***calculeHistogram(Mat & src)***

Entrée: objet choisit

Sortie : les valeurs de couleur de trois canaux d’objet choisit

(thresh\_red1, thresh\_red\_2, thresh\_green\_1, thresh\_green\_2, thresh\_blue\_1, thresh\_blue\_2)

***generateBinaryImage(Mat& tmp)***

Entrée: image <<tmp>>

Sortie : image binaire

Explication : diviser l’image <<tmp>> aux trois canaux (Bleu, Vert, Rouge). Binarisation par trois canaux.

Calculer intersection de trois canaux

**Initialisation :**

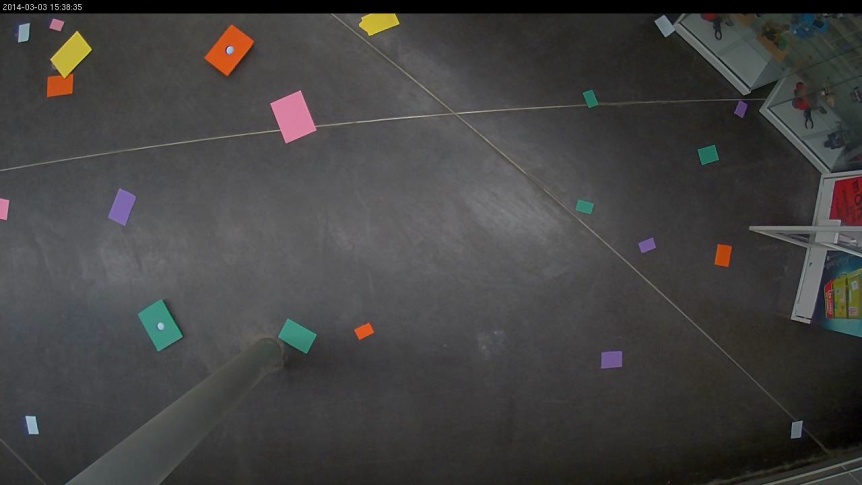
1. calculer les couleurs de trois canaux(BGR) d’objet choisit
2. binariser première image de vidéo(binary\_fond)
3. Par exemple, on choisit un objet vert.

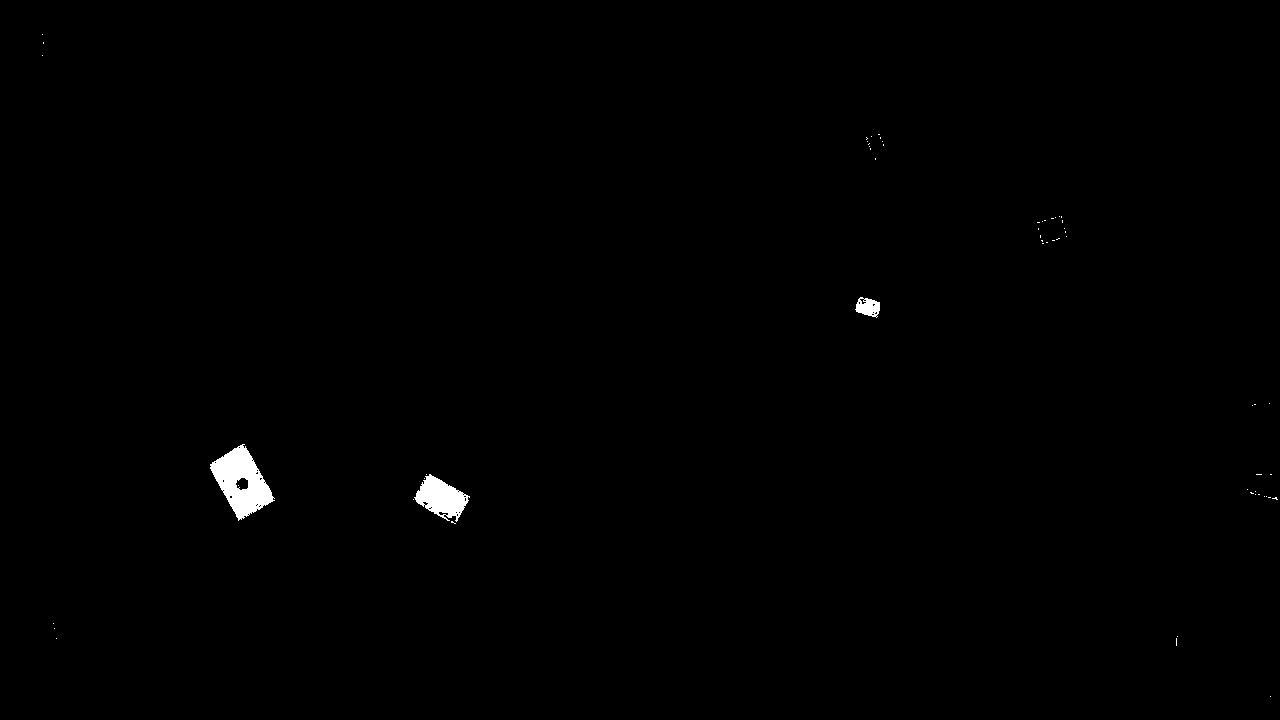


On obtient 6 valeurs de couleurs de cet objet :

histogram.JPG

1. Générer image binaire de première image de vidéo





**Algorithme de suivie :**

Entrée : image de frame courant et numéro de frame courant

Sortie : trajectoire

1. Générer image binaire d’image courant (binary\_frame)

2. Récupérer l’image proper, seulement avec objet choisit

(binary\_fond + binary\_frame) – binary\_fond = image de l’objet choisit(img\_act)

