler, et R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11) :1222 –1239, nov 2001.

[17] Yuri Boykov et Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *Int. J. Comput. Vision*, 70 :109–131, November 2006.

[18] Yuri Boykov et Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26 :1124–1137, September 2004.

[19] Y.Y. Boykov et M.-P. Jolly. Interactive graph cuts for optimal boundary amp ; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105 –112 vol.1, 2001.

[20] Claude R. Brice et Claude L. Fennema. Scene analysis using regions. (17), Apr 1970.

[21] Thomas Brox et Joachim Weickert. Level set based image segmentation with multiple regions. In *Proceedings of 26th DAGM*, pages 415–423, 2004.

[22] Yu-Ting Chen, Chu-Song Chen, Chun-Rong Huang, et Yi-Ping Hung. Efficient hierarchical method for background subtraction. In *Pattern Recognition*, 2007.

[23] R. Cipolla, K.E. Astrom, et P.J. Giblin. Motion from the frontier of curved surfaces. In *Fifth International Conference on Computer Vision, 1995.*, pages 269 –275, 1995.

[24] Rita Cucchiara, Andrea Prati, et Roberto Vezzani. Advanced video surveillance with pan tilt zoom cameras. In *Proc. of Workshop on Visual Surveillance (VS) at ECCV*, 2006.

[25] James Davis et Xing Chen. Calibrating pan-tilt cameras in wide-area surveillance networks. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 144–, Washington, DC, USA, 2003. IEEE Computer Society.

[26] Alberto Del Bimbo, Fabrizio Dini, Andrea Grifoni, et Federico Pernici. Exploiting single view geometry in pan-tilt-zoom camera networks. In *Proc. of ECCV Int.'l Workshop on Multi-camera and Multi-modal Sensor Fusion (M2SFA2)*, 2008.

[27] A. Dervieux et F. Thomasset. A finite element method for the simulation of rayleigh-taylor instability. In *Approximation Methods for Navier-Stokes Problems, R. Rautman, ed.*, 1979.

[28] Xavier Descombes, Robert Minlos, et Elena Zhizhina. Object extraction using a stochastic birth-and-death dynamics in continuum. *J. Math. Imaging Vis.*, 33 :347–359, March 2009.

[29] Yoann Dhome, Nicolas Tronson, Antoine Vacavant, Thierry Chateau, Christophe Gabard, Yann Goyat, et Dominique Gruyer. A benchmark for background subtraction algorithms in monocular vision : a comparative study. In *IPTA*, 2010.

[30] Richard O. Duda et Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, 1973.

[31] Ahmed Elgammal, David Harwood, et Larry Davis. Non-parametric model for background subtraction. In *FRAME-RATE WORKSHOP, IEEE*, pages 751–767, 2000.

[32] Shireen Elhabian, Khaled, et Sumaya. Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art. *Recent Patents on Computer Science*, 1 :32–34, 2008.

[33] Pedro F. Felzenszwalb et Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2) :167–181, 2004.

[34] F. Fleuret, J. Berclaz, R. Lengagne, et P. Fua. Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2) :267 –282, feb. 2008.

[35] L. R. Ford et D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8 :399–404, 1956.

[36] J. Freixenet, X. Mu noz, D. Raba, J. MartÃ, et X. CufÃ. Yet another survey on image segmentation : Region and boundary information integration. In *In ECCV*, pages 408–422, 2002.

[37] Alfredo Gardel, José Luis Lázaro, et J.M. Lavest. Influence of mechanical errors in a zoom camera. *Image Analysis and Stereology*, 2003.

[38] Stuart Geman et Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6) :721 –741, nov. 1984.

[39] D. M. Greig, B. T. Porteous, et A. H. Seheult. Exact Maximum A Posteriori Estimation for Binary Images. In *Journal International Journal of Computer Vision*, 1989.

[40] Constant Guillot, Maxime Taron, Patrick Sayd, Quoc-Cuong Pham, Christophe Tilmant, et Jean-Marc Lavest. Background subtraction for ptz cameras performing a guard tour and application to cameras with very low frame rate. In *ACCV, Visual Surveillance workshop*, 2010.

[41] S. Guler, J.A. Silverstein, et I.H. Pushee. Stationary objects in multiple object tracking. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, 2007.

[42] Bohyung Han et Ramesh Jain. Real-time subspace-based background modeling using multi-channel data. In *Proceedings of the 3rd international conference on Advances in visual computing - Volume Part II*, ISVC'07, pages 162–172, Berlin, Heidelberg, 2007. Springer-Verlag.

[43] C. Harris et M. Stephens. A Combined Corner and Edge Detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[44] Justin Hart, Brian Scassellati, et Steven W. Zucker. Epipolar geometry for humanoid robotic heads. *ICVW*, 2008.

[45] R. I. Hartley et A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN : 0521540518, second edition, 2004.

[46] Marko HeikkilÃ¤ et Matti PietikÃ¤inen. A texture-based method for modeling the background and detecting moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 :657–662, 2006.

