

# Vision par ordinateur

## TP 3 : Détection et suivi de mouvement

### (TP en binôme)

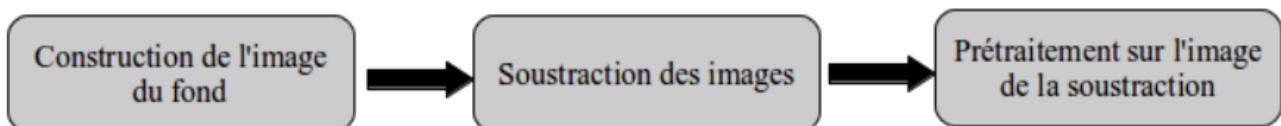
**MEDOU Daniel Magloire, Promo 21**  
**RAKOTOARIVELO Anjara Nobby, Promo 21**  
**Étudiants en Master 2, option SIM, IFI**

#### Objectif du TP

Ce TP consiste à implémenter un programme qui nous permettra de détecter les mouvements des personnes (ou objets) dans une vidéo et même de suivre ces mouvements tout en nous servant du filtre de Kalman pour l'effectivité de cette option. Alors, pour que cela soit possible, notre programme implémenté prendra en entrée les paramètres que nous jugerons importants voir indispensable pour la réussite de cette détection et de ce suivi et aussi le nom de la vidéo à détecter. En sortie, le programme devrait nous délivrer comme résultat l'image du fond de la vidéo, les images en binaire des mouvements détectées, aussi la trace de la trajectoire des mouvements des personnes (objets) qui est la partie suivi. A la suite de ce rapport, il vous sera présenté de façon détaillée les différentes parties que constituent notre programme à savoir **la détection de mouvement** et **le suivi de mouvement** et la partie **expérimentations et analyse des résultats**.

#### 1. La détection de mouvement

Dans cette section, il est préférable pour nous de mentionner les différentes étapes à suivre pour détecter les objets en mouvement dans une vidéo et ceci dans chacune des fenêtres de la séquence des images d'une vidéo. La schéma ci-dessous nous présente ce processus :



**Figure1** : Processus de la détection du mouvement

##### a) Construction de l'image de fond

Cette étape consiste à extraire l'arrière plan de la vidéo. La possibilité de construire cette image de fond nous est donnée par le **paramètre du nombre de séquence** qui est entre autre le nombre de fenêtres à prendre en compte pour la construction de cette image de fond. La valeur de ce paramètre entrée par l'utilisateur, affecte la qualité du résultat de la détection du mouvement chose que nous observerons bien dans la partie expérimentale. Chaque image capturée, lors de la lecture de la vidéo, est ensuite transformée en niveau gris. Après avoir atteint le nombre de capture choisi par l'utilisateur, nous pouvons passer à la construction de l'image de fond. Pour cela, on calcule la valeur médian de chaque pixel à partir de la liste des images capturées. Cette fonctionnalité donne comme résultat une image à niveau gris qui est considérée comme image du fond de la vidéo. Cette image résultante est servie pour la prochaine étape de la détection du mouvement. Pour la construction de l'image de fond, la lecture de toute la vidéo n'est pas nécessaire. Car, si les scènes

de la vidéo sont passées sur un même fond ou s'il n'y pas beaucoup de changement d'éclairage au niveau du fond de la vidéo. Pourtant, s'il y a de changement de fond dans la vidéo, on a besoin de mettre à jour l'image de fond.

### b) Soustraction des images

Pour cette étape qui consiste à détecter le mouvement, nous allons lire la vidéo jusqu'à la fin et pour chaque capture, on calcule la différence de l'image capturée à l'image de fond. La capture se fait très rapidement lors de la lecture de la vidéo ; environ 25 images pour chaque seconde. Cette fonctionnalité prend comme donnée en entrée l'image capturée et l'image du fond qu'on vient d'obtenir à partir de l'étape précédente. La soustraction se fait en temps réel. Autrement dit, à chaque capture d'images, on détermine tout de suite l'image de différence entre l'image nouvellement capturée à l'image du fond. Notons qu'avant de calculer la différence entre les images, nous appliquons **un lissage à l'image d'arrière-plan** et à l'image courante de la vidéo en utilisant la fonction « **GaussianBlur** » d'opencv. Pour le calcul de la soustraction, nous prenons en compte la valeur du **paramètre seuil** entrée par l'utilisateur. Pour cela, nous parcourrons tous les pixels de l'image capturée et nous calculons la différence de chaque pixel avec le pixel correspondant à l'image de fond. Si la valeur de la soustraction est inférieure à la valeur du seuil, on met zéro comme valeur de ce pixel, dans le cas contraire, on donne une valeur égale à 255 pour ce pixel. Autrement dit, si la valeur du pixel est 255, ce pixel appartient à un objet en mouvement mais non pas dans l'image de fond. L'ensemble des pixels en blanc forme des régions que l'on peut considérer comme objet en mouvement dans la vidéo. Cette étape nous donne comme résultat une image binaire qui présente les objets détectés dans l'image capturée. Cette image résultante présente encore des bruits, mais le fait d'appliquer la fonction. Donc, nous allons faire quelques prétraitements sur cette image pour améliorer les résultats de la détection de mouvement. Cela est présenté dans l'étape suivante.

### c) Prétraitement sur l'image de la soustraction

L'objectif de ce traitement est d'améliorer les résultats obtenus lors de l'étape de la soustraction en appliquant quelques opérations morphologiques et quelques méthodes de suppression des bruits sur l'image. Les étapes d'amélioration des images se présentent comme suit :

- ✓ Application de la fonction « **GaussianBlur** » / « **medianBlur** » de OpenCv sur l'image de la soustraction qui permet de faire de lissage sur l'image en utilisant le filtre médian. Cette action a pour but de réduire en amont les sources de bruits dans l'image caractérisant le mouvement.
- ✓ Utilisation de la fonction **findContours** sur l'image pour déterminer les contours de chaque objet détecté. Cette fonction prend en compte la valeur du paramètre **seuilContour** entrée par l'utilisateur. Elle considère les pixels à une distance supérieur au seuil comme des bruits. Elle permet de supprimer les bruits en éliminant les pixels en dehors des contours détectés.
- ✓ Application des fonctions morphologiques comme l'érosion et la dilatation. Ces fonctions morphologiques permettent de remplir les trous dans les objets de l'image. Nous avons utilisé une fois la dilatation , l'érosion et la fermeture morphologique.

La valeur du seuil entrée par l'utilisateur influence sur le résultat de la détection des images.

