

TP2 : Détection de mouvement dans les vidéos

- Algorithme de seuillage :

Pour l'algorithme de seuillage je me suis basé sur le schéma donné.

Je fais la différence pixel par pixel de la première image à l'image actuelle en niveau de gris (image convertie avant de passer dans la fonction) , puis si le résultat est supérieur à la valeur de T (Threshold) que j'ai mit à 10 .

Voilà donc le résultat :

```
cv::Mat compare_frame(cv::Mat first_frame, cv::Mat current_frame)
{
    cv::Mat mask(first_frame.rows, first_frame.cols, CV_8UC1);
    int pixel;

    for (int i = 0; i < current_frame.rows ; i++) // pour chaque ligne
    {
        for (int y = 0; y < current_frame.cols ; y++) // et chaque colonne
        {

            pixel = abs((int)current_frame.at<unsigned char>(i, y)
                        - (int)first_frame.at<unsigned char>(i, y));
            if(pixel > 10)
                mask.at<unsigned char>(i, y) = 255;
            else
                mask.at<unsigned char>(i, y) = 0;

        }
    }
    return mask;
}
```

- Comparaison des Algorithmes :

Afin de réaliser une comparaison entre les algorithmes je teste sur une frame donnée la vitesse pour créer le masque .

Pour cela j'utilise la bibliothèque chrono. et je récupère l'heure avant et après l'utilisation de la fonction. Puis je stocke la durée.

Et pour la fonction maison on ajoute aussi la conversion en niveau de gris qui prend un peu de temps aussi.

```

//avec Algo MOG2
    t1 = std::chrono::high_resolution_clock::now(); //heure de debut
    pBackSub_MOG2->apply(frame, mask_MOG2);
    t2= std::chrono::high_resolution_clock::now(); //heure de fin
    time_algo_MOG2 = t2 - t1; //calcul durée

//avec Algo KNN
    t1 = std::chrono::high_resolution_clock::now();
    pBackSub_KNN->apply(frame, mask_KNN);
    t2 = std::chrono::high_resolution_clock::now();
    time_algo_KNN = t2 - t1;

//avec Algo maison
    t1 = std::chrono::high_resolution_clock::now();
    cvtColor(frame, frameG, COLOR_BGR2GRAY);
    mask = compare_frame(first_frameG, frameG);
    t2 = std::chrono::high_resolution_clock::now();
    time_algo_maison = t2 - t1;

```

A la fin on se retrouve avec des valeur comme celle ci :

```

l'agorithme maison s'execute en : 0.0239699s
l'agorithme MOG2 s'execute en : 0.0161518s
l'agorithme KNN s'execute en : 0.0706055s

```