

# Techniques de vision par ordinateur pour la vidéo surveillance.

X. Desurmont<sup>a</sup>, C. Chaudy<sup>a</sup>, C. Parisot<sup>a</sup>, J.F. Delaigle<sup>a</sup> et B. Macq<sup>b</sup>

{desurmont, chaudy, parisot, delaigle}@multitel.be

<sup>a</sup>, Multitel A.S.B.L, Belgique

<sup>b</sup>Université catholique de Louvain, Belgique

## Résumé

La sécurité vidéo devient de plus en plus importante de nos jours ainsi que le nombre de systèmes installés en atteste. Il y a de nombreux défis d'applications commerciales pour surveiller le trafic routier ou même la sécurité des personnes. Le travail présenté est issu de motivations tant au niveau recherche qu'au niveau industriel. En effet, nos buts sont en premier lieu de répondre aux contraintes fortes des systèmes de surveillance industrielle, c'est à dire le fonctionnement temps réel, être distribués, génériques et robustes, mais c'est aussi d'avoir une plate-forme de développement qui permet aux chercheurs d'imaginer et de tester facilement de nouveaux algorithmes grâce à une modularité et un paramétrage facile. Ce papier sera axé sur les modules d'analyses d'images. Nous considérerons les différents types d'entrées, les étapes du traitement et les modèles d'algorithmes.

**Mots clefs:** multi-caméra, architecture distribuée, temps-réel, vision par ordinateur, suivi, vidéosurveillance intelligente.

## 1. Introduction.

La vidéosurveillance est un marché important. Il y a un besoin pour des systèmes génériques qui peuvent s'interfacer dans des réseaux de caméra (CCTV) afin d'augmenter l'intelligence et de permettre un traitement automatique. Des exemples de défis d'applications [1] sont la surveillance de stations de métro [2] ou bien la détection d'embouteillage, la détection de délits, ainsi que l'accès intelligent aux contenus. Les exigences pour ces systèmes sont d'être connectés en réseaux, multi-caméras, modulaires, facile d'utilisation ; les modules de visions doivent pouvoir être montés à chaud et le système doit être robuste et fiable dans son ensemble. Le travail présenté ici se base sur des motivations de recherche mais aussi industrielles. Un état de l'art succinct se trouve dans [3, 4, 5]. Dans cet article, nous présentons un système de vidéosurveillance intelligent, générique, flexible, temps réel et robuste.

Le papier est organisé comme suit: la partie 2 décrit le système dans sa globalité et ses caractéristiques principales, la partie 3 regarde plus spécifiquement chaque module, la partie 4 est réservée au module d'analyse d'image. Enfin la partie 5 conclue et ouvre sur des perspectives futures.

## 2. Vue d'ensemble du système.

Le système de vidéosurveillance présenté dans ce papier repose sur une architecture de type réseau. Ce type de système peut être déployé dans un bâtiment ou même utiliser directement les connexions de réseaux de données existantes. En bref, le système est composé d'ordinateurs reliés entre eux. Les différentes caméras sont connectées soit sur une carte d'acquisition sur un PC, soit directement sur le réseau dans le cas de caméras IP. Une interface homme-machine et une unité de stockage sont aussi connectées au système. L'avantage principal d'une telle architecture est sa flexibilité. L'architecture logique a été conçue d'une manière modulaire pour permettre une juste allocation des ressources. Les besoins futurs en puissance de calcul seront simplement résolus en ajoutant un PC à la grappe. Les composants essentiels de cette architecture physique sont présentés dans la figure 1. Dans cette mise en place, chaque module logiciel est dédié à une tâche précise, par exemple la compression vidéo ou bien la gestion du réseau. Les modules de traitements sont distribués sur les unités PC selon un paramétrage de la configuration. Un opérateur peut définir l'architecture et l'ordonnancement du système distribué et de plus, il peut personnaliser les actions des différents modules. Une tâche maîtresse du management communique ensuite la configuration à toutes les unités PC. Un changement de la distribution des tâches entre les unités demandera simplement un changement dans le fichier de configuration (pas besoin de recompiler.)



Figure. 1: Eléments principaux de l'architecture.

La robustesse du système dans son ensemble repose sur l'architecture logique. Elle supervise les divers problèmes qui peuvent arriver : un paquet perdu dans le réseau, un disque dur en arrêt, etc.

## 2.1 Gestion des données

La gestion des communications entre les modules prend en compte l'accès concurrent aux données pour plusieurs producteurs et consommateurs. Cela optimise les besoins en mémoire en évitant les re-copies de données. Le gestionnaire pilote les communications réseau (TCP/IP) pour les multiples instances d'une donnée (avec compression si nécessaire), la connexion dynamique et l'accès aux flux, la bufferisation pour absorber les pics de traitement, la conversion implicite des données (ex : d'une image RGB vers YUV), la propagation des besoins des consommateurs vers les producteurs (ex : si plus aucun module n'a besoin des résultats de la segmentation, alors le module de segmentation en sera avisé et ne le produira plus), l'étude (monitoring) dynamique des flux (débits, performances), l'envoi (dispatching) des données par priorité pour chaque module spécifique (aide à rendre les processus temps-réel en baissant la latence). La figure 2 montre un exemple de flux de données (contrôle, image source, descripteurs dynamiques de la scène, événements) et un traitement typique. L'acquisition des images est faite à chaque image (25 fps), mais le suivi (ainsi que l'interprétation) ne se fait qu'une image sur deux (12,5 fps).

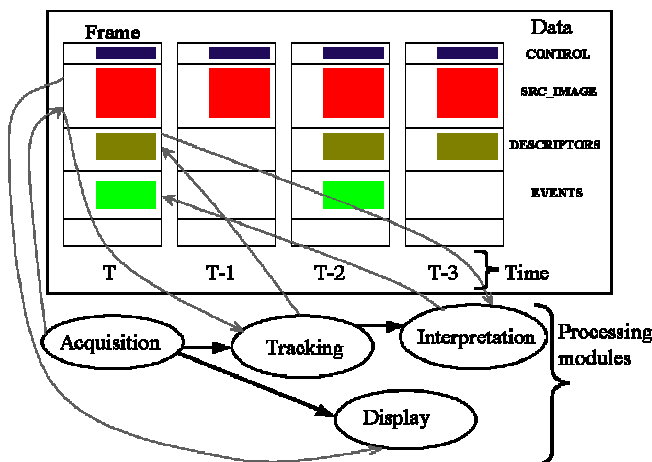


Figure 2: Exemple de gestion de données.