### d) Fonctionnement du programme de détection de mouvement

Le programme mise place est structuré en un fichier nommé « **tp3-binome-rakotoarivelo-medou\_detection\_suivi.cpp** » et pour fonctionner, ce dernier prend en entrée la **vidéo requête**, le **nombre de séquence** à prendre en compte pour la détection et le **seuil de détection**. En retour

(sortie), nous avons l'image de l'arrière-plan de la vidéo qui s'enregistre dans le sous-répertoire « **arriere\_plan** », les séquences d'images composant la vidéo originale enregistrées dans le sous-répertoire « **images\_videos** » et les séquences d'images nous faisant état des mouvements détectés des objets de la vidéo requête qui sont enregistrées dans le sous-répertoire « **images\_mouvement** ». Pour compiler notre programme, nous devons nous rassurer que nous sommes positionné dans le répertoire contenant notre code source du programme, cliquer droit sur votre souris et le menu contextuel se déroule puis cliquer sur « **Ouvrir dans un terminal** » et enfin taper la commande « **make** » puis appuyer sur la touche **ENTER** du clavier et la compilation de notre programme est lancée immédiatement.

Pour exécuter notre programme, nous devons taper après compilation réussie la commande suivante:

« **./tp3-binome-rakotoarivelo-medou\_detection\_suivi** ». Il est important de respecter la syntaxe donnée sinon vous allez générer une grande source d'erreurs. Une fois que cela est bien fait : Nous avons l'interface optionnelle ci-dessous qui s'affiche nous donnant la possibilité d'effectuer le choix sur le type d'opération que nous souhaitons.

```
daniel@daniel-Satellite-L855D:/media/daniel/DANIEL/IFI MASTER1&2 2017-2018/MASTER 2/VISION PAR ORDINATEUR/TP3_medou.p21/rakotoarivelo.p21-medou.p21-tp3$ ./tp3-binome-rakotoarivelo-medou_detection_suivi
*****
VEUILLEZ OPERER VOTRE CHOIX*****
1: Detection de mouvement
2: Suivi de mouvement

Veuillez entrez un chiffre pour le choix d'une opération: 1
```

Figure2 : Option sur le type d'opération sur le programme

- 1 : pour la « **Detection de mouvement** » ;
- 2 : pour la « **Suivi de mouvement** » .

Exemple : Si nous choisissons « 1 » pour « **Detection de mouvement** » et appuyons sur la touche « **ENTER** » du clavier, la figure ci-dessous s'affiche.

```
daniel@daniel-Satellite-L855D:/media/daniel/DANIEL/IFI MASTER1&2 2017-2018/MASTER 2/VISION PAR ORDINATEUR/TP3_medou.p21/rakotoarivelo.p21-medou.p21-tp3$ ./tp3-binome-rakotoarivelo-medou_detection_suivi
*****
VEUILLEZ OPERER VOTRE CHOIX*****
1: Detection de mouvement
2: Suivi de mouvement

Veuillez entrez un chiffre pour le choix d'une opération: 1
*****
BIENVENUE POUR LA DETECTION DE MOUVEMENT*****
Veuillez saisir le nom de la vidéo: 1
```

Figure3 : Option « 1 » pour Détection de mouvement

Ce qui nous reste à faire c'est de renseigner tous les paramètres :

- ✓ Nom de la vidéo requête ;
- ✓ Le nombre de séquence ;
- ✓ Le seuil de détection.

```

daniel@daniel-Satellite-L855D:/media/daniel/DANIEL/IFI MASTER1&2 2017-2018/MASTER 2/VISION PAR ORDINATEUR/TP3_medou.p21/rakotoarivelo.p21-medou.p21-tp3$ ./tp3-binome-rakotoarivelo-medou_detection_suivi

*****VEUILLEZ OPERER VOTRE CHOIX*****
1: Detection de mouvement
2: Suivi de mouvement

Veuillez entrez un chiffre pour le choix d'une opération: 1
*****BIENVENUE POUR LA DETECTION DE MOUVEMENT*****
Veuillez saisir le nom de la vidéo: Walk1.mpg
Veuillez saisir la valeur du nombre de séquences: 200
Veuillez saisir la valeur du seuil de détection: 50

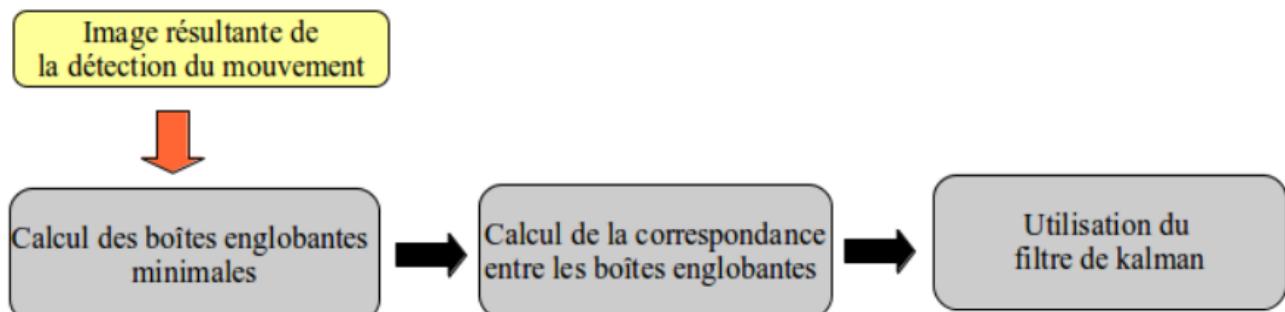
```

**Figure4** : Trois paramètres renseignés

Enfin appuyez sur la touche « **ENTER** » du clavier et la procédure de détection est lancée. La suite est réservée à la partie expérimentation (Section 3.).

## 2. Le suivi de mouvement

Dans cette deuxième partie, nous allons suivre les mouvements détectés dans la première partie en utilisant le filtre de Kalman. Quelques étapes ont été implémenté pour réaliser le suivi des mouvements. La figure suivante montre le processus pour le suivi du mouvement.



