Détection et suivi de mouvement

La détection et le suivi de mouvement sont un secteur très important dans le domaine de la vision par ordinateur. Les champs d'application sont nombreux et variés.
Par exemple :

- Inspection et contrôle de qualité
- Flot de véhicules routiers et piétons
- Reconnaissance des expressions faciales
- Identification par la pupille
- Reconnaissance des gestes
- Imagerie médicale 3D
- Robotique intelligente
- Sécurité

Plusieurs approches sont disponibles pour résoudre ce problème. Certaines approches sont plus compliquées et d'autre beaucoup plus simple. Le choix et la complexité de l'approche dépendent souvent de facteurs non contrôlables inhérents à la scène qui doit être observée.

Nous allons utiliser une approche simple qui permet de détecter le mouvement des objets dans un environnement stable et contrôlé. Elle se divise normalement en trois étapes.

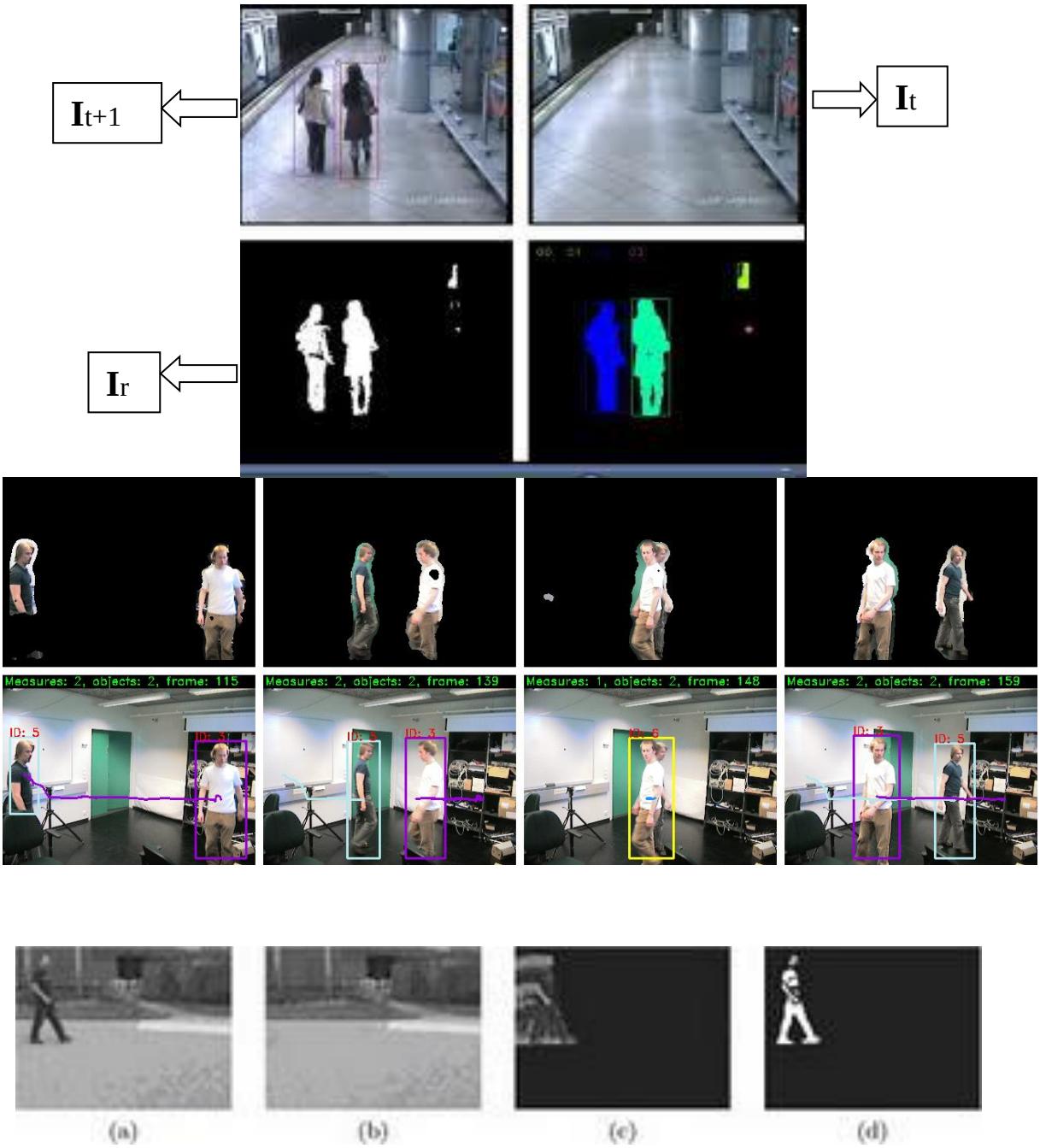
1- Soustraction de l'arrière-plan.
2- Filtrage
3- Détection des limites par sommation des lignes et des colonnes

Nous allons implanter dans ce travail les étapes 1 et 3. L'étape 2 étant passablement compliquée, nous allons compenser en utilisant des séquences vidéo tournées dans un environnement stable à partir d'une caméra fixe.

