

TP 3 : DÉTECTION ET SUIVI DE MOUVEMENT DANS UNE VIDÉO

Groupe : **FOTSING SIKADIE Gervais p20**

BIAKOTA BOMBIA Cephass Herbert p21

Ce présent rapport sur ce travail explique la mise en place ou l'implémentation d'un programme ou d'un système de détection et suivi le mouvement des objets dans une vidéo. Ainsi, pour ce travail nous allons exposer de manière détaillée cette réalisation ainsi que les résultats des expérimentations obtenues

1- Détection du mouvement

Description détaillée

le système que nous avons mis en place pour la détection de mouvement dans une vidéo reçoit en paramètre ou argument) lors de l'exécution du programme 5 paramètres:

nbFrame: Le nombre de fenêtres capturées par une seconde: Ce paramètre permet de considérer quelques frames pour la détection et le suivi du mouvement. En faisant cela, on gagne en temps et en espace de stockage. Donc on observe rapidement les résultats de suivi.

- **seuilDiff:** Ce seuil est utilisé pour faire la différence entre la fenêtre courante et l'image de fond. Il s'applique à la fonction `threshold()`

- **seuilBruit:** Ce paramètre nous permet de déterminer le nombre de pixel minimal d'un objet en mouvement. Dans le cas où le nombre de pixel d'un objet est plus petit que ce seuil, nous allons considérer l'objet comme du bruit.

- **seuilCorrespondance:** Ce paramètre permet de déterminer dans quel distance on peut retrouver la correspondance des objets en mouvement entre deux frames capturées. Donc pour un objet qui se déplace, si la distance entre sa position dans la fenêtre précédente et sa position dans la fenêtre actuelle est inférieure à ce seuil, on garde le numéro de l'objet.

Fonctionnement du programme

Pour pouvoir utiliser du programme, il est nécessaire d'effectuer les actions à priori : il suffit de se positionner dans le répertoire « tp3 » par ligne de commande. Il faudra au préalable exécuter notre fichier « MakeFile » grâce à la commande « make » afin de créer l'exécutable et ensuite taper la Commande **`/detection_suivi_move nbImage nom_de_video seuilDiff seuilBruit seuilCorrespondance`** » Exemple : La commande « **`/detection_suivi_move video1.mpg 5 60 40 100`** »

1.1- Structure du programme

Notre programme est constitué de trois grandes parties que sont l'extraction d'arrière-plan, la détection de mouvements et l'amélioration de l'image de détection. Nous présentons en détail chacune de ces étapes ci-dessous.

- **Extraction de l'arrière-plan:** A ce niveau, nous exploitons un certain nombre d'images (paramètre nombre de séquences) de notre vidéo afin construire l'image d'arrière-plan parce que nous avons remarqué qu'il n'était pas toujours nécessaire de considérer la vidéo entière pour obtenir de bons résultats et cela nous permet également de réduire le temps de calcul nécessaire. L'algorithme utilisé se base sur la détermination de la médiane des valeurs des pixels. Pour n séquences d'images la valeur d'un pixel de l'arrière-plan correspond donc à la médiane des n valeurs possibles de ce pixel;

• **La détection de mouvement:** Ici, nous détectons tous les objets (les personnes) en mouvement dans la vidéo. Pour y parvenir, nous calculons la différence absolue entre chaque image de la vidéo et l'image d'arrière-plan. La fonction "absdiff" de opencv a été utilisée à cet effet. 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. Cette action a pour but de réduire en amont les sources de bruits dans l'image caractérisant le mouvement.

• **Amélioration de l'image décrivant le mouvement :**

Afin réduire un maximum les bruits et les trous dans l'image détectée on va tout d'abord appliquer la méthode canny d'openCV pour trouver les contours, ensuite à l'aide aux méthodes pyrUp et pyrDown on va lisser l'image et ainsi supprimer d'avantages de bruits. La méthode floodFill va permettre de relier les composantes connexes de l'image et de les colorier en blanc et enfin on utilise les fonctions érosion et dilation comme vu en cours pour supprimer les bruits et les trous de l'image.

1.2- Expérimentations

Dans cette section, nous avons conduit quelques expérimentations visant à observer la capacité de notre programme à détecter des mouvements dans une vidéo. La base de test utilisée est composée de vidéos provenant des bases INRIA et PORTUGAL disponibles à <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>.