**Figure5** : Processus de suivi du mouvement

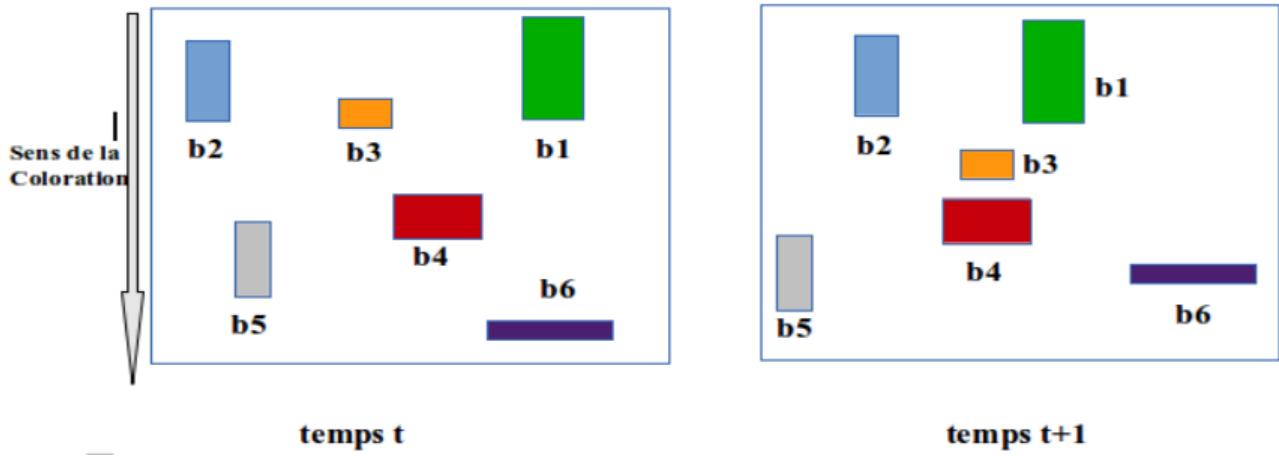
### a) Calcul des boîtes englobantes minimales

Après avoir déterminé la liste des objets en mouvement dans une fenêtre, nous allons déterminer les boîtes englobantes minimales de chaque objet détecté, de donner un numéro et une couleur à chaque boîte englobante et de les tracer sur la vidéo.

Pour calculer les boîtes englobantes de chaque capture, nous allons tout d'abord déterminer les contours de chaque objets détectés en utilisant la fonction « **findContours** » qui est une fonction implémenté dans OpenCv. A partir de ces contours détectés, nous pouvons construire les boîtes englobantes maximales des objets. Pour chaque objet détecté, on lui attribue un numéro. Ce numéro commence par 1 et incrémenté d'une pas de 1 à chaque fois que le programme détecte un nouvel objet. Dès qu'on arrive à attribuer des numéros à toutes les boîtes, on donne à chacune d'elles une couleur. Cette technique de coloration des boîtes englobantes va nous servir à la détection des correspondances des boîtes englobantes. L'ordre de coloration se fait de sens vertical. Autrement dit, on donne une couleur unique pour chaque boîte englobante en les ordonnant selon sa position sur l'axe vertical. La valeur du couleur est un entier ( $R=G=B=$  valeur du couleur). On peut changer

aussi l'ordre de coloration par la position par rapport à l'horizontal ou la taille des boîtes englobantes.

La figure suivante montre la détermination des boîtes englobantes :



**Figure6** : Détermination des boîtes englobantes

Une fois que les attributs des boîtes englobantes ont été détecté (position x, position y, centre, numéro, couleur, taille), nous traçons ces boîtes sur la fenêtre courante de la vidéo. On n'affiche pas la couleur des boîtes englobantes, elle sert seulement à détecter des correspondances avec les autres boîtes englobantes dont nous allons expliquer dans la section suivante.

### b) Calcul de la correspondance entre les boîtes englobantes

A chaque fenêtre capturée, on détermine les boîtes englobantes dans cette fenêtre. Pour ne pas donner des nouveaux numéros à toutes les boîtes englobantes détectées, nous allons se référer aux boîtes englobantes de la fenêtre précédente. Cette référence permet de détecter qu'un objet est nouveau ou il est déjà détecté dans la fenêtre précédente. Pour cela, nous allons prendre en compte la valeur du paramètre **seuilCorresp** entrée par l'utilisateur et les couleurs des boîtes.

Pour déterminer la correspondances des boîtes englobantes, nous avons utilisé la matrice d'appariement. Pour cela, nous avons calculé la distance spatiale (distance selon les positions des deux boîtes) et la distance couleur (distance selon les couleurs des deux boîtes). Après avoir calculé ses distances, nous allons unifié les différents critères en élaborant la matrice d'appariement unifié. Les tableaux suivants présentent la technique de correspondance implémenté dans notre programme.

T\T+1	b1	b2	b3	b4
b1	10	60	60	70
b2	60	5	40	90
b3	60	50	30	5
b4	70	50	0	50

**Tableau 1:** Distance entre centres

T\T+1	b1	b2	b3	b4
b1	0	1	1	1
b2	1	0	1	0
b3	1	1	1	1
b4	1	1	0	1

**Tableau 3:** Distance couleurs

T\T+1	b1	b2	b3	b4
b1	0	x	x	x
b2	x	0	x	90
b3	x	x	x	x
b4	x	x	0	x

**Tableau 2:** Fusion des deux critères

Les trois tableaux juste avant nous montrent que pour déterminer les correspondances entre les boîtes englobantes de la fenêtre à l'instant T avec celles de la fenêtre à l'instant T+1, on calcule tout d'abord les distances entre les centres de gravité des boîtes. Le calcul se fait deux à deux entre elles. Ensuite, on calcule la distance des couleurs des boîtes englobantes. Ces deux calculs nous permettent de conclure s'il y a correspondance entre les boîtes de la fenêtre courante avec celle de la

fenêtre précédente. Pour la correspondance, au niveau de la distance, on trie les distances calculées. Si la distance minimale est inférieure au seuil, il y a correspondance sinon il faut tester la valeur minimale suivante si elle est inférieur au seuil. Pour la distance de couleur, il y a correspondance si la boîte a la même couleur avec une boîte dans la fenêtre précédente. Si la boîte ne vérifie pas les deux critères, on la considère comme nouvelle boîte et on lui donne un nouveau numéro avec une nouvelle couleur.

### c) Utilisation du filtre de Kalman.

Dans cette section, nous allons suivre les mouvements des objets en utilisant le filtre de Kalman. Pour cela, nous allons déterminer la trajectoire prédictive, la trajectoire mesurée et la trajectoire corrigée de chaque objet pour chaque fenêtre capturée, et nous allons présenter dans une fenêtre. Nous avons fait aussi le suivi de mouvement pour chaque objet détecté. On peut voir donc le suivi global et le suivi de chaque objet.

Pour la présentation des différentes trajectoires sur une fenêtre, nous avons utilisé les symboles suivants:

- ✓ Croix bleue pour la trajectoire prédictive
- ✓ Rectangle en vert pour la trajectoire mesurée
- ✓ Cercle en rouge pour la trajectoire corrigée.