3. Faire l’érosion, pour supprimer les parties qu’on ne veut pas, on obtient une image « clean\_act »

4. Rechercher les contours d’objet choisit

5. Si (contours.size()>=1)

Localisez le center de objet et trouver le rayon d’objet

Si (trajectoire n’est pas vide)

Si(les distances entre objet courant et objet de dernier frame < déplacement max)

Ajouter le point à trajectoire, modifier l’objet courant et dessiner le centre et le cercle d’objet

Sinon

Si (pre\_img == true, le premier center apparaît)

pre\_img = false ;

Ajouter le point à la trajectoire

Si (Dans le dernier frame, l’objet déplace très rapidement)

Re déterminer la distance et Ajouter le point à la trajectoire

Sinon

Si (l’objet réapparaît)

Ajouter le point à la trajectoire

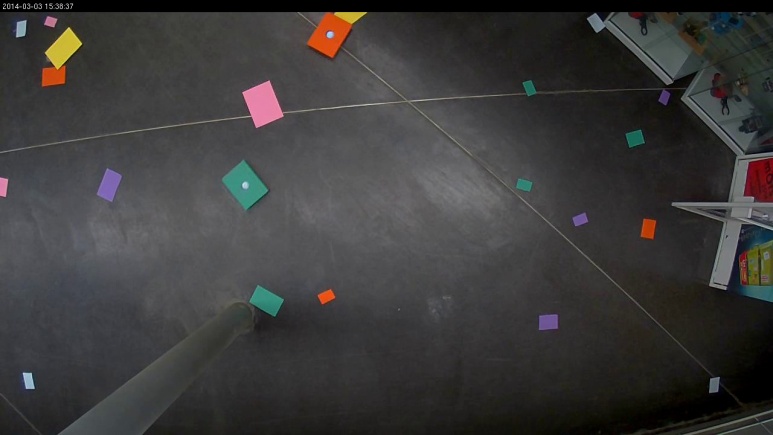
Sinon

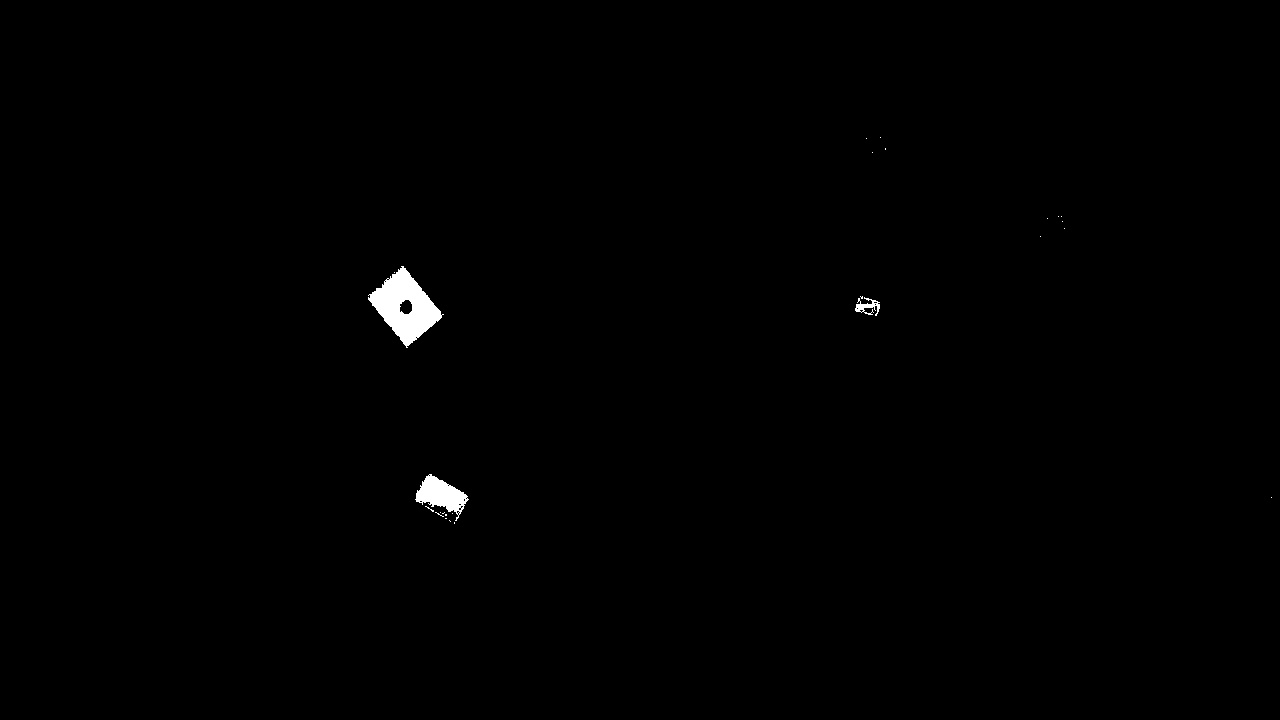
Modifier l’image binaire « «binary\_fond » et count\_refond++