Nos premières expérimentations ont portées sur l'extraction de l'arrière plan des vidéos. Il est à souligner que les images d'arrière-plan sont statiques durant toute l'expérimentation car les vidéos utilisées sont principalement des vidéos de caméras de surveillance dont le décor change peu dans le temps. Les images ci-dessous montrent des images d'arrière-plan obtenues en considérant différents nombres de séquences.

- Extraction d'arrière-plan avec 50 premières séquences



fig1 : image de background de Video1.mpg



image de background de Video2.mpg video2

Sur le schéma ci-dessus, pour la construction avec 50 frames, nous obtenons une bonne extraction d'arrière-plan au niveau de l'image de la vidéo1 (bon résultat) ainsi que l'apparition des personnes sur l'image de fond (mauvais résultat) de la vidéo2.

Extraction d'arrière-plan avec 200 premières séquences



fig2 : image de background de Video1.mpg



image de background de Video2.mpg video2

En considérant 200 séquences nous remarquons que l'image de la video1 reste correctement détectée comme précédemment et une amélioration dans l'extraction de l'arrière-plan principalement au niveau de l'image de la video2 dont les personnes sont toujours présentes mais nous remarquons qu'elles sont moins nettement visible ce qui confirme l'amélioration précédemment évoquée. Enfin.

Nous concluons que le nombre de séquences considérées influent nettement sur la qualité de l'extraction des arrière-plan. Pour ce fait, il n'est pas nécessaire de traiter toutes les séquences de la vidéo pour en extraire l'arrière-plan.

2. Analyses des résultats de la détection de mouvement

Le programme réussi bien à détecter des objets dans les vidéos. Nous présentons quelques bons et mauvais résultat de détection

Exemple1

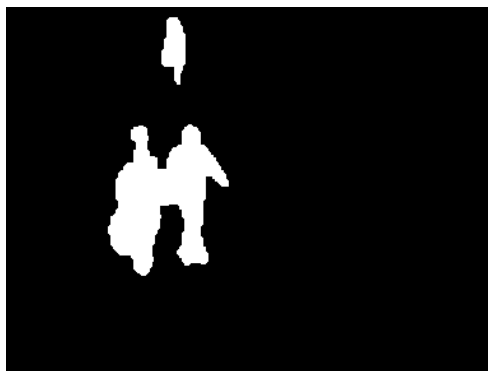


fig3 : Masque illustrant la personne détectée



image du frame

Dans les rectangles en rouge, on voit des objets détectés (fig3 :image frame). On peut voir sur cette image, d'autre partie blanche (les bruits) qui sont détectés comme objet en mouvements, et cela est due à la variation de luminosité sur le sol tout le long de la vidéo.

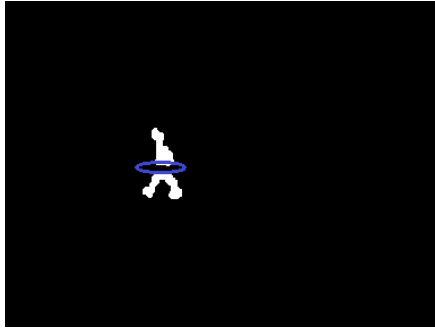


fig3 : image détectée



image du frame

Sur cette image on peut voir que l'homme qui se déplace n'est pas détecté entièrement (voir les cercles bleus). Cela peut s'expliquer par la différence des valeurs de pixels du pantalon de l'homme avec ceux de l'arrière plan qui sont presque identiques. D'où le vide en rouge qu'on peut voir sur l'image.

III. Suivi de mouvement

1. Conception du programme

Pour avoir de meilleur de résultat de suivi, on doit essayer de corriger le maximum de problème rencontré lors de la détection. Par exemple pour éviter que la variation de lumière soit suivi comme un objet dans la vidéo, lors du calcul des boites englobantes minimales nous allons compter le nombre de pixel d'un objet détecté (contour) et si ce nombre est inférieur au seuilBruit alors l'objet est considéré comme du bruit et ne sera pas stocké dans la liste des boites englobantes.

