

Version du	17/12/09	RCSO-TIC ViBIH 2 Lucas-Kanade	08IT16-DT01
Créé par	MID		

Table des matières

1. INTRODUCTION	2
2. EXEMPLES DE DETECTION DE MOUVEMENTS AVEC LUCAS-KANADE	3
2.1 IMAGE DE FLUX OPTIQUE DENSE LUCAS-KANADE.....	3
2.2 IMAGES DE FLUX OPTIQUE PARTIEL PYRAMIDAL LUCAS-KANADE	3
3. AMELIORATION DE LA DETECTION.....	4
3.1 SOUSTRACTION DE FOND AVEC LUCAS-KANADE PYRAMIDAL ET PARTIEL	4
3.2 UTILISATION DE LA SEGMENTATION	5
4. CONSTRUCTION DE CARTES DE DEPLACEMENT.....	6
4.1 CARTES DES NOMBRES ET DE L'AMPLITUDE DES DEPLACEMENTS	6
4.2 CARTE DE « ROSES DES DEPLACEMENTS ».....	7
5. CONCLUSION	7
6. ANNEXES.....	8
6.1 TOUCHES UTILISEES PAR LE PROGRAMME D'ESSAI LUCAS-KANADE	8
6.2 AFFICHAGES DU PROGRAMME D'ESSAI LUCAS-KANADE.....	8
6.3 OPTIONS DU PROGRAMME D'ESSAI LUCAS-KANADE.....	9

1. Introduction

Le suivi de mouvement, entre deux images, peut être effectué principalement de deux manières :

- a) **Suivre le déplacement de chaque pixel de l'image** et y associer une vitesse (représentée effectivement par le déplacement entre l'image précédente et l'image courante). Cette technique est appelée **flux optique dense**. La difficulté de cette technique est qu'elle produit un grand nombre de points en mouvements. Détecter correctement les déplacements est difficile et coûteux en temps de calcul.
- b) **Suivre le déplacement de points (zones) caractéristiques**. Cette technique est appelée **flux optique partiel**.

La méthode Lucas-Kanade implémentée dans OpenCV permet de tester ces deux techniques. D'autre part, la technique de flux optique partiel Lucas-Kanade implémentée dans OpenCV utilise une version pyramidale qui améliore la détection de mouvements rapides (les mouvements sont détectés dans une pyramide d'images, en commençant par les images de basse résolution au sommet, et en poursuivant vers le bas où il y a les images détaillées, cf. figure ci-dessous).