## 2.2 Manipulation des données Asynchrones

Les applications de vidéosurveillance ont besoin de fonctionner en temps-réel car les aspects de sécurité requièrent un temps de réaction minimum. Par conséquent, la vidéosurveillance automatique impose de faibles délais et des contraintes temporelles pour le traitement. Les systèmes classiques temps-réel sont souvent basés sur une boucle périodique interne (ex : une image est acquise toutes les 40ms et doit être traitée par un processus en moins de 40 ms dans tous les cas, sinon le système s'arrête). Si jamais le processus prend plus de 40 ms pour l'un des étages du pipeline (s'il n'y a pas de pipeline, on considère le système comme un pipeline à un seul niveau), cela entraîne un délai qui peut alors grandir à chaque nouvelle image. Typiquement, ce délai est limité à un maximum (ex : 200 ms), s'il excède cette limite alors le système faillit (ex : le buffer de mémoire est plein, il y a donc perte de données, de traitement de ces données, les fonctions

d'intégration ou dérivation risquent alors de faire de mauvais calculs). Nous outrepassons ce genre de problèmes en ajoutant à chaque observation (image) un horodatage (timestamp), et puis en considérant ce temps à chaque traitement (ex : pour l'adaptation de l'image de référence, vitesse des véhicules, etc.). Ainsi si les ressources de traitement nécessaires sont supérieures aux ressources disponibles du système, certaines informations sont ignorées sans déranger l'intégrité du système.

## 3. Description des composants du système.

Les divers modules de la partie logicielle sont expliqués ci-après. Nous décrivons successivement le module d'acquisition, le module de codage-décodage vidéo (codec), le module de réseau, le module de stockage et le module d'interface homme/machine. La section 4 dépeindra le module d'analyse d'image.

### 3.1. Acquisition

Nous sommes actuellement capables de manipuler plusieurs protocoles pour l'acquisition d'image : IP (JPEG et MJPEG), IEEE1394 (raw et DV), wireless (analogique, Wifi) et composite (PAL, NTSC, SECAM). Un horodatage est attaché à chaque image au moment de l'acquisition. Comme nous l'avons signalé, cette information est utilisée dans les étapes de traitements en aval. Pour les replays de vidéos nous utilisons la bibliothèque logicielle Ffmpeg [6].

### 3.2. Codage-décodage des images

Le but du processus de codage est d'avoir un bon compromis entre le taux de compression et l'occupation de la bande passante. Nous proposons ici le codec MPEG4 SP puisqu'il surpasse le codage MJPEG classique. L'inconvénient principal de MJPEG est qu'il n'utilise pas de redondance temporelle pour augmenter le facteur de compression. De notre expérience, le MPEG4 SP a un facteur de compression 10 fois meilleur à celui du MJPEG pour la même qualité d'image. En effet les scènes de surveillance vidéo sont tout à fait statiques quand les caméras sont fixées. Les méthodes de compression supprimant la redondance temporelle, comme MPEG4 SP, sont donc plus efficaces.

Par contre, nous ne pouvons pas avoir directement accès aux images indépendamment. Nous devons re-synchroniser le flux sur une I-image (image intra) chaque fois qu'une erreur de transmission réseau arrive. De plus, pour limiter le retard entre le temps de codage et le temps d'exposition, nous ne codons pas d'images comme des B-images (des images codées bidirectionnelles). Cette technique nous permet de transmettre jusqu'à 20 CIF (352 x 288) flux vidéo à 25 fps sur un réseau Ethernet 100baseT typique.

### 3.3. Le réseau

Un système distribué implique une utilisation efficace de bande passante. Nous avons vu que les divers modules utilisant l'entrée vidéo peuvent être dispatchés sur plusieurs ordinateurs. Par exemple, nous pouvons avoir l'acquisition sur

l'ordinateur 1, le stockage sur l'ordinateur 2 et l'affichage sur l'ordinateur 3. Nous avons choisi une technique de multicast pour résoudre le problème d'occupation de bande passante. Chaque source vidéo a un canal de multicast associé. Ce canal de multicast se base sur une connexion UDP. L'UDP n'offrant pas de qualité de service (QoS), nous avons développé un protocole qui peut détecter quand un échec de transmission arrive. Nous pouvons garantir de petits délais parce que la charge de réseau est contrôlée pour éviter la formation d'une queue d'envoi de paquet. Ce délai est assez petit pour être imperceptible pour l'utilisateur.

### 3.4. Stockage

Le module de stockage doit traiter l'énorme quantité de données produites par les différents modules et capteurs. Il doit permettre de stocker les images 24 heures sur 24. Ce module a deux niveaux : le niveau 0 est un processus de stockage classique avec la technologie MPEG4. Ce niveau stocke un flux vidéo CIF à 25 fps pendant sept jours sur un disque dur de 60 Gb. Lors d'une opération de maintenance, un opérateur peut améliorer le taux de compression en utilisant une deuxième passe de l'encodeur pour passer d'une compression à bande passante constante à qualité constante (on baisse la bande passante dans les périodes où cela permet de garder une qualité acceptable). Le niveau 1 est un processus de stockage intelligent. Il stocke seulement des événements intéressants que l'utilisateur a définis. Ce niveau économise un espace de données conséquent. De plus, il permet une recherche rapide pour récupérer une séquence stockée grâce à l'indexation.

## 4. Module d'analyse d'image.

L'interprétation de haut niveau d'événements dans la scène requiert un traitement d'image bas niveau et un suivi des objets se déplaçant dans la scène. Il est nécessaire pour produire une représentation des objets dans la scène. Pour notre système, l'architecture de la partie de vision est divisée en trois niveaux principaux de traitement et est présentée à la figure 4: Le niveau image (l'acquisition et filtrage des images, l'évaluation de l'image de référence et la segmentation), le niveau des régions mobiles (la description, le filtrage des régions mobiles, la mise en correspondance temporelle, la description et le filtrage), le niveau des événements (analyse du suivi, machine à états finis, l'évaluation des performances).