Pour la traçage de ces trajectoires, nous allons considérer la valeur du paramètre vitesse entrée par l'utilisateur. Cette valeur permet de prendre en compte la vitesse. Si cette valeur est nulle, on ne prend pas en compte la vitesse, on étudie seulement les positions. Les graphes résultantes de ce filtre permettent d'évaluer la détection et le suivi offert par notre programme. Nous allons analyser les résultats des expérimentations faites dans la partie expérimentation (Section 3.).

### d) Fonctionnement du programme

Le programme mis en place est structuré en un fichier « **tp3-binome-rakotoarivelo-medou\_detection\_suivi.cpp** » et pour fonctionner, ce dernier prend en entrée la **vidéo requête**, le **nombre de séquence** à prendre en compte pour la détection, le **seuil de détection** et le **seuil de correspondance**. En retour (sortie), nous avons l'image de l'arrière-plan de la vidéo qui s'enregistre dans le sous-répertoire « **arriere\_plan** », les séquences d'images composant la vidéo originale enregistrées dans le sous-répertoire « **images\_videos** » et les séquences d'images nous faisant état des mouvements suivis des objets de la vidéo requête qui sont enregistrées dans le sous-répertoire « **images\_suivi** ».

Pour compiler notre programme, nous devons nous rassurer que nous sommes positionné dans le répertoire contenant notre code source du programme, cliquer droit sur votre souris et le menu contextuel se déroule puis cliquer sur « **Ouvrir dans un terminal** » et enfin taper la commande « **make** » puis appuyer sur la touche **ENTER** du clavier et la compilation de notre programme est lancée immédiatement.

Pour exécuter notre programme, nous devons taper après compilation réussie la commande suivante:

« **./tp3-binome-rakotoarivelo-medou\_detection\_suivi** ». Il est important de respecter la syntaxe donnée sinon vous allez générer une grande source d'erreurs. Une fois que cela est bien fait : Nous avons l'interface optionnelle ci-dessous qui s'affiche nous donnant la possibilité d'effectuer le choix sur le type d'opération que nous souhaitons.

```
daniel@daniel-Satellite-L855D:/media/daniel/DANIEL/IFI MASTER1&2 2017-2018/MASTER 2/VISION PAR ORDINATEUR/TP3_medou.p21/rakotoarivelo.p21-medou.p21-tp3$ ./tp3-binome-rakotoarivelo-medou_detection_suivi
*****
VEUILLEZ OPERER VOTRE CHOIX*****
1: Detection de mouvement
2: Suivi de mouvement

Veuillez entrez un chiffre pour le choix d'une opération: 
```

**Figure5** : Option sur le type d'opération sur le programme

- 1 : pour la « **Detection de mouvement** » ;
- 2 : pour la « **Suivi de mouvement** » .

**Exemple :** Si nous choisissons « 2 » pour « **Suivi de mouvement** » et appuyons sur la touche « **ENTER** » du clavier, la figure ci-dessous s'affiche.

```
daniel@daniel-Satellite-L855D:/media/daniel/DANIEL/IFI MASTER1&2 2017-2018/MASTER 2/VISION PAR ORDINATEUR/TP3_medou.p21/rakotoarivelo.p21-medou.p21-tp3$ ./tp3-binome-rakotoarivelo-medou_detection_suivi
*****
VEUILLEZ OPERER VOTRE CHOIX*****
1: Detection de mouvement
2: Suivi de mouvement

Veuillez entrez un chiffre pour le choix d'une opération: 2
*****
BIENVENUE POUR LE SUIVI DE MOUVEMENT*****
Veuillez saisir le nom de la vidéo: 
```

**Figure6** : Option « 2 » pour Suivi de mouvement

Ce qui nous reste à faire c'est de renseigner tous les paramètres :

- ✓ Nom de la vidéo requête ;
- ✓ Le nombre de séquence ;
- ✓ Le seuil de détection ;
- ✓ Le seuil de correspondance.

```
daniel@daniel-Satellite-L855D:/media/daniel/DANIEL/IFI MASTER1&2 2017-2018/MASTER 2/VISION PAR ORDINATEUR/TP3_medou.p21/rakotoarivelo.p21-medou.p21-tp3$ ./tp3-binome-rakotoarivelo-medou_detection_suivi
*****
VEUILLEZ OPERER VOTRE CHOIX*****
1: Detection de mouvement
2: Suivi de mouvement

Veuillez entrez un chiffre pour le choix d'une opération: 2
*****
BIENVENUE POUR LE SUIVI DE MOUVEMENT*****
Veuillez saisir le nom de la vidéo: Walk1.mpg
Veuillez saisir la valeur du nombre de sequences: 200
Veuillez saisir la valeur du seuil de detection: 50
Saisir la valeur du seuil de correspondance: 25
```

**Figure7** : Quatre paramètres renseignés

Enfin appuyez sur la touche « **ENTER** » du clavier et la procédure de suivi est déclenchée.

### 3. Expérimentations et analyse des résultats

Cette partie constitue un point focal important pour évaluer les performances de notre programme dans la détection des mouvements des personnes ou objets dans une vidéo. Ces tests expérimentaux seront effectués à partir des vidéos extraits des bases INRIA et Portugal. Vidéos donc vous pouvez trouver en cliquant sur ce lien: <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>

Il est important pour nous, de faire les tests sur l'extraction de l'arrière plan de différentes vidéos avec des valeurs différentes du nombre de séquence (25, 50, 100, 200) pour prendre que ces valeurs et de retenir la meilleure valeur du nombre de séquence qui nous produira un meilleur arrière plan et cette dernière sera utile pour la suite du projet.

Nous verrons également dans cette partie l'influence des différents paramètres, aussi les qualités des résultats de notre programme face aux différents niveaux de difficultés de suivi du mouvements dans des vidéos.

#### a) Extraction de l'arrière plan

Les vidéos qui nous permettront d'extraire les arrières-plans sont celles des caméras de surveillance donc le décor est le même pour une certaine position de la caméra.

##### Cas1 : Nombre de séquence 25



Figure8



Figure9



Figure10

**Figure8 : Arrière plan de la vidéo EnterExitCrossingPaths1cor.mpg**

**Figure9 : Arrière plan de la vidéo EnterExitCrossingPaths2front.mpg**

**Figure10 : Arrière plan de la vidéo Meet\_Crowd.mpg**

La **Figure8** présente l'arrière-plan obtenu à partir des 25 premières séquences et nous donne en sortie de fond, des personnes que nous observons. Ces dernières étant en mouvement dans la vidéo nous pouvons dire que c'est une erreur. L'explication de cette erreur peut être donnée par le simple fait que ces personnes masquent l'arrière-plan. Ce qui montre inefficacité du nombre de séquence appliquée à ces vidéos.