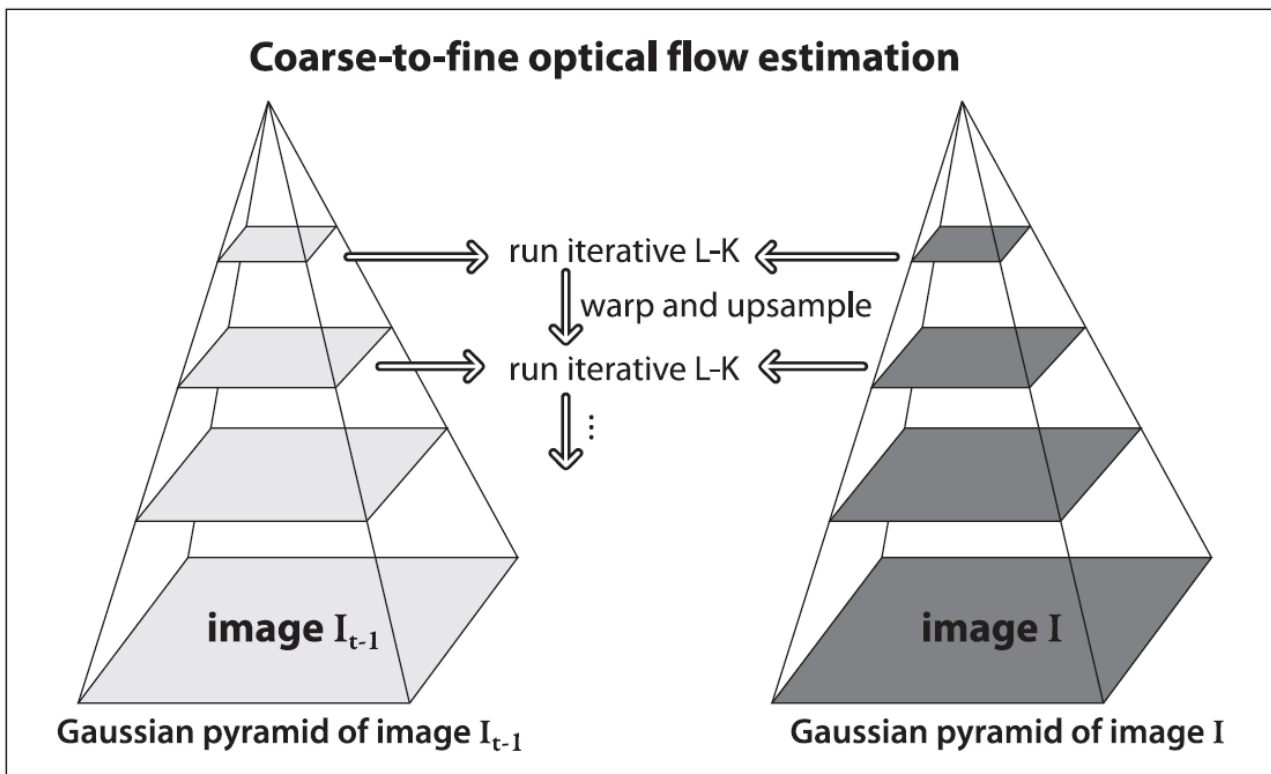
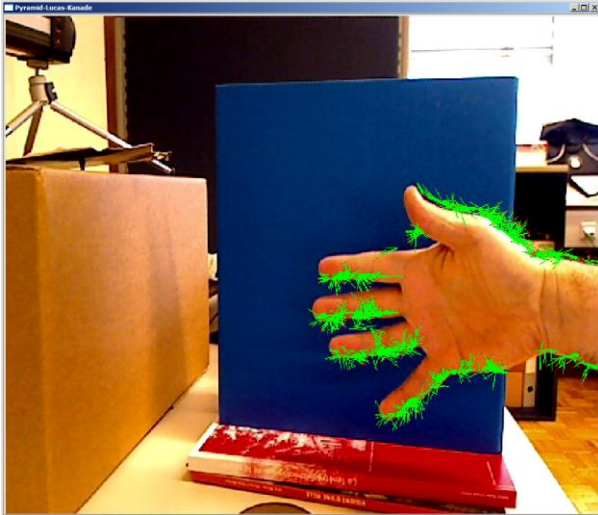


Figure 10-9. Pyramid Lucas-Kanade optical flow: running optical flow at the top of the pyramid first mitigates the problems caused by violating our assumptions of small and coherent motion; the motion estimate from the preceding level is taken as the starting point for estimating motion at the next layer down

Figure du livre « O'Reilly : Learning Opencv »

2. Exemples de détection de mouvements avec Lucas-Kanade

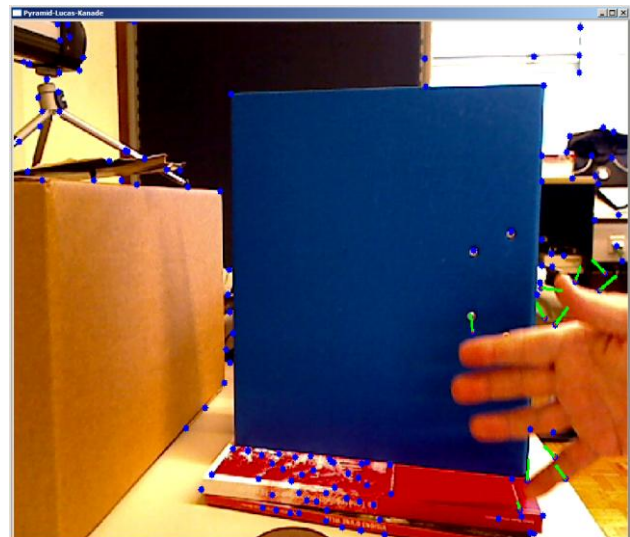
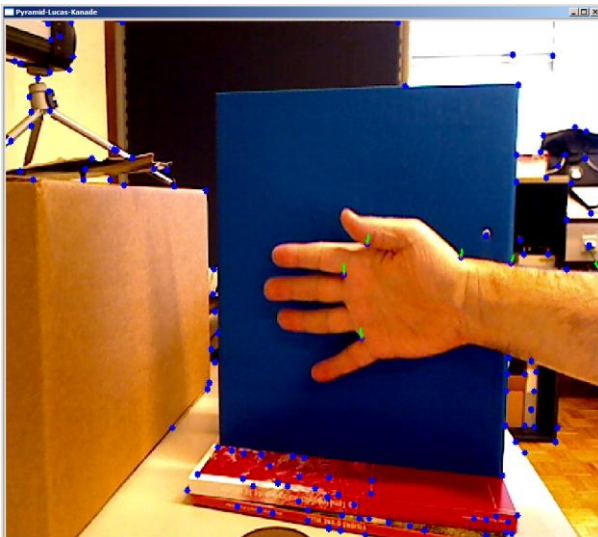
2.1 Image de flux optique dense Lucas-Kanade