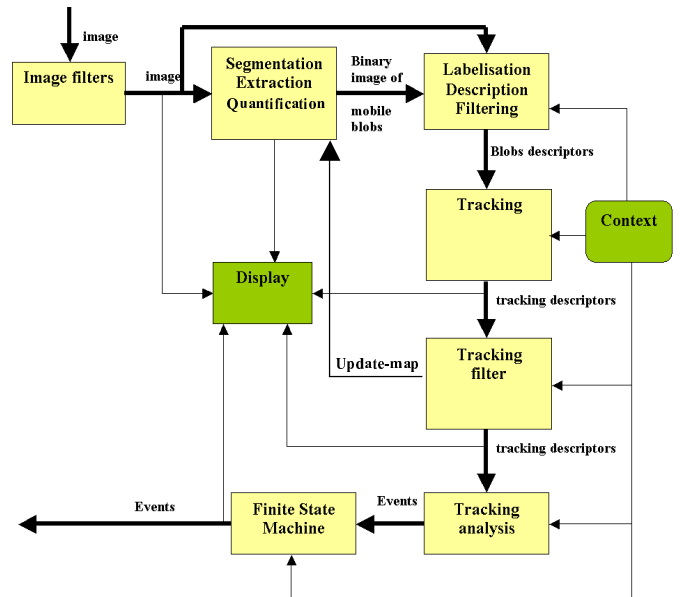


Figure 4: architecture des différents composants du système de vision.

### 4.1 Calibrage et contexte 2D et 3D

Les applications avec caméras multiples ou en faisant la 3D avec connaissance du plan du sol requièrent un calibrage des caméras. Nous avons développé un outil pour le calibrage manuel (figure 5) avec la même procédure que dans [7]. Les déformations radiales sont gérées via à la bibliothèque logicielle OpenCV [8]. Pour une caméra fixe, un contexte 2D pourrait être défini par des utilisateurs pour identifier des secteurs dans l'image comme les régions d'entrée/sortie, les zones à ignorer, etc. Un contexte 3D (figure 6) peut être paramétré pour décrire la scène 3D (ex: le sol, les murs et les objets présents dans la scène au début de la séquence). Troisièmement, il permet à un utilisateur de définir des informations contextuelles globales sur la scène (des informations correspondant à beaucoup de caméras voyant la même scène) ou spécifiques. Ces informations contextuelles sont représentées au moyen des polygones 2D sur l'image, chacun d'entre eux ayant une liste d'attributs : IN\_OUT, NOISY, OCCLUSION, AREA\_OF\_INTEREST zone, etc. Ce type d'informations est utilisé par module d'analyse d'image pour aider le processus d'identification de scénario et la gestion des alarmes.

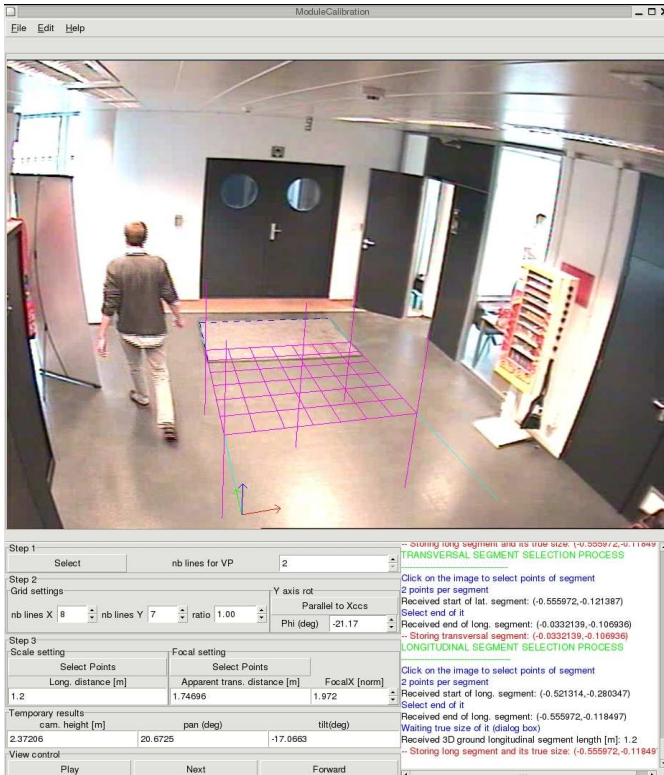
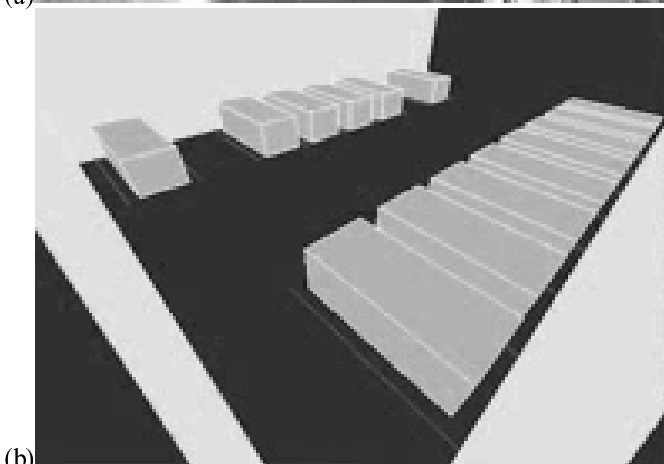


Figure 5: outil interactif pour le calibrage manuel.



(a)



(b)

Figure 6: (a) vue initiale d'une camera fixe d'un parking, (b) contexte 3D Initial du même parking.

## 4.2 Filtrage des images

Après l'acquisition, l'image actuelle est filtrée pour diminuer le bruit spatial de haute fréquence. Typiquement l'image est convoluée via un filtre linéaire avec un noyau gaussien. Mais, dans notre cas, nous utilisons un filtre de Shen exponentiel à deux passes optimisé pour la vitesse (plus rapide que la convolution). L'image est ensuite diminuée en taille pour réduire les besoins de ressource du processus de segmentation.

## 4.3 Re-calibrage automatique

Dans le cas de caméra se déplaçant en raison de vibrations légères (par exemple le vent) ou de type pan-tilt-zoom l'image est changée et doit subir une transformation géométrique homographique 2D inverse [9] pour être traitée de manière correcte par rapport à l'image de référence (figure 7). Au stade actuel de développement, le modèle de déplacement n'autorise que la translation visuelle. (pan-tilt avec un petit angle de vue).