Ce pendant, les **Figures 9 et 10** nous retournent des bons arrière-plan. Ceci est dû au fait que, les premières secondes de ces vidéos se succèdent par une suite d'images d'arrière-plan vide. D'où la meilleure extraction.

## Cas2 : Nombre de séquence à différentes valeurs de la vidéo EnterExitCrossingPaths1cor.mpg

Temps de la vidéo : 11 secondes



Figure11



Figure12



Figure13



Figure14



Figure15



Figure16

**Figure11 : Arrière plan de la vidéo EnterExitCrossingPaths1cor.mpg\_150**

**Figure12 : Arrière plan de la vidéo EnterExitCrossingPaths1cor.mpg\_200**

**Figure13 : Arrière plan de la vidéo EnterExitCrossingPaths1cor.mpg\_300**

**Figure14 : Arrière plan de la vidéo EnterExitCrossingPaths1cor.mpg\_350**

**Figure15 : Arrière plan de la vidéo EnterExitCrossingPaths1cor.mpg\_375**

**Figure16 : Arrière plan de la vidéo EnterExitCrossingPaths1cor.mpg\_380**

Le nombre des fenêtre à considérer pour construire l'image du fond infecte beaucoup sur le résultat. Il est vraiment coûteux au niveau du temps et des ressources pour lire la vidéo entière afin de construire l'image de fond. Pourtant, une quantité faible des captures à considérer provoque la mauvaise qualité de l'image de fond et notons également que le nombre de séquence (quantité de capture) est importante en fonction du temps de la vidéo. La **figure16** ci-dessus nous montrent l'intérêt de bien calibrer les nombres des fenêtres à lire pour la construction de l'image de fond, bien qu'à 380 nombre de séquence l'arrière-plan n'est pas toujours totalement parfait mais est nettement mieux que les autres figures car cette vidéo a un temps de 11 secondes. A plus de ce nombre de séquence, le programme ne lit plus la vidéo requête que nous lui avons soumis. Car, les personnes sont toujours présentes mais nous remarquons qu'elles sont moins nettement visible ce qui confirme l'amélioration évoquée.

## Cas2 : Nombre de séquence à différentes valeurs de la vidéo TwoEnterShop1cor.mpg

Temps de la vidéo : 63 secondes



Figure17



Figure18



Figure19



Figure20



Figure21



Figure22

Figure17 : Arrière plan de la vidéo TwoEnterShop1cor.mpg\_500

Figure18 : Arrière plan de la vidéo TwoEnterShop1cor.mpg\_600

Figure19 : Arrière plan de la vidéo TwoEnterShop1cor.mpg\_800

Figure20 : Arrière plan de la vidéo TwoEnterShop1cor.mpg\_1000

Figure21 : Arrière plan de la vidéo TwoEnterShop1cor.mpg\_1100

Figure22 : Arrière plan de la vidéo TwoEnterShop1cor.mpg\_1200

Le temps de cette vidéo étant relativement long que la précédente alors, nous avons eu la possibilité d'augmenter le nombre de séquences jusqu'à **1200** et le résultat est celui de la **figure22** qui est parfait. Car toute personne se trouvant dans cette vidéo a été extrait et nous obtenons un arrière-plan correct.

Nous pouvons conclure que, le temps de la vidéo est un facteur favorable pour l'extraction d'un arrière-plan parfait ainsi que le nombre de séquence influence aussi sur la qualité de l'extraction de l'arrière-plan. Sur ce, il n'est pas du tout nécessaire de traiter toutes les séquences de la vidéo pour en extraire l'arrière-plan.

Nous considérons 380 comme nombre de séquence pour la suite et la meilleure valeur du seuil de réductions des bruits sera retenue après plusieurs tests à la suite de nos expérimentations sur la détection de mouvement.

b) détection de mouvement : Influence du seuil de détection ( seuil de soustraction)

Nom de la vidéo test : TwoEnterShop1cor.mpg



Figure23 : Image originale

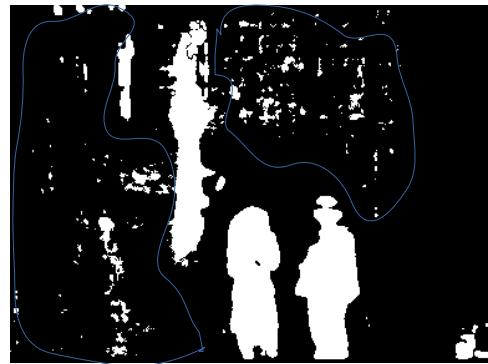


Figure24 : Soustraction de 10 : Masque illustrant les personnes détectées



Figure25:Personne détectée avec boîte englobante

La **figure24** nous montre qu'il y a beaucoup de bruits sur l'image résultante de la fonction soustraction si le seuil est très petit. Cela s'explique par le fait qu'il considère les petits bruits qui ont des valeurs supérieures au seul comme objets.



Figure26 : Personne détectée avec boîte englobante

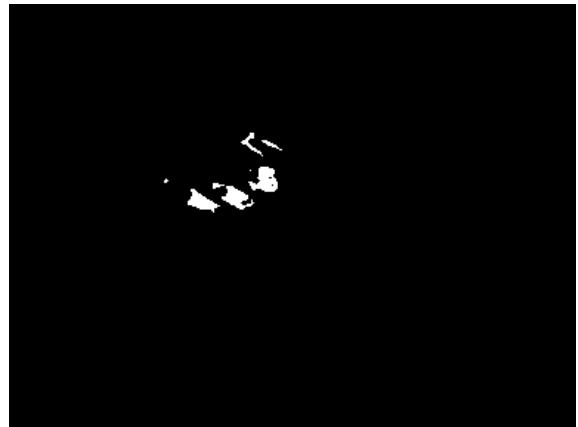


Figure27 : Soustraction de 50 : Masque illustrant le personne détectée

Avec le seuil de 50, le résultat de la soustraction est bon, seul l'objet en mouvement est détecté et la boîte englobante encadre bien le personnage. Les bruits sont supprimés et l'objet est bien rempli. Ce résultat provient du test effectué avec la vidéo **EnterExitCrossingPaths2cor.mpg**.



**Figure28 : Personne détectée avec boîte englobante**



**Figure29 : Soustraction de 100 : Masque illustrant les personnes détectées**

Lorsque nous augmentons la valeur du seuil à **100**, les bruits sont toujours supprimés mais il y a beaucoup de trous au niveau des objets détectés; cas de la **figure29**, aussi, certains régions importantes sont annulées comme les bras et les têtes des hommes se trouvant sur la **figure29**. Concernant la qualité des boîtes englobantes, elle est liée à la disposition de la caméra.