Explication des étapes :

1- Soustraction de l'arrière-plan :

La soustraction de l'arrière-plan est relativement simple. Il s'agit de prendre chaque pixel de l'image **I** pris au temps t ($\mathbf{I}t$) et de les soustraire au pixel correspondant dans l'image **I** prise au temps t+1 ($\mathbf{I}t_{+1}$). La soustraction se fait de manière absolue. Le résultat produira une nouvelle image **I**r ($\mathbf{I}r = |\mathbf{I}t - \mathbf{I}t_{+1}|$). De ce fait, l'image résultat **I**r est une image noir et blanc ou l'ont retrouve exclusivement tout ce qui a changé entre l'image **I**t et l'image $\mathbf{I}t_{+1}$.



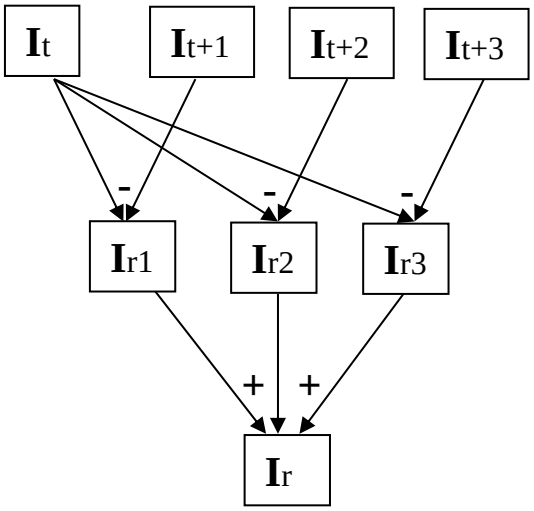
Exemple de résultats visuels de l'opération de soustraction :

- http://www.youtube.com/watch?v=X4DnOcsuR10
- http://www.youtube.com/watch?v=rGMqXBvYxog&feature=related
- http://www.youtube.com/watch?v=NfUJLLZyCTA&feature=related

Afin de rendre cette technique plus robuste, il faut effectuer cette opération sur quelques images adjacentes dans le temps. Si N est le nombre d'image adjacente sur lesquelles on effectue l'opération de soustraction. Alors, nous aurons :

$$\mathbf{I}r = \sum_{i=1}^{N} \left| \mathbf{I}t - \mathbf{I}t_{+i} \right|$$

Schématiquement avec N=3 :



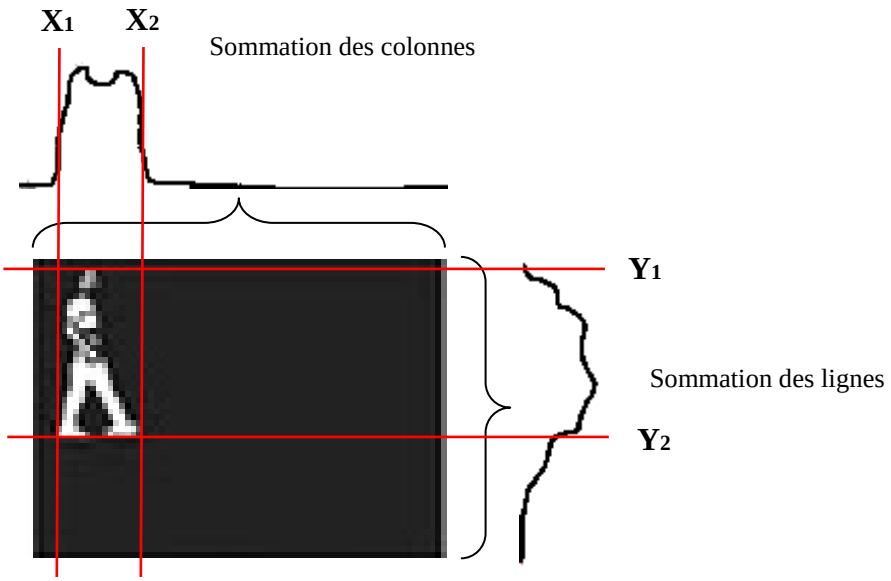
2- Filtrage

Normalement, à cette étape des éléments indésirables dans **I**r sont éliminés. Ces éléments indésirables peuvent être produits par le déplacement des objets dans la scène qui engendre de petits changements au niveau de la lumière. Ce qui produit souvent un effet de cascade sur la lumière environnante et sur les ombrages des objets environnant. *Cette étape n'est pas implantée ici dans le cadre de ce travail.*

3- Détection des limites par sommation des lignes et des colonnes :

Il faut maintenant tenter de découvrir à quel emplacement dans l'image il y a eu du mouvement. Le principe qui sera utilisé est très simple. Étant donné que tout ce qui bouge est blanc et que tout ce qui ne bouge pas est noir, il faut trouver en X et en Y les emplacements ou la luminosité est la plus forte.

Nous allons donc faire la somme des pixels sur chaque ligne et chaque colonne. Les lignes possédant un niveau de blanc très élevé devraient produire des niveaux importants de luminosité.



Il s'agit par la suite de trouver dans les graphiques l'endroit ou les cassures sont les plus importantes afin de trouver les bordures X_1, X_2, Y_1 et Y_2. Un seuil doit être déterminé afin de trouver les bonnes cassures dans les graphiques.

Étapes :

- Construction du kernel de soustraction des images pour l'étape 1. La valeur N doit être paramétrable.
- Construction du kernel de sommation des colonnes et des lignes avec extraction de X_1, X_2, Y_1, Y_2 pour l'étape 2. Le seuil doit être paramétrable.
- Implantation de la solution pour une webcam.