Détermination de la boite englobante

Notre structure de boite est composé de 2 points opposés pour dessiner la boite et aussi d'un numéro qu'on associera à chaque boite déterminé. Pour déterminer les boites englobantes minimale entourant les objets,

- on déterminer les contours dans l'image détectée avec la fonction `findContours()`
- Pour chaque contour déterminé, on dessine le rectangle qui l'entoure en utilisant la fonction `boundingRect()`

2. Suivi de mouvement

Pour le suivi de mouvement, nous allons utilisé la position du centre de gravité de chaque boite englobante. Ensuite en utilisant le filtre de Kalman, nous pouvons prédire, mesurer et corriger le suivi du mouvement d'un objet

Pour suivre particulièrement un objet, il est très important de déterminer dans l'image courante quel est la boite englobante correspondante à l'objet dans l'image capturée précédente. Le filtre de Kalman a une fonction de prédiction de la position de l'objet dans le frame suivant. Le programme cherche un objet voisin qui est le plus proche parmi tous les autres objets de voisin (en utilisant le seuil de correspondance), donc cet objet trouvé est l'objet correspond. Ce programme ajoute le modèle de vitesse des objets, donc on essaye de suivre correctement l'objet.

3. Expérimentation et analyse des résultats

Illustration 1: Suivi du mouvement de deux personnes sans croisement de

trajectoires ainsi que d'une seule personne avec trajectoire simple et complexe

image1



image2



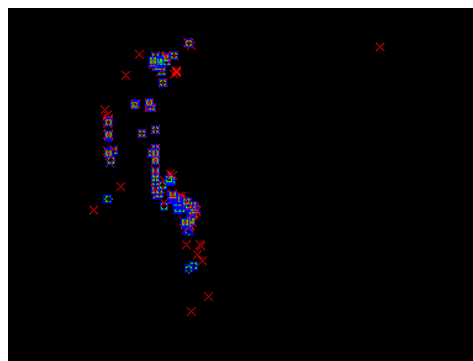
image3



image4



image5



Résultat du suivi de toute la vidéo

Ces images présentent principalement le suivi des personnes qui se déplacent de bas du couloir pour se diriger vers le fond du couloir de manière automatique (Voir image 1). Ensuite cette personne entre de côté gauche se dirige vers le bas (image3) mais elle s'arrête en mi-chemin et repart comme elle est entrée. Donc on a bien un mouvement complexe. L'image du suivi (image 5) nous montre qu'on a réussi à suivre parfaitement le mouvement des personnes.

On constate aussi que les trois types de point (point prédit, point mesuré et point corrigé) sont assez proches. Donc le filtre Kalman fonctionne bien pour ce cas. Comme nous l'avons mentionné plus haut, mais la variation de lumière sur le sol est parfois considéré comme des objets en mouvement. Ce qui explique que sur l'image 5 (partie en Vert) une légère erreur de suivi correspondant à l'objet 5. Analyse d'une trajectoire simple En jaune (sur l'image 5) il s'agit d'une personne qui quitte du bas de la scène marche jusqu'en haut de la scène et (Trajectoire simple). Donc on peut constater que notre programme est capable de bien suivre le mouvement des personnes ayant des trajectoires simples ou complexes et sans croisements entre ces personnes

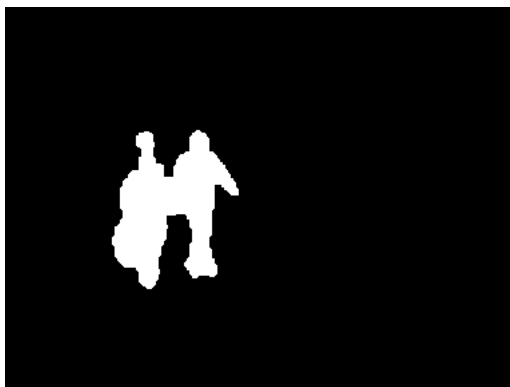
→ Illustration 2: Suivi de plusieurs objets en mouvements avec croisement de trajectoire et occlusion

Dans cette vidéo, on a principalement 3 objets dont on va suivre à savoir 2 personnes se déplaçant tout droit sur le couloir vers le bas de la scène



Image de fond construite

La première chose qu'on remarque dans cet exemple c'est le fond d'écran sur lequel on trouve aucun objet en déplacement. L'explication de cette absence est dû sous le fait qu'au debut de la vidéo nous avons aucun objet en déplacement.