Notre programme peut, tout de même détecter plusieurs personnes à la fois comme le présente la **figure28 avec les trois petites boîtes englobantes que nous observerons**.

En conclusion, La valeur trop petite ou trop grande du seuil de soustraction influence beaucoup sur les qualités de détection des mouvements et aussi le profil ou bien la position de la caméra est aussi importante pour la bonne qualité de détection des mouvements sur une vidéo.

### c) Suivi de mouvement

Les expérimentations du suivi de mouvements seront effectuées avec certains niveaux de difficultés.

#### ✓ Test avec un seul objet en mouvement sur une trajectoire simple

Pour ce test, la vidéo que nous avons opté est la suivante : **TwoLeaveShop1front.mpg** avec **25** comme seuil de correspondance.



**Figure30**



**Figure31**

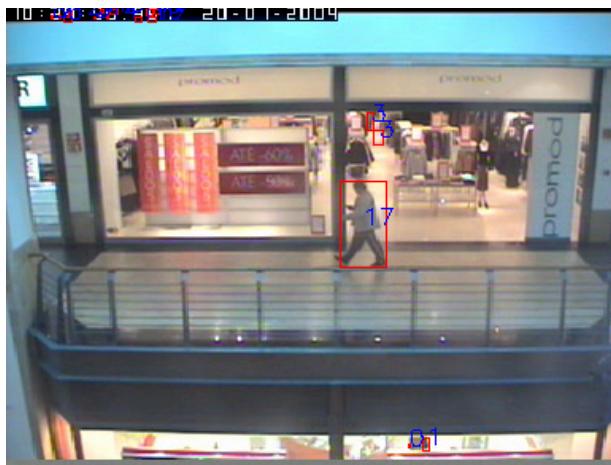


Figure32



Figure33



Figure37

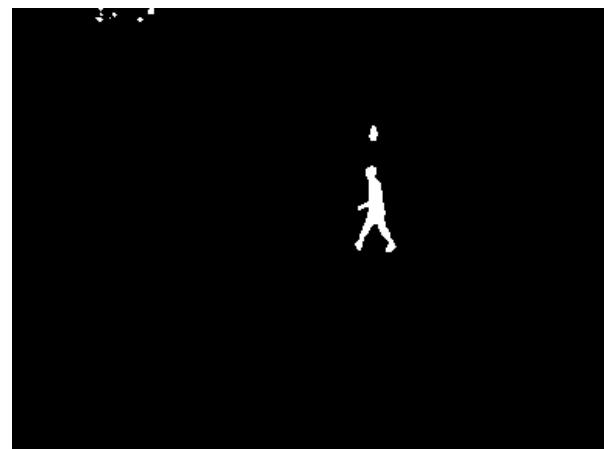


Figure35

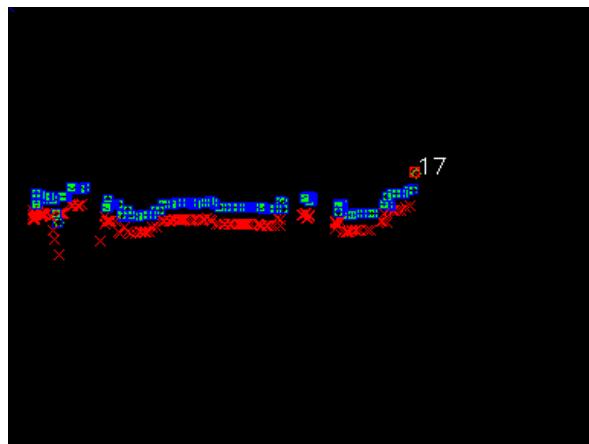


Figure36 : Filtre de Kalman du suivi de mouvement de la personne Figure30 et 32

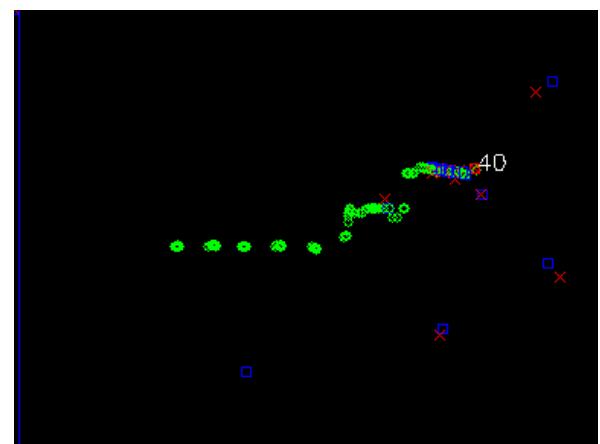


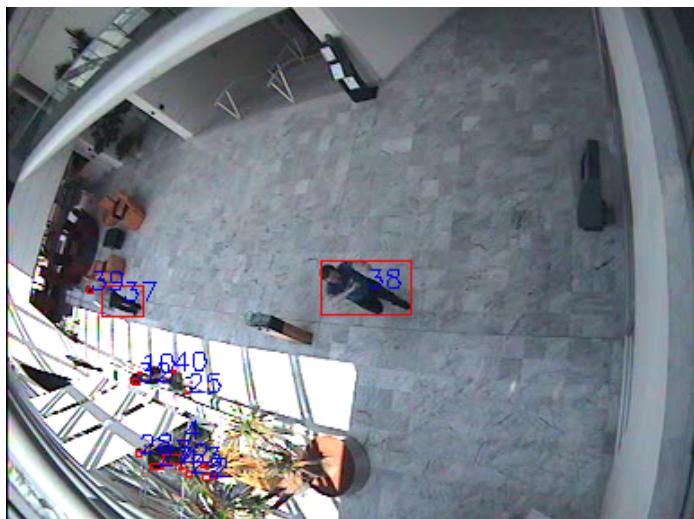
Figure37 : Filtre de Kalman du suivi de mouvement de la personne Figure37

Les figures ci-dessus montrent les séquences d'images (40 et 41) capturées à partir de la vidéo **TwoLeaveShop1front.mpg** et une autre choisie de façon aléatoire (1148). Les résultats retournés par le programme sont assez bons. La personne est bien détectée. Aussi, l'objet est bien rempli, il n'y a pas des trous à l'intérieur de l'objet. Les images 40,41 et 1148 ne présentent aucun bruit et la taille de l'objet correspond bien à l'image originale. Donc, la détection du mouvement est très bien. Ce qui