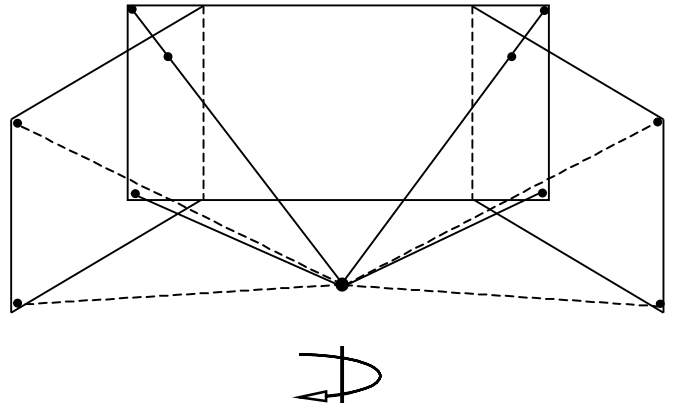


Figure 7: Les images de gauche et droite sont projetées sur l'image du milieu (vue de la scène).

## 4.4 Segmentation

L'approche du bas vers le haut (bottom-up) habituelle pour segmenter les objets mobiles est l'évaluation du fond de la scène et l'extraction du premier plan de la scène (approche dite de background subtraction) [10]. Les problèmes typiques liés à cette technique sont les changements d'illumination, les objets fantômes, des vibrations de caméra, le déplacement de branches d'arbres, etc. Nous pouvons utiliser beaucoup de modèles d'image de fond pour représenter la scène de référence comme ils sont intégrés au système (filtre récursif temporel passe-bas, filtre médian, modèle de moyenne et variance, mélange de gaussiennes, vector quantisation [11]). Dans la littérature, le modèle d'image de fond est mis à jour pour prendre en considération des petits changements d'illumination par un filtrage passe-bas récursif:

$$B_t = \alpha I_t + (1 - \alpha) B_{t-1} \quad (1)$$

L'équation (1) est accomplie pour chacun des pixels (x,y) de l'image.  $\alpha$  est un paramètre de l'algorithme compris entre 0 et 1,  $B_{t-1}$  la valeur précédente de l'image de fond,  $I_t$  est la valeur

de pixel d'image actuelle et  $B_t$  est la valeur actuelle mise à jour de l'image de fond. Le premier plan est alors typiquement déduit par un seuil  $T$  de la différence de  $B_t$  et  $I_t$ . Un pixel est dans le premier plan quand (2) est vrai :

$$|B_t - I_t| > T \quad (2)$$

Ce modèle simple peut manipuler les variations de base petites ou lentes du fond de la scène mais n'est pas assez efficace dans les cas plus complexes de bruits comme le mouvement des branches d'un arbre ou le scintillement d'un moniteur. L'algorithme de segmentation conçue généralise l'approche habituelle et la rend plus rationnelle en séparant celui-ci en deux processus distincts: estimation du premier plan et mise à jour de la scène de fond. Nous ajoutons aussi la possibilité de prévoir l'état de fond à un moment donné et aussi la possibilité de réactions (feedback) pour mettre à jour sélectivement des parties de la scène de référence (notamment grâce à un retour du suivi) [12]. La figure 8 explique pourquoi il est nécessaire d'avoir des étapes différentes pour mettre à jour et évaluer la scène de référence. Au temps  $t1$ , le modèle a l'historique de 0 à  $t1$  ( $0:t1$ ). Ensuite il y a une mise à jour au temps  $t2$ . On peut demander une évaluation de la scène de référence au temps  $t2$  et la comparer à la valeur du pixel actuellement pour savoir s'il doit être classifié comme un premier plan ou un pixel de fond. Dans cet exemple (figure 8) la scène de référence est bimodale et la lumière augmente le long du temps  $t1$  ( $0:t1$ ). Par ailleurs, grâce à l'implémentation logicielle modulaire, la segmentation est entièrement configurable, on peut choisir dynamiquement, entre tous les types de modèles de fond et des paramètres.

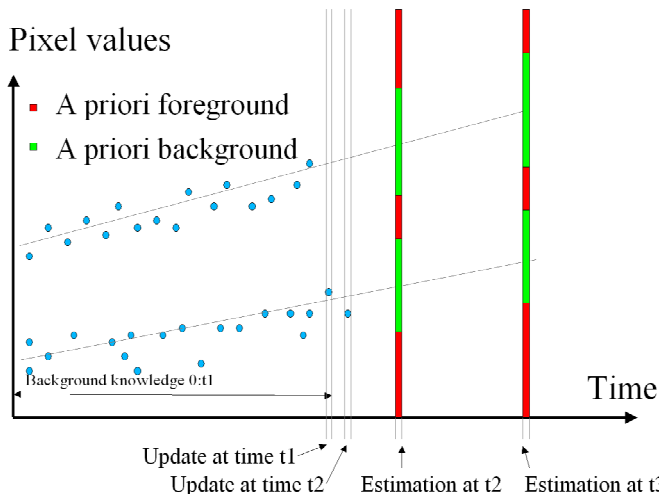


Figure 8: Représentation de l'historique d'un pixel et d'un modèle de scène de référence. Ici, la scène de référence est bimodale et la lumière augmente le long du temps  $t1$  ( $0:t1$ ).

Ci-après nous expliquons les 4 étapes de la segmentation. Soient  $t1 < t2$  deux moments d'acquisition d'images à la suite. Le processus est divisé en plusieurs parties (figure 9 et 10) :

1) On considère une carte de croyance (la probabilité) a priori des pixels pour faire partie de l'avant scène. La carte est calculée à partir de l'image actuelle acquise au temps  $t2$  et du modèle de fond mis à jour au temps  $t1$ . Les pixels sont

décidés comme étant priori de l'avant scène si la probabilité est supérieure à 0.5. A ce stade, il n'y a pas de connaissance de voisinage.

2) Si le pixel est à l'intérieur d'un contour fermé de pixel avec une probabilité  $> 0.5$ , la règle de voisinage le considérera comme le pixel de premier plan. Une deuxième règle de voisinage est appliquée: Un pixel ne peut pas être le premier plan s'il y a aucun chemin de pixel de premier plan a priori connecté jusqu'à un pixel de probabilité  $> 0.8$ . Ces deux règles permettent un phénomène d'hystérésis et diminuent le bruit dans le premier plan.