Dans l'image 2 on constate à 2 objets en mouvement sont bien détectés, mais considéré comme étant un seul objet. Cela est dû à l'action de prétraitement car les méthodes d'érosion et dilatation font en sorte que les 2 objets deviennent des composants connexes.

La second raison est l'arrière-plan qui influe sur la détection des boîtes englobantes minimales et qui pour cet exemple rendra difficile le suivi de mouvements.



fig4 : image1



image2

Dans l'image 1 et 2 ci dessus, on suit le mouvement d'une personne qui quitte le bas de la scène vers le côté gauche de scène. Sur l' image 2 on a un cas d'occlusion pour lequel le programme va perdre la trace des objets suivis et ne va plus en détecter qu'un seul (voir image 2).



fig4 : image3



image4



image5

Dans l'image 3 et 4 ci-dessus on peut également voir une autre personne qui se déplace de bas de la scène vers le haut droit. A cause de l'occlusion (image4) il est difficile de suivre cet objet. Ce qui fait dans l'image 5 la personne n'est détectée, cependant si on regarde l'image de tracking on remarque que son mouvement est tout de même estimé. en vert sur l'image6 de suivi correspondant à la vidéo).

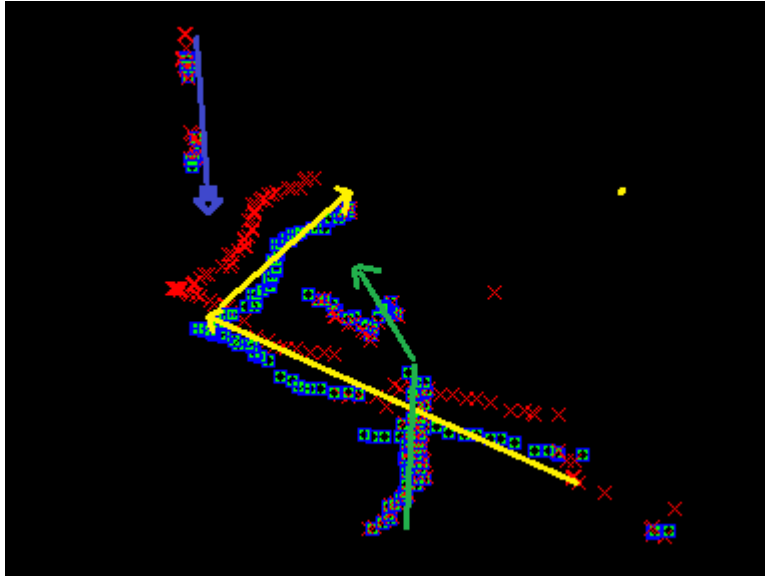


Image6

Pour le résultat final du suivi, on peut bien observer tout ce qui est décrit dans les paragraphes précédents. On a 3 principaux mouvement qui sont représenté par les droites (jaune), (bleu) et (vert).

Pour le mouvement correspondant à la droite (bleu) est celui de l'objet détecté qui part de haut de scène vers le bas de la scène. La droite jaune correspond à un mouvement linéaire du 2^e objet tout en changeant de direction. La droite verte correspond à au mouvement de l'objet3

IV. Conclusion

Au cours de ce TP, nous avons concevoir un programme de détection et de suivi de mouvement en utilisant certaines fonctions existantes dans Opencv. On a pu montrer à travers des analyses faites On peut voir grâce aux différents scénarios sur lesquels on a exécuté le programme que la détection comme le suivi de mouvements dépendent de nombreux paramètres et de techniques utilisées. Un des facteurs important est la construction de l'arrière-plan, dans ce tp nous avons utilisé la méthode vue en cours, c'est à dire utiliser la valeur médiane des pixels des images pour construire le fond. Cependant cette méthode est loin d'être parfaite et peut entraîner dans certains cas une mauvaise détection des objets et leur suivi. C'est le cas dans le dernier exemple. Le second facteur est la présence de luminosité dans la scène. Ce facteur influence aussi sur le nombre d'objet à suivre.