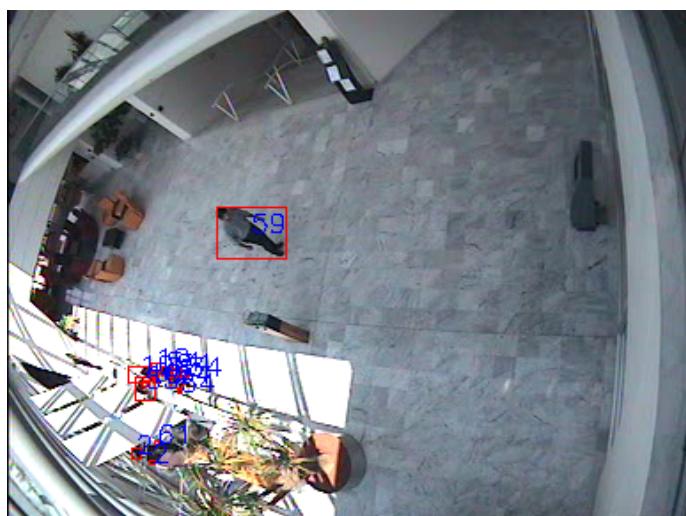
influence aussi la qualité du suivi du mouvement est, la personne reçoit un seul numéro d'étiquette (numéro 1) et ne change pas jusqu'à ce qu'elle arrive au bout. Ce qui montre qu'on est entrain de suivre les mouvements de cette dernière. Le **filtre de Kalman** permet de confirmer cette information. Les **figures 36 et 37** nous montrent bien les trajectoires des personnes suivies.

- ✓ Test avec un seul objet en mouvement avec une trajectoire complexe

Pour ce test, la vidéo que nous avons opté est la suivante : **Browse1.mpg** avec **15** comme seuil de correspondance.



## Capture 124



Capture 524

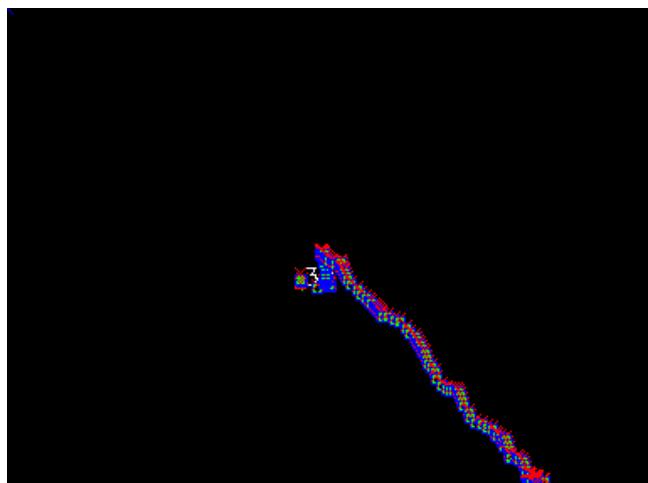


## Capture 761

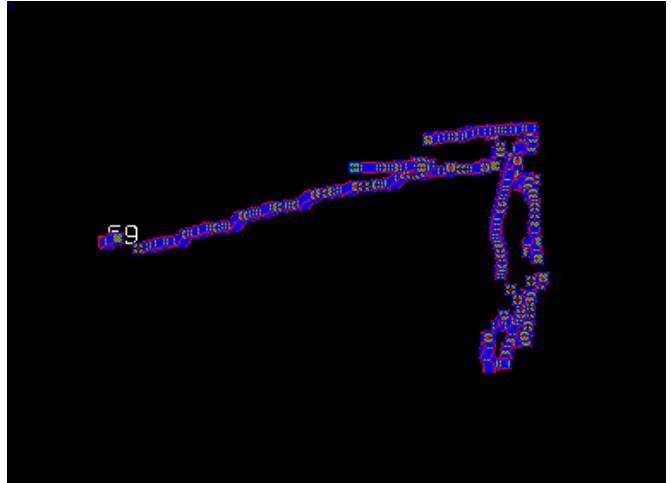


**Capture 1025**

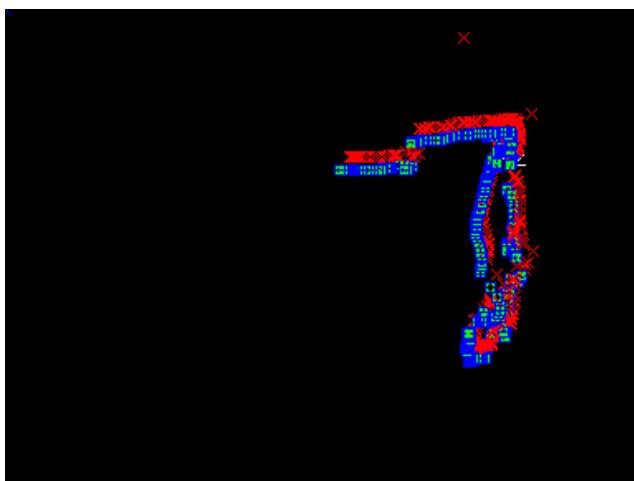
Les captures suivantes représentent respectivement le suivi des objets capturés avec le filtre de Kalman.



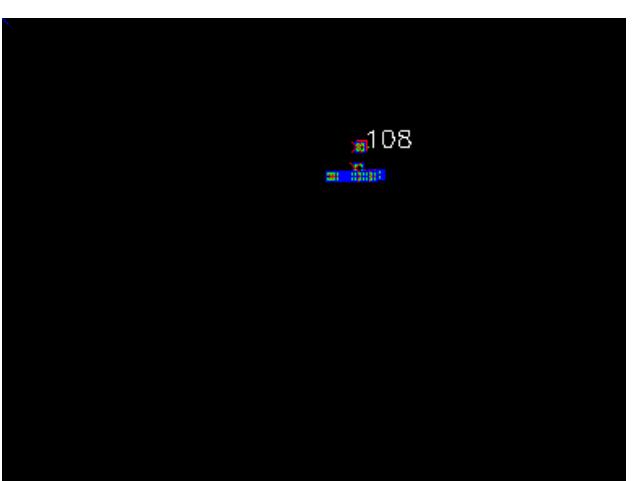
**Capture 124**



**Capture 524**



**Capture 761**



**Capture 1025**

Pour suivre les mouvements d'un homme dans une trajectoire complexe, nous avons choisi quelques captures intéressants permettant d'évaluer la fonctionnalité de notre programme. L'explication de chaque capture se présente comme suit :

- ✓ A la **capture 124**, le corps en entier de l'homme est bien détecté avec boîte englobant parfaite. L'objet est étiqueté du numéro **38**. Au niveau de la détection des mouvements, il y a un autre objet moins bien détecté dernière avec deux numéros 37 et 39, une femme qui fait des vas et vient et une autre personne en dehors du grille. Tous les bruits sont supprimés, il ne reste que les objets en mouvements.
- ✓ A la **capture 524**, le programme détecte parfaitement l'homme. Il est parfaitement entouré par la boîte englobante. Le numéro d'identité ou de suivi de l'homme est **59**. A sa droite plusieurs objets sont en mouvement et sont aussi détectés par notre programme. Tous les bruits sont supprimés, il ne reste que les objets en mouvements.
- ✓ A la **capture 761**, le programme ne détecte que le pied de l'homme. Au niveau de la détection des mouvements, l'homme se présente seulement par sa tête et ses pieds. Ceci s'explique par le fait que la couleur de la chemise de l'homme se ressemble à l'image de fond. Donc, la partie supérieure de l'homme est considéré comme image de fond. Le numéro d'étiquette est **72**.
- ✓ A la **capture 761**, le corps entier de l'homme est de nouveau bien détecté et reçoit le numéro d'étiquette **108**. Il y a la main de l'homme qui est coupée.