[47] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4) :629–642, 1987.

[48] Ankur Jain, Dan Kopell, Kyle Kakligian, et Yuan fang Wang. Using stationary dynamic camera assemblies for wide-area video surveillance and selective attention. In *In IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[49] R.C. Jain et H.H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. In *PAMI*, 1979.

[50] R. Jonker et A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4) :325–340, 1987.

[51] Imran Junejo et Hassan Foroosh. Optimizing PTZ camera calibration from two images. *Machine Vision and Applications*, pages 1–15, 2011.

[52] Michael Kass, Andrew Witkin, et Demetri Terzopoulos. Snakes : Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4) :321–331, 1988.

[53] S.M. Khan et M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2009.

[54] Kyungnam Kim, T. H. Chalidabhongse, D. Harwood, et L. Davis. Background modeling and subtraction by codebook construction. In *ICIP*, 2004.

[55] Ross Kindermann et J. L. Snell. *Markov Random Fields and Their Applications (Contemporary Mathematics)*. Amer Mathematical Society, 1980.

[56] P. Kohli et P.H.S. Torr. Efficiently solving dynamic markov random fields using graph cuts. In *ICCV*, pages II : 922–929, 2005.

[57] V. Kolmogorov et R. Zabin. What energy functions can be minimized via graph cuts ? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2) :147 –159, feb. 2004.

[58] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28 :1568–1583, October 2006.

[59] Ivan Kovtun. Partial optimal labeling search for a np-hard subclass of (max,+) problems. In *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 402–409. Springer Berlin / Heidelberg, 2003.

[60] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2 :83–97, 1955.

[61] Liyuan Li et M.K.H. Leung. Integrating intensity and texture differences for robust change detection. *Image Processing, IEEE Transactions on*, 11(2) :105 –112, feb 2002.

[62] Mengxiang Li et J.-M. Lavest. Some aspects of zoom lens camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(11), nov 1996.

[63] Ying li Tian, Rogerio Feris, et Arun Hampapur. Real-time detection of abandoned and removed objects in complex environments. In *Eight International Workshop on Visual Surveillance*, 2008.

[64] Ying li Tian, Max Lu, et Arun Hampapur. Robust and efficient foreground analysis for real-time video surveillance. In *Computer Vision and Pattern Recognition*, pages 1182–1187, 2005.

[65] Huei-Hung Liao, Jing-Ying Chang, et Liang-Gee Chen. A localized approach to abandoned luggage detection with foreground-mask samplingxo. In *Advanced Video and Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on*, 2008.

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

[67] B. D. Lucas et T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[68] Fengjun Lv, Xuefeng Song, Bo Wu, Vivek Kumar, et Singh Ramakant Nevatia. Left luggage detection using bayesian inference. In *In PETS*, 2006.

[69] Fengjun Lv, Xuefeng Song, Bo Wu, Vivek Kumar, et Singh Ramakant Nevatia. Left luggage detection using bayesian inference. In *In PETS*, 2006.

[70] R. Malladi, J.A. Sethian, et B.C. Vemuri. Shape modeling with front propagation : a level set approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(2) :158 –175, feb 1995.

[71] R. Mathew, Zhenghua Yu, et Jian Zhang. Detecting new stable objects in surveillance video. In *Multimedia Signal Processing, 2005 IEEE 7th Workshop on*, pages 1 –4, 30 2005-nov. 2 2005.

[72] C. Micheloni, B. Rinner, et G.L. Foresti. Video analysis in pan-tilt-zoom camera networks. *Signal Processing Magazine, IEEE*, 27(5) :78 –90, sep. 2010.

[73] R. Miezianko et D. Pokrajac. Localization of detected objects in multi-camera network. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2376 –2379, oct. 2008.

[74] Krystian Mikolajczyk et Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10) :1615–1630, 2005.

[75] J. Mille, R. Bone, P. Makris, et H. Cardot. Greedy algorithm and physics-based method for active contours and surfaces : A comparative study. In *Image Processing, 2006 IEEE International Conference on*, pages 1645 –1648, oct. 2006.

[76] Anurag Mittal et Nikos Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *CVPR (2)*, pages 302–309, 2004.

[77] John Moussouris. Gibbs and markov random systems with constraints. *Journal of Statistical Physics*, 10 :11–33, 1974. 10.1007/BF01011714.

[78] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transaction on Pattern Analysis and Maching Intelligence*, 26 :756–777, June 2004.

[79] Philippe Noriega, Benedicte Bascle, et Olivier Bernier. Local kernel color histograms for background suntraction. In *VISAPP*, 2006.

[80] Philippe Noriega et Olivier Bernier. Real time illumination invariant background subtraction using local kernel histograms. In *BMVC*, 2006.

[81] R. Ohlander, K. Price, et D. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8(3) :313–333, 1978.

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

[83] Stanley Osher et James A. Sethian. Fronts propagating with curvature dependent speed : Algorithms based on hamilton-jacobi formulations. *JOURNAL OF COMPUTATIONAL PHYSICS*, 79(1) :12–49, 1988.