3) Deux étapes de décision sont ensuite faites à deux phases différentes du processus pour filtrer les objets du premier plan : après la description (objet dans une zone à ignorer, objet trop petit, objet fantôme, etc.) mais aussi après le suivi (objet intégré). Nous ne considérons pas ici ces processus (voir la subdivision 4.5, 4.6, 4.7 et 4.8). Après eux, quelques objets de premier plan de 2) sont ignorés ou quelques objets non détectés sont ajoutés à la structure des données qui contient la carte de premier plan. A ce stade, La carte qui contient les pixels de l'avant plan est appelée update-map.

4) Le modèle de fond est alors mis à jour avec l'image au temps  $t2$  pour les régions de la scène que la carte de mise à jour (update-map) définit comme l'étant du fond de la scène.

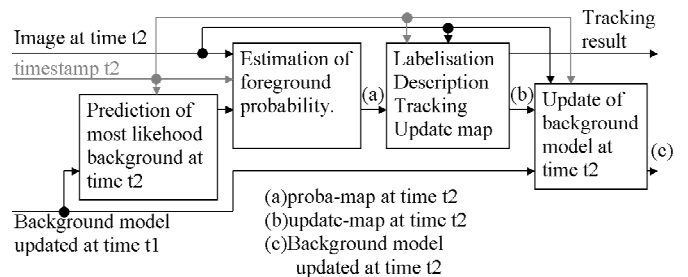


Figure 9: Architecture du processus de segmentation inséré dans le système global.

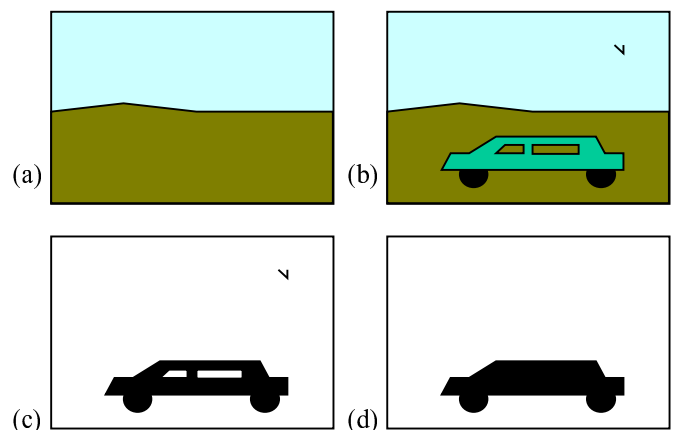


Figure 10: (a) représentation du modèle de fond de scène (appelé aussi image de référence) au temps  $t1$  (b) image de la scène au temps  $t2$ , (c) image de l'avant scène à l'étape 2, (d) avant scène au à l'étape 3 (les petits objets et les trous ont été enlevés).

Dans les sous parties 4.5, 4.6, 4.7 et 4.8, nous regarderons les processus qui se déroulent entre 1) et 4).

#### 4.5 Description et filtrage des régions mobiles

La description et le filtrage de régions font l'interface entre l'extraction du premier plan et le suivi. Le processus de description traduit des données vidéo dans une représentation symbolique (c'est-à-dire des descripteurs). Le but est de réduire la quantité d'informations à ce qui est nécessaire et suffisant pour le module de suivi. Le processus de description calcule à partir de l'image et des résultats de segmentation au temps  $t$ , les  $k$  différentes caractéristiques observées d'une région  $i$  : La position 2D dans l'image, la position 3D dans la scène, la boîte englobante, la couleur en RGB ou YUV, l'axe d'inertie de la forme 2D de la région, les points extrêmes de la forme, la probabilité d'une région d'être fantôme, etc. (voir figure 11) Suit après un autre processus de filtrage pour enlever les petites régions, ainsi que celles dans un secteur de l'image non considéré, etc. D'autres descripteurs visuels pourraient être aussi intégrés à la demande pour des applications spécifiques utilisant des modèles de véhicule ou d'humain en 3D.

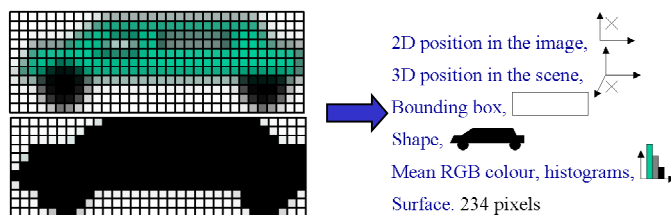


Figure 11 : Représentation de la description des régions mobiles.

#### 4.6 Algorithme de suivi

Comme les autres modules, la partie « suivi » du système est flexible et entièrement paramétrable dynamiquement. L'initialisation (set-up) devrait être faite selon un compromis entre les ressources de calcul et les besoins en robustesse. Le suivi est divisé en quatre étapes (figure 12) qui suivent une approche du bas vers le haut niveau :

- la prédiction,
- la matrice de coût,
- la/les décision(s) de mise en correspondance,
- la mise à jour des pistes.

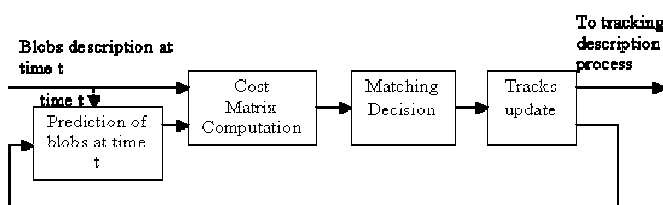


Figure 12. Architecture basique du suivi.

**Prédiction:** le processus d'évaluation est très simple. Il prévoit les caractéristiques de la région (la position, la couleur, la taille, etc.). Il est intégré comme un estimateur récursif pour pouvoir manipuler des cœurs d'estimateurs

comme le filtre Kalman [13], Euler explicite, etc. L'évaluation est alors prise comme le maximum *a posteriori* de la probabilité (MAP).

$$\hat{f}_{k,i}^t = \arg \max_{f_{k,i}^t} p(f_{k,i}^t | f_{k,i}^{1:t-1}) \quad (3)$$

**Calcul de matrice de coût:** le but de cette matrice est de connaître le coût  $C_{i,j}$  pour la correspondance entre une région  $i$  de l'image courante et celle  $j$  de l'image précédente (ou leur estimé dans l'image courante). Plus des régions sont semblables, plus le coût de leur mise en correspondance est bas. Ce coût est calculé à partir des caractéristiques selon une procédure de fusion simple. Par exemple, si l'image est en couleur, le coût dépendra de la différence de couleurs entre les deux régions. Actuellement,  $C_{i,j}$  est une combinaison linéaire de distances pondérées entre chaque caractéristique (4). Si  $\alpha_k = 0$  alors la distance  $d_k$  n'est pas calculée.

$$C_{i,j} = \sum_{k=1 \dots N} \alpha_k d_k(F_k^i, F_k^j) \quad (4)$$

**Mise en correspondance:** À partir d'une matrice de coût donnée, il y a beaucoup de matrices d'hypothèse potentielles pour la matrice de correspondance réelle [14]. Trois algorithmes peuvent être exécutés : "test de toutes les mises en correspondance possible", "test restreint des mises en correspondance possibles" et "choix simple directs d'une mise en correspondance".