Notre programme nous présente des résultats moyennement bon pour ce type de mouvement. IL arrive à détecter tous les mouvements de l'homme. Mais, à chaque direction que ce dernier va prendre, il lui attribue un nouveau numéro de détection. Le programme ne parvient pas à faire la différence entre l'image de fond et la couleur du vêtement de la personne quand celle-ci a un vêtement de même couleur que celui du fond. Sur ce, nous pouvons dire que le facteur éclairage et l'ambiguïté de couleur de fond influence énormément nos résultats.

Quant au filtre de Kalman, il nous montre que la trajectoire est complexe et il y a beaucoup d'erreurs au niveau de la prédiction.

#### ✓ **Test avec plusieurs objets en mouvement avec et sans croisement de trajectoire**

Pour ce test, la vidéo que nous avons opté est la suivante : **Browse1.mpg** avec **15** comme seuil de correspondance.



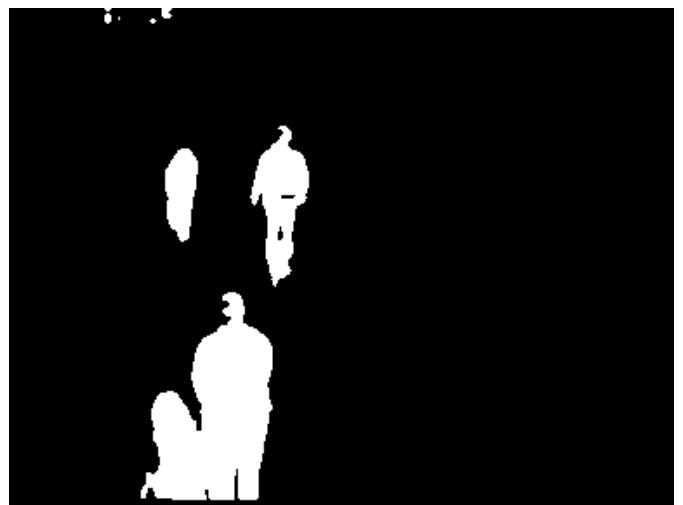
**Capture 369**



Capture 408

Pour cette première partie de la vidéo, nous allons faire un suivi des mouvements sur la trajectoire sans croisement. Les résultats retournés par le programme sont bons parce que il arrive à bien déetecter les mouvements des deux personnes tout en gardant leurs numéros suivis **50** et **55**. A la capture **369** et **408**, la détection ne présente aucun bruit et les images sont bien remplies. Les tailles des images correspondent bien aux tailles originales. Pourtant, il y a un problème pour cette vidéo. La détection affiche 3 objets. Mais, dans les fenêtres originales, il n'y a que deux personnes qui bougent. Le troisième objet est dû à l'image de fond. Ceci s'explique par le fait que le vieil homme existe toujours dans toutes les scènes de la vidéo. Il est donc impossible de l'enlever. Et comme la construction de l'image de fond se base sur la soustraction et les valeurs médians des pixels, la position initiale de l'homme est considérée comme image de fond. C'est pour ça il s'affiche toujours dans toutes les fenêtres de détection du mouvements.

Le programme arrive à bien suivre les mouvements des deux personnes tant qu'il n'y a pas de croisement entre les deux personnes. Les deux personnes gardent toujours leurs numéros jusqu'à ce que le croisement aura lieu. Dans la deuxième partie de la vidéo, nous allons suivre les mouvements des personnes avec le croisement dans la trajectoire.



Capture : 794

Le **Capture : 794**, lors du croisement de l'homme avec la femme, les deux personnes forment un objet dont le numéro est l'ancien numéro de la boîte englobante de l'homme dans des fenêtres précédentes. Le numéro de la personne derrière change souvent lors de son déplacement. Ceci est dû au changement d'éclairage et de l'ombre. Aussi, l'objet de l'homme derrière est partagé en

plusieurs parties d'objets dans certains temps. Ceci s'explique par le fait que lors du déplacement de cet homme, les couleurs de ses habits se ressemblent aux couleurs du fond. Donc, le programme considère quelque parties de cet homme comme image de fond. D'où le partage de l'objet.

#### ✓ Test avec plusieurs objets en mouvement avec occlusions

Pour ce test, la vidéo que nous avons opté est la suivante : **ThreePastShop2cor.mpg** avec **15** comme seuil de correspondance.



## Conclusion

En dernière analyse, le TP nous a permis d'implémenter un programme nous permettant de faire la détection et le suivi de mouvement des objets dans la vidéo. Ce pendant, la détection de mouvements, le programme exige certaines étapes pour que la détection de mouvements soit effective. Nous avons le construction de l'image de fond, la soustraction des images et le prétraitement sur l'image de la soustraction. Dans le cas du suivi de mouvements, le programme implémenté s'est vu intégré le calcul des boîtes englobantes minimales, le calcul de la correspondance entre les boîtes englobantes et l'utilisation du filtre de Kalman. Toutes ces étapes sont bien détaillées dans ce rapport.

Les résultats de la détection des mouvements influencent beaucoup au niveau de la qualité de résultats. Nous avons implémenté une méthode au niveau de la détermination de la correspondance des boîtes englobantes afin d'améliorer les résultats des mouvements. Pour cela, nous avons utilisé la notion de distance spatiale et la distance couleur. Dans la plupart des tests avec différents niveaux de difficulté, notre programme arrive à donner des bons résultats et surtout arrive à conserver les numéros attribués à chaque objet pour certains cas. Les résultats retournés par le programme sont acceptables selon les interprétations à partir du filtre du Kalman. Il y a quelques points à améliorer dans notre programme pour avoir des meilleurs résultats. Ces points à améliorer peut être la prise en charge des changement de l'intensité lumineuse, amélioration de la construction de l'image de fond en utilisant d'autres méthodes.