[84] Michael Péchaud. tutorial : Introduction aux graphcut en vision par ordinateur, 2006.

[85] M. Piccardi. Background subtraction techniques : a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099 – 3104 vol.4, oct. 2004.

[86] Fatih Porikli. Achieving real-time object detection and tracking under extreme conditions. *Journal of Real-Time Image Processing*, 2006.

[87] Fatih Porikli, Yuri Ivanov, et Tetsuji Haga. Robust abandoned object detection using dual foregrounds. *EURASIP J. Adv. Signal Process*, 2008, January 2008.

[88] Lionel Robinault, Stéphane Bres, et Serge Miguet. Real time foreground object detection using ptz camera. In *VISAPP*, 2009.

[89] Carsten Rother, Vladimir Kolmogorov, et Andrew Blake. "grabcut" : interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 309–314, New York, NY, USA, 2004. ACM.

[90] J. Rymel, J. Renno, D. Greenhill, J. Orwell, et G.A. Jones. Adaptive eigen-backgrounds for object detection. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 3, pages 1847 – 1850 Vol. 3, oct. 2004.

[91] J.C. San Miguel et J.M. Martinez. Robust unattended and stolen object detection by fusing simple algorithms. In *Advanced Video and*

Signal Based Surveillance, 2008. AVSS '08. IEEE Fifth International Conference on, 2008.

[92] Karthik Sankaranarayanan et James W. Davis. Ptz camera modeling and panoramic view generation via focal plane mapping. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part II*, ACCV'10, pages 580–593, Berlin, Heidelberg, 2011. Springer-Verlag.

[93] Jianbo Shi et Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8) :888–905, 2000.

[94] Sudipta Sinha, , Sudipta N. Sinha, et Marc Pollefeys. Towards calibrating a pan-tilt-zoom camera network. In *In Workshop on Omnidirectional Vision and Camera Networks at ECCV*, 2004.

[95] C. Stauffer et W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[96] Richard Szeliski. Computer vision : Algorithms and applications, 2010.

[97] K. Toyama, J. Krumm, B. Brumitt, et B. Meyers. Wallflower : principles and practice of background maintenance. In *Proc. Seventh IEEE International Conference on Computer Vision The*, volume 1, pages 255–261, 20–27 Sept. 1999.

[98] Miroslav Trajkovic. Interactive Calibration of a Pan-Tilt-Zoom (PTZ) Camera for Surveillance Applications. In *Asian Conference Computer Vision*, 2002.

[99] Rémi Trichet et Bernard Mérialdo. Keypoints labeling for backgroung substraction in tracking applications. In *ICME, IEEE International Conference on Multimedia Expo, Hannover, Germany*, 06 2008.

[100] A. Utasi et Cs. Benedek. A 3-d marked point process model for multi-view people detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[101] Akos Utasi et Benedek Csaba. Multi-camera people localization and height estimation using multiple birth and death dynamics. In *ACCV workshop on Visual Surveillance*, 2010.

[102] B. Valentine, S. Apewokin, L. Wills, S. Wills, et A. Gentile. Midground object detection in real world video scenes. In *Proceedings of the 2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 517–522, Washington, DC, USA, 2007. IEEE Computer Society.

[103] P.L. Venetianer, Z. Zhang, W. Yin, et A.J. Lipton. Stationary target detection using the objectvideo surveillance system. In *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, 2007.

[104] J.A. Vijverberg, M.J.H. Loomans, C.J. Koeleman, et P.H.N. de With. Global illumination compensation for background subtraction using gaussian-based background difference modeling. In *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pages 448 –453, sept. 2009.

[105] Dingrui Wan et Jie Zhou. Stereo vision using two ptz cameras. *Computer Vision and Image Understanding*, 112 :184–194, November 2008.

[106] Dingrui Wan et Jie Zhou. Self-calibration of spherical rectification for a ptz-stereo system. *Image and Vision Computing*, 28 :367–375, March 2010.

[107] Matthew Paul Wand et Chris Jones. *Kernel Smoothing*. Crc Press, 1995.

[108] C. Wren, A. Azarbayejani, T. Darrell, et A. Pentland. Pfinder : real-time tracking of the human body. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 51 –56, oct 1996.

[109] Zhifei Xu, Pengfei Shi, et Irene Y. H. Gu. An eigenbackground subtraction method using recursive error compensation. In *PCM'06*, pages 779–787, 2006.

[110] Jian Yao et Jean-Marc Odobez. Multi-layer background subtraction based on color and texture. In *CVPR 2007 Workshop on Visual Surveillance (VS2007)*, 6 2007.

[111] Qiang Zhu, S. Avidan, et Kwang-Ting Cheng. Learning a sparse, corner-based representation for time-varying background modelling. In *ICCV*, volume 1, pages 678 – 685, oct. 2005.

[112] Song Chun Zhu et Alan Yuille. Region competition : Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 :884–900, 1995.