- "Test de toutes les mises en correspondance possible" : toutes les hypothèses de correspondance sont évaluées, les  $K$  meilleurs hypothèses globales sont retenues. Cela prend beaucoup de temps de calcul. Par exemple à partir des  $K$  hypothèses précédentes, dans le cas d'un suivi de 6 régions, le nombre d'hypothèses à tester est de  $2^{6 \times 6}$ .
- "Test restreint des mises en correspondance possibles": On réduit le nombre de matrices de mise en correspondance possible. En effet, pour une région dans l'image courant, on ne regarde seulement la correspondance des  $N$  régions qui ont un coût d'appariement le plus bas.
- "Choix simple direct d'une mise en correspondance": C'est l'algorithme de correspondance le plus simple. Chaque région de l'image précédente est liée à la région la plus proche (ayant le coût le plus bas) dans l'image actuelle). De même, chaque région de l'image actuelle est aussi liée à la région la plus proche dans l'image précédente. Cependant, si le coût entre deux régions excède un seuil donné en paramètre, alors la liaison n'est pas prise en compte.

Notons qu'à chaque itération on peut permuter l'algorithme de mise en correspondance par un autre.

Notons, de plus, qu'il y a des prédictions multiples et des matrices de coûts multiples quand la dernière décision de

mise en correspondance n'est pas unique. Néanmoins, il y a seulement des décisions de correspondance multiples pour quelques algorithmes dits MHT (tracking multi-hypothèses [14]).

#### 4.7 Description et filtrage des résultats du suivi

Le but de la description et du filtrage du suivi est de faire l'interface entre le suivi et les processus d'analyses en aval en simplifiant l'information.

**Description:** Après le suivi, les seules nouvelles informations sont les correspondances entre les régions. La description du suivi ajoute quelques informations utiles à ces données. Il calcule l'âge de chaque piste, (c'est-à-dire la durée entre une région d'une image et le moment de son apparition), le temps avant la mort (c'est-à-dire la durée de la piste avant que la région ne disparaisse), mais aussi des informations de variations de caractéristiques comme la vitesse des objets à partir des positions successives. Il décrit aussi des morceaux de piste qui reste inclus dans une petite zone limitée comme une région arrêtée.

**Filtrage:** le filtrage de la description du suivi est effectué à la sortie de la description. Il est aussi nécessaire que les autres filtres du système de vision. Il sert à enlever le bruit. A ce niveau, le filtrage peut s'aider de la cohérence temporelle. Etant donné que la description du suivi est une construction morceau par morceau pendant l'avancée de la séquence vidéo, le filtrage peut être en ligne ou off-line. Nous décrivons ci-après quelques types de filtres qui peuvent être utilisés à la chaîne.

**Smalltrack:** détecte et enlève les pistes qui durent moins d'une durée fixée. La figure 13 montre la différence avec et sans ce filtre. Un filtre semblable est utilisé dans [15]. Ce type de bruit survient quand la segmentation détecte un bruit dans l'image comme une région mobile.

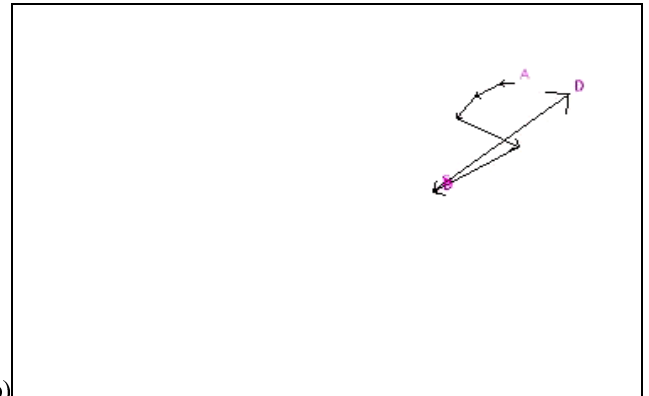
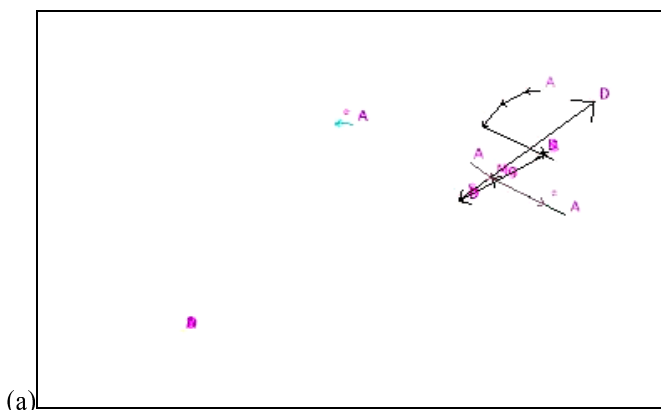


Figure 13: (a) description brute du suivi, (b) description filtrée par *smalltrack*. Les symboles A, D, S, Mg et \* signifient respectivement apparition, disparition, séparation, fusion (merge) et "objet actif dans la dernière image analysée".

**Simplifycurvetrack:** Simplifie des pistes en enlevant les positions qui donnent de pauvres informations (par exemple, si la région se déplace lentement). On pourrait voir cela comme un algorithme de re-échantillonnage dynamique. La figure 14 montre la différence avec et sans ce filtre.