Méthode OpenCV utilisée : `cvCalcOpticalFlowLK()`.

Dans cette image, on voit clairement **un grand nombre de vecteurs de déplacements** (en vert). D'autre part, la direction de **beaucoup de ces vecteurs ne correspond pas au déplacement réel** de la main vers le bas.

2.2 Images de flux optique partiel pyramidal Lucas-Kanade



Méthodes OpenCV utilisées : `cvGoodFeaturesToTrack()` et `cvCalcOpticalFlowPyrLK()`.

Dans ces images, les points caractéristiques (d'intérêt, de `cvGoodFeaturesToTrack()`) sont affichés en bleu et les mouvements détectés (avec `cvCalcOpticalFlowPyrLK()`) en vert. On constate, évidemment que les surfaces unies n'ont pas de points caractéristiques. Dans l'image de gauche, on voit que **peu de mouvements (vers le bas) sont correctement détectés devant une surface unie**. Par contre, on voit avec l'image de droite que **de mauvais déplacements peuvent être détectés devant des surfaces non unies**.

3. Amélioration de la détection

Comme indiqué dans l'introduction, on voit que détecter correctement les déplacements est difficile voire impossible et couteux en temps de calcul avec le flux optique dense. Par contre, la détection avec flux optique partiel pyramidal donne de bons résultats si le fond est uni. C'est donc cette détection que l'on va utiliser et améliorer en utilisant une soustraction de fond.

3.1 Soustraction de fond avec Lucas-Kanade pyramidal et partiel

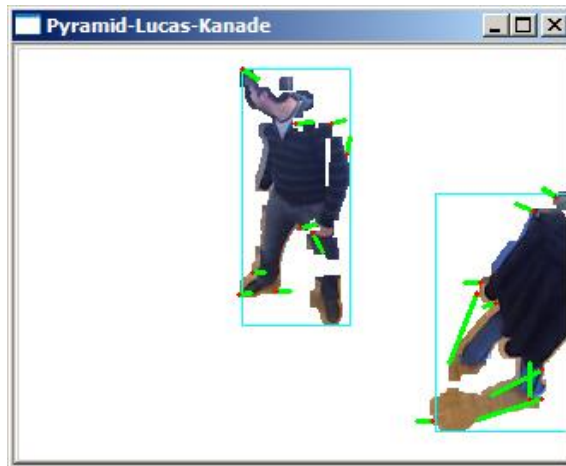
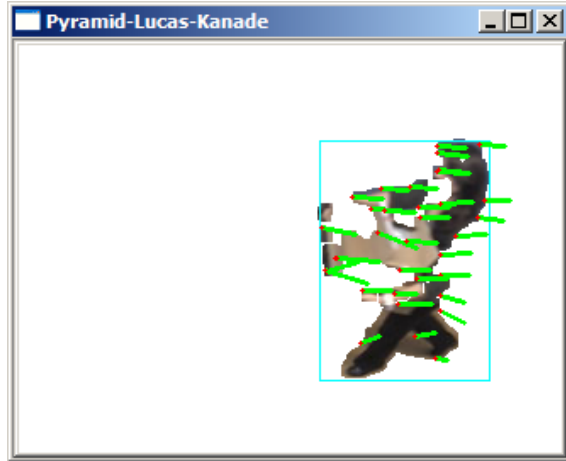
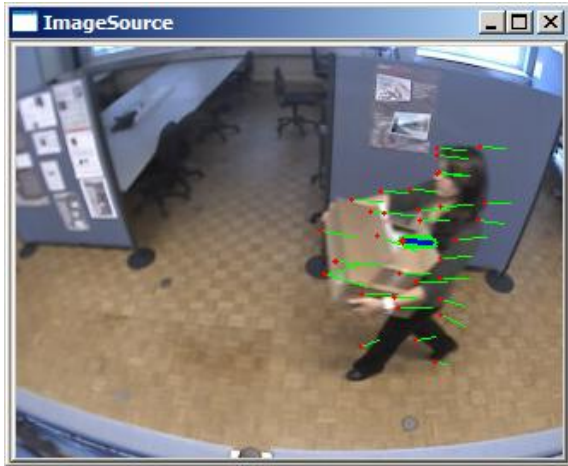