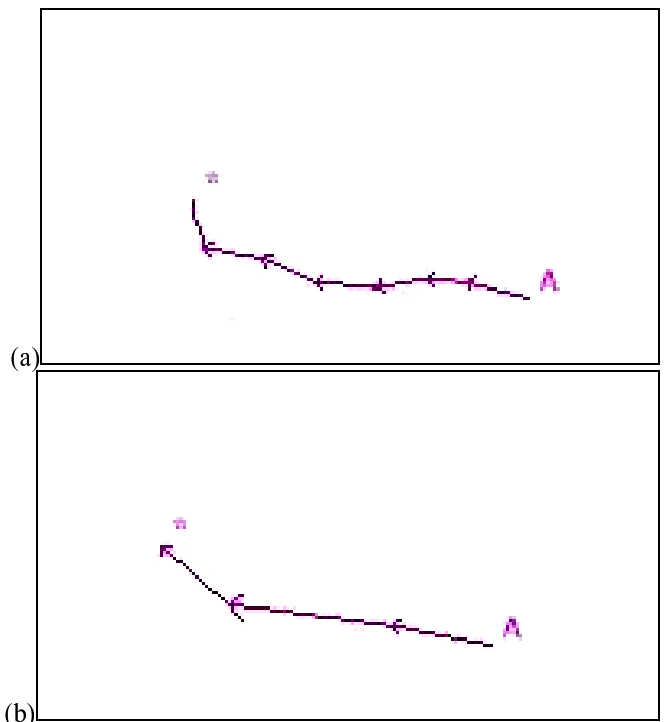


Figure 14: (a) description brute du suivi, (b) description filtrée par *simplifycurvetrack*.

**Simplifysplitmerge:** Enlève une partie d'une piste dédoublée provenant d'une séparation suivie d'une fusion. Ce type de bruit vient quand la segmentation détecte parfois deux régions alors qu'il n'y en a vraisemblablement qu'une (phénomène de sur-segmentation). La figure 15 montre les résultats avec ou sans le filtre.

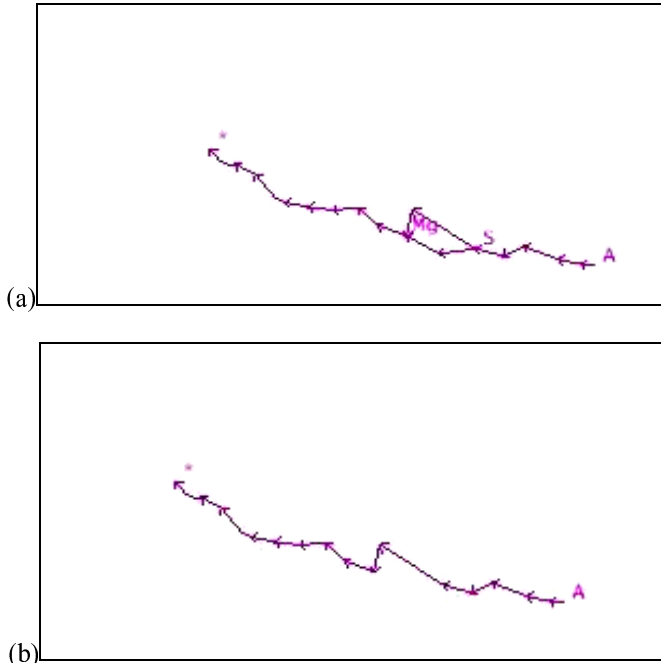


Figure 15: (a) description brute du suivi, (b) description filtrée par *simplifysplitaftermerge*.

**Simplifysplitaftermerge:** Découpe les pistes fusionnées puis séparées en fonctions des caractéristiques visuelles des objets (e.x :couleur). Ce type de bruit est typique de la méthode de tracking utilisée. La figure 16 montre les résultats avec ou sans le filtre.

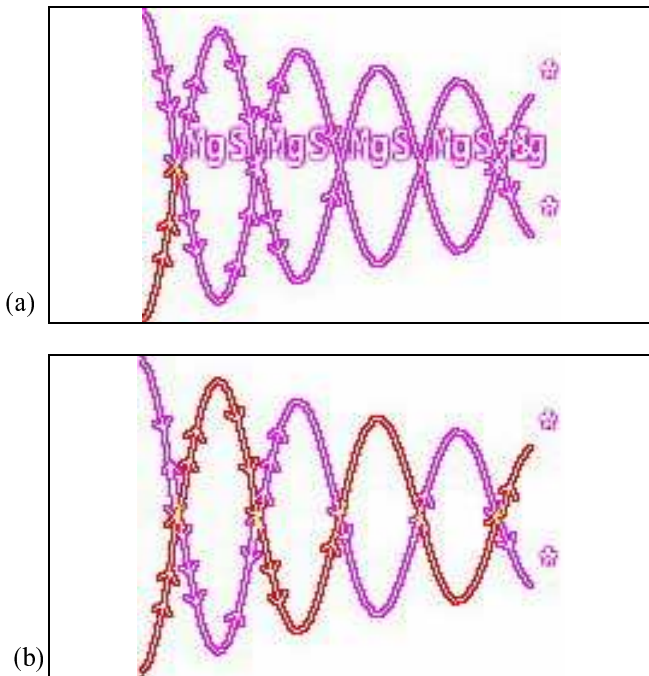


Figure 16: (a) description brute du suivi, (b) description filtrée par *simplifysplitaftermerge*.

Nous ne décrivons pas en détail les autres filtres mis en oeuvre. D'autres filtres peuvent enlever des pistes qui débutent ou aboutissent dans des régions définies de l'image, font disparaître les pistes d'objets fantômes.

#### 4.8 Bouclage vers la segmentation

Après le filtrage des pistes, il est possible de prendre des décisions de haut niveau. Ces décisions peuvent être utiles pour le module de segmentation bas niveau. Pour les relier, il faut un bouclage (feedback) qui s'ajoute à notre système en chaîne. Il est par exemple possible, avec une heuristique du filtrage du suivi, de reconnaître les objets fantômes et ainsi de demander à la segmentation d'intégrer la région correspondante dans le fond de la scène (background). La sortie de la segmentation est une carte de région. Une région peut représenter un objet réel ou un fantôme (c'est à dire une région qui est étiquetée faussement par la segmentation comme un objet). Une heuristique simple permet de détecter environ 90% de ces cas, elle utilise l'EMSG (Edge Mean Square Gradient), c'est à dire la somme quadratique des distances entre les valeurs des pixels intérieurs et extérieurs au contour, le tout divisé par la longueur du contour de la région. Quand cette mesure est en dessous d'un seuil, la région est considérée comme fantôme. Un exemple est donné en figure 17.



Figure 17. Représentation de l'EMSG pour une région fantôme.