- Haut gauche : masque détecté (par soustraction de fond) de la personne en déplacement.
- Bas gauche : idem, mais avec le masque dilaté.
- Haut milieu : personne et mouvements détectés.
- Bas milieu : personne détectée (en gris) par soustraction de fond dans l'image actuelle.
- Bas droite : idem mais pour l'image précédente.
- Haut droite : image source

Comme on le voit sur l'image en haut au milieu, des mouvements sont affichés en violet. Ce sont les mouvements dont la destination n'est pas incluse dans le masque dilaté (bas gauche). C'est probablement des mouvements parasites détectés par la méthode Lucas-Kanade, on ne les utilisera donc pas.

3.2 Utilisation de la segmentation

Afin de séparer les mouvements de personnes ou objets différents, on ajoute la segmentation.



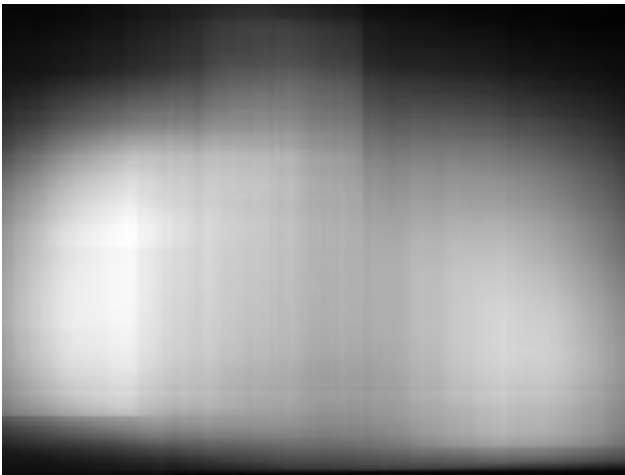
La segmentation est visible dans les images de droite.

A noter que les mouvements moyens des personnes sont affichés avec un trait bleu foncé entouré de vert dans les images source à gauche.

4. Construction de cartes de déplacement

Afin d'analyser les déplacements, on va s'intéresser au sens de ces déplacements que l'on classera par N divisions d'angles, ceci pour chaque pixel de l'image. Dans notre cas, N équivaut à 8 ce qui nous donnera 8 cartes avec des déplacements classés à 0°, 45°, 90°, 135°, 180°, 225°, 270° et 315°. Pour chaque pixel de l'image et pour chacun de ces angles, on enregistre le nombre de déplacements et l'amplitude des déplacements. En utilisant ces données normalisées, on peut construire des cartes pour le nombre des déplacements ainsi que pour la somme des amplitudes.

4.1 Cartes des nombres et de l'amplitude des déplacements

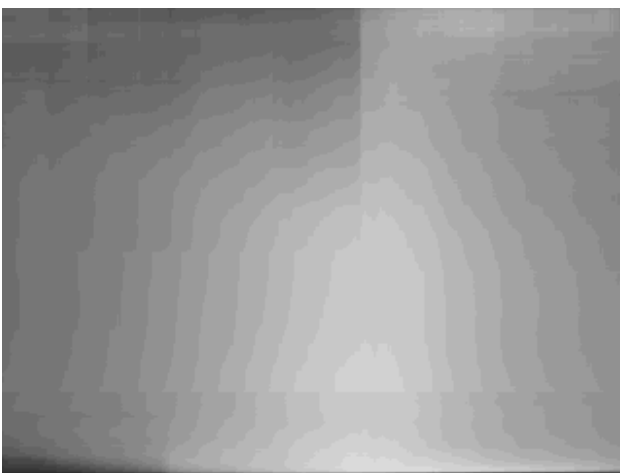


Nombre de déplacements à 0°.

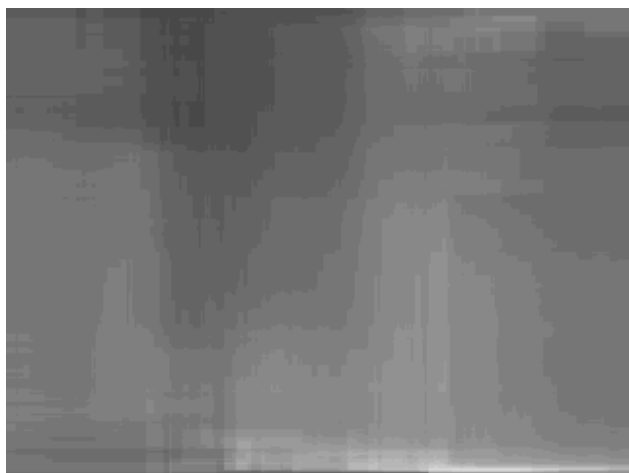


Nombre de déplacements à 45°.

On voit très clairement que dans ces exemples ci-dessus, qui correspondent au champ de vue des images des chapitres précédents, que les déplacements à 0° (Est) sont très nombreux (noir correspond à aucun déplacement, blanc correspond au nombre de déplacement maximum) dans la quasi-totalité de l'image, alors qu'il n'y a pratiquement pas de déplacements à 45° (NE).



Amplitudes à 0°.

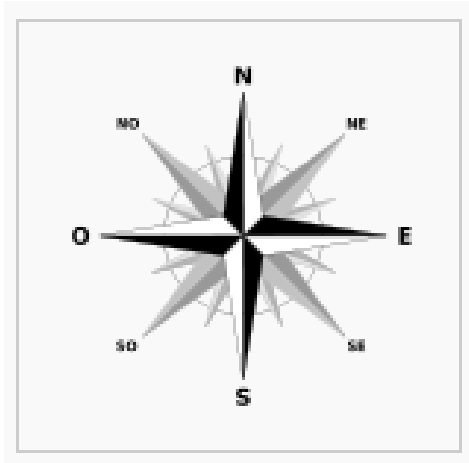


Amplitudes à 45°.

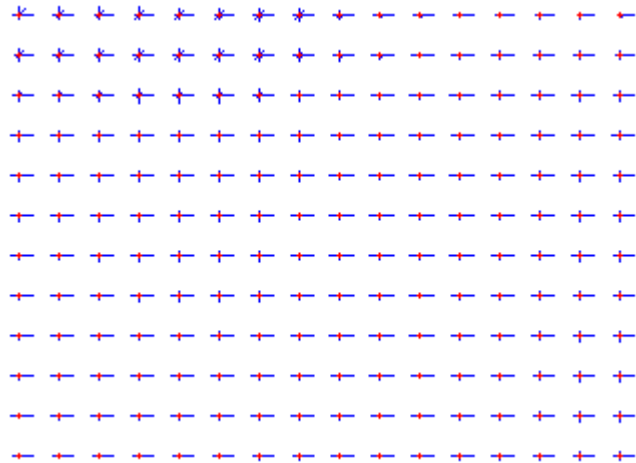
Les cartes des amplitudes n'apportent elles pas beaucoup d'information utiles si ce n'est que les amplitudes sont assez similaires pour les différents angles. On remarque toutefois dans ce cas, une amplitude moyenne un peu plus élevée au centre de l'image à 0°.

4.2 Carte de « Roses des déplacements »

De façon à mieux se représenter les déplacements selon les angles dans l'image, une carte de « rose des déplacements » (selon l'idée de la rose des vents) est affichée par petite zone de l'image. L'amplitude de chaque sens de la « rose des déplacements » dépend du nombre de déplacements dans le sens (normalisé pour la zone).



Rose des vents (Source : Wikipédia)



Carte de « Roses des déplacements »

On voit clairement dans cette représentation que les déplacements à 0° (E) et 180° (O) sont majoritaires, ce qui correspond bien au couloir filmé.

5. Conclusion

Détecter correctement les déplacements est difficile voire impossible et couteux en temps de calcul avec le flux optique dense de Lucas-Kanade. Par contre, la détection avec flux optique partiel pyramidal de Lucas-Kanade donne de bons résultats si le fond est uni. En ajoutant une soustraction de fond et une segmentation les résultats sont bons, indépendamment du fond.