#### 4.9 Analyse du suivi et génération d'évènements

L'analyse des résultats de suivi est un processus qui reçoit la description filtrée du suivi. Elle peut trouver des motifs prédéterminés comme des objets entrant par une zone définie de l'image et sortant par une autre, que la vitesse d'un objet a excédé une certaine vitesse limite, les objets fantômes issus d'une séparation d'un objet en mouvement. La figure 18 montre ce motif particulier. Dans notre application nous utilisons ce motif comme "quelqu'un prend un objet". Grâce au filtrage du suivi, l'analyse des scénarios est tout à fait facile à exécuter. Une approche à peu près semblable a été décrite dans [13].

La grammaire de la description du suivi de comportement d'objet inclut l'apparition (nouvelle cible), la séparation, la fusion, la disparition, l'arrêt, l'objet sans surveillance, l'entrée dans une zone, la sortie par une zone.

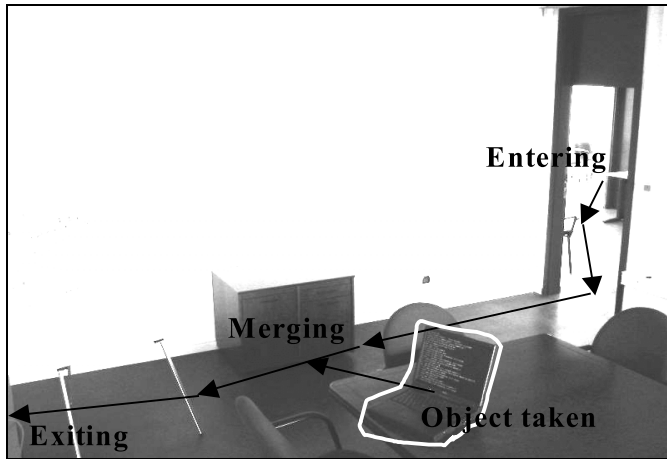


Figure 18: description du suivi pour le motif « objet volé ».

## 5. Conclusion

Nous avons introduit une approche pour l'intégration d'une plate-forme de surveillance vidéo de nouvelle génération qui peut fournir la flexibilité nécessaire aux chercheurs ainsi qu'aux exigences d'efficacité du monde industriel. Ce système a été validé sur des utilisations diverses. Nous avons décrit les parties des modules d'analyse d'image se concentrant sur la segmentation, le suivi et l'analyse.

Nous examinons actuellement des nouveaux modules de vision, par exemple, une meilleure segmentation et méthode de suivi. De plus, d'autres extensions et améliorations seront faites sur le système dans son ensemble. Par exemple, nous sommes en train d'intégrer de nouvelles briques standards pour la gestion des communications (middleware utilisant les technologies webservice [17]). Des travaux futurs intégreront aussi la pleine évaluation des paramètres et des méthodes en comparant et analysant [16] les sorties du système pour des applications spécifiques.

## Références

- [1] A. Cavallaro, D. Douxchamps, T. Ebrahimi and B. Macq, "Segmenting moving objects : the MODEST video object kernel", WIAMIS 2001, Workshop on Image Analysis for Multimedia Interactive Services, Tampere, Finland, May 16-17, 2001.
- [2] F. Cupillard, F. Brémond and M. Thonnat, "Tracking groups of people for video surveillance", Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems, London, September 2001.
- [3] T. Shcoepflin, C. Lau, R. Garg, D. Kim and Y. Kim, "A research Environment for Developing and Testing Object Tracking Algorithms", Proceedings of the SPIE, Electronic Imaging 2001, vol. 4310, pp. 667-675.
- [4] C. Jaynes, S. Webb, R. Steele and Q. Xiong, "An open development environment for evaluation of video surveillance systems", 3<sup>rd</sup> Int. Workshop on PETS, 1, pp. 32-39.
- [5] M. Valera and S.A. Velastin: "An Approach for Designing a Real-Time Intelligent Distributed Surveillance System", First Symposium on Intelligent Distributed Surveillance Systems (IDSS), IEE, 26 February 2003, London, pp.6/1-6/5.
- [6] <http://ffmpeg.sourceforge.net/>
- [7] A. D. Worrall, G. D. Sullivan and K. D. Baker, "A simple, intuitive camera calibration tool for natural images", Proceedings of 5th British Machine Vision Conference, University of York, York, pp. 781-790, 13-16 September 1994.
- [8] <http://sourceforge.net/projects/opencvlibrary/>
- [9] K. Okuma "Automatic Acquisition of Motion Trajectories: Tracking Hockey Players", M.Sc. Thesis, the University of British Columbia, May 2003.
- [10] A. Elgammal, R. Duraiswami, D. Harwood and L.S. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation", Proceedings of the IEEE, vol. 90, No. 7, pp. 1151- 1163, July 2002.
- [11] K. Kim, T. Horprasert, D. Harwood, L. Davis, "Codebook-based Background Subtraction and Performance Evaluation Methodology".
- [12] R. Cucchiara, C. Grana, A. Prati, R. Vezzani, "Using computer vision techniques for dangerous situation detection in domotics applications", Second Symposium on Intelligent Distributed Surveillance Systems, IEE, London, pp. 1-5, February 2004.
- [13] J.H. Piater, S. Richetto, and J. L. Crowley, "Event-based Activity Analysis in Live Video using a Generic Object Tracker", Proceeding 3<sup>rd</sup> IEEE Int. Workshop on PETS, Copenhagen, pp. 1-8, June 1 2002.
- [14] I. J. Cox and S.L. Hingorani, "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the purpose of Visual Tracking", IEEE Transactions on Pattern Analysis and Machine intelligence, vol. 18, pp. 138-150, February 1996.
- [15] A.E.C. Pece, "From Cluster Tracking to People Counting", Institute of Computer Science University of Copenhagen, Proceedings 3<sup>rd</sup> IEEE Int. Workshop on PETS, pp. 9-17, Copenhagen, June 1 2002.
- [16] T. Ellis, "Performance Metrics and Methods for Tracking in Surveillance", Information Engineering Centre, School of Engineering, City University, London. Proceeding 3<sup>rd</sup> IEEE Int. Workshop on PETS, pp. 9-17, Copenhagen, June 1 2002.
- [17] T. Gu, H.K. Pung and D.Q. Zhang, "Toward an OSGi-Based Infrastructure for Context-Aware Applications", IEEE Pervasive Computing, pp. 66-74, October 2004.