6. Annexes

6.1 Touches utilisées par le programme d'essai Lucas-Kanade

ESC : quitter le programme.

n : mode « nuit » actif ou non

y : mode pyramidal ou dense

b : soustraction de fond active ou non

q : sauvegarde et affichage des cartes de nombre de déplacements par angle

w : sauvegarde et affichage des cartes d'amplitude des déplacements par angle

l / s : chargement / sauvegarde des données binaires des cartes

i : mode image par image actif (ESPACE : prochaine image) ou non

Avec mode pyramidal actif :

m : auto-détection des mouvements active ou non

d : affichage des mouvements ou non

p : affichage des points d'intérêt ou non

Sans le mode auto-détection des mouvements :

r : initialisation des points d'intérêt

c : effacement des points d'intérêt

click gauche : ajout ou suppression d'un point d'intérêt

6.2 Affichages du programme d'essai Lucas-Kanade

Les modes activés par les touches y et m sont affichés dans la console ainsi que le nombre de points d'intérêt et le nombre d'images par seconde traitées par le programme.

Les déplacements (vecteurs de vitesse) sont affichés en vert, avec la destination affichée par un point rouge. La segmentation est affichée par un (des) rectangle(s) bleu clair et les points d'intérêt sont affichés en bleu foncé dans l'image nommée « Pyramid-Lucas-Kanade ». Les mouvements moyens des personnes sont affichés avec un trait bleu foncé entouré de vert dans l'image source. Dans celle-ci on a aussi les affichages du nombre d'images traitées, le nom de fichier de l'image en cours (si traitement d'images enregistrées précédemment), la moyenne des niveaux de gris de l'image ainsi que la déviation standard des niveaux de gris (utilisées pour stopper l'enregistrement (en live) si ces valeurs sont trop basses), ainsi qu'un petit disque rouge lorsque l'enregistrement (en live) est en cours quand des déplacements sont détectés.

6.3 Options du programme d'essai Lucas-Kanade

Voici des options qui sont dans le fichier « param.h » :

```
//=== Path and extensions for images loading and saving =====
#define MAPS_PATH_NAME          "D:\\ViBIH2\\Lucas-Kanade\\Medias\\Cartes"
#define SRC_IM_PATH_NAME       "D:\\ViBIH2\\Lucas-Kanade\\Medias\\Flux optique\\Seq1\\ImgCam"
#define SRC_RECT_RELPATH_NAME  "..\\Rect\\" // Related to SRC_IM_PATH_NAME
#define DST_IM_PATH_NAME       "D:\\ViBIH2\\Lucas-Kanade\\Medias\\Test17"
#define IM_FILE_EXT            "png"
#define FULL_IM_FILE_EXT       "*.png"

#define NB_IMAGES_BACK_SUBS    120 // Nb images without move necessary for background subs. init.
#define NB_IMAGES_AFTER_MOVE  1 // Nb images to save after move detected

#define MEAN_GREY_LIMIT        25 // Mean grey limit of pixels to process source image
#define STD_GREY_LIMIT         15 // Std_dev grey limit of pixels to process source image

#define IP_CAMERA              true // false: xxxTestLucasKanade() use his webcam true: other cam (IP)
#define IM_FROM_FILES          true // false: read image from camera true: read images from files
#define LIVE_SAVING            true // false: no image saving true: saving if (IM_FROM_FILES == false)
#define CONTINUE_SAVING        false // false: when program is launched, new saved process is number 1
// true: when program is launched, new saved process follow the last saved process
// !!! Available if IM_FROM_FILES is false !!!
#define USE_SEGMENTATION       true // true: use segmentation rectangles for moves records
```

Dans le fichier « PyrLucasKanade.cpp », d'autres options sont configurables :

```
//--- Global image size for whole process and saved maps
#define PROCESS_IMAGE_WIDTH    320
#define PROCESS_IMAGE_HEIGHT   240
```

Ce qui permet d'avoir des cartes de taille indépendante de l'image source.

```
//--- Size for background substraction
#define MASK_BGSUB_IMAGE_WIDTH 320
#define MASK_BGSUB_IMAGE_HEIGHT 240
```

Ce qui permet de ne pas avoir un temps de process trop élevé car le temps de la soustraction de fond augmente de façon exponentielle avec la grandeur de l'image, 320 par 240 est un peu la limite acceptable.

```
//=== Lucas-Kanade Tracking =====

const int TRACK_WIN_SIZE = 10;
const int MAX_TRACKED_POINTS = 500;

//=== Moves record structures and defines =====

const int NB_MOVE_SENS = 8; // 8 sens --> Delta angle : +/- 22.5°
const int DISP_MOVE_DELTA = 20; // dx and dy to display moves

//=== Rectangle for persons detected with segmentation =====

#define PERS_RECT_NB 10 // Nombre max. de personnes suivies dans l'image
```