6. Sinon

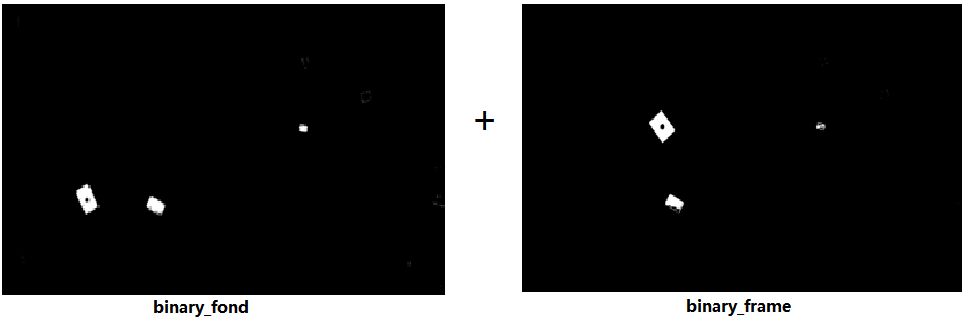
Vider la trajectoire

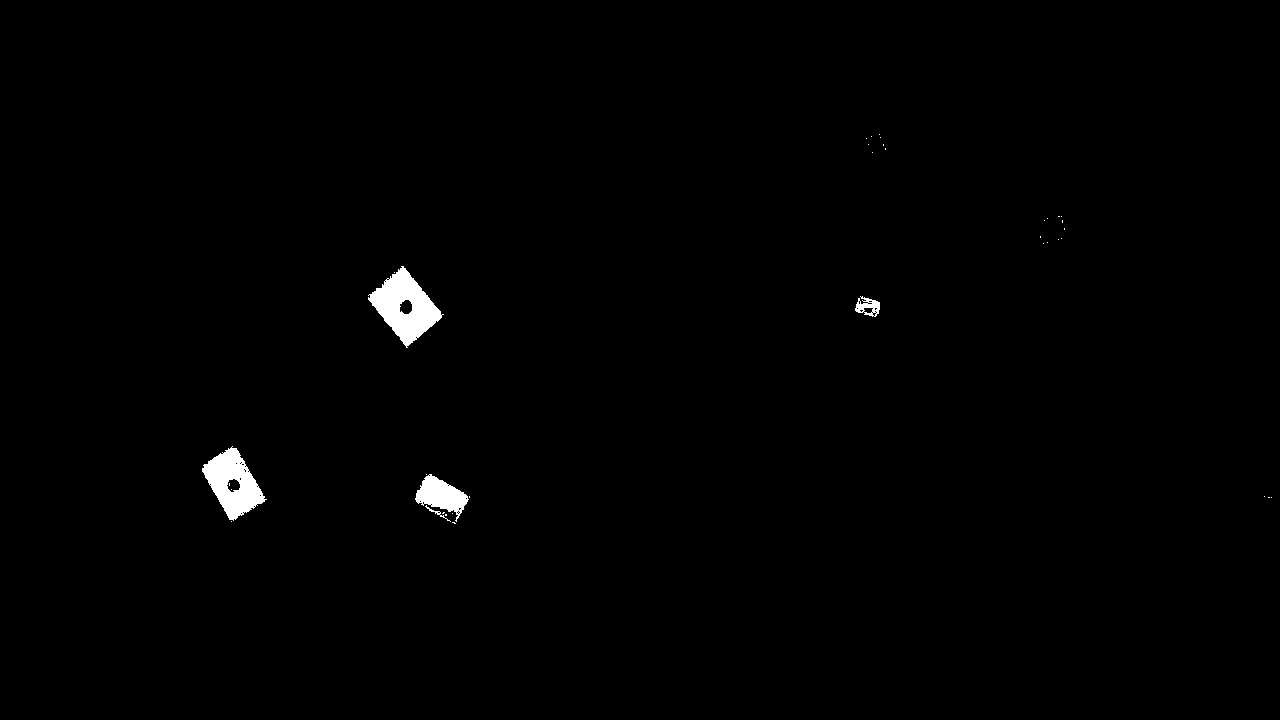
1. Générer l’image binaire du frame courant. L’objet choisit est objet vert. Donc les parties blanches sont les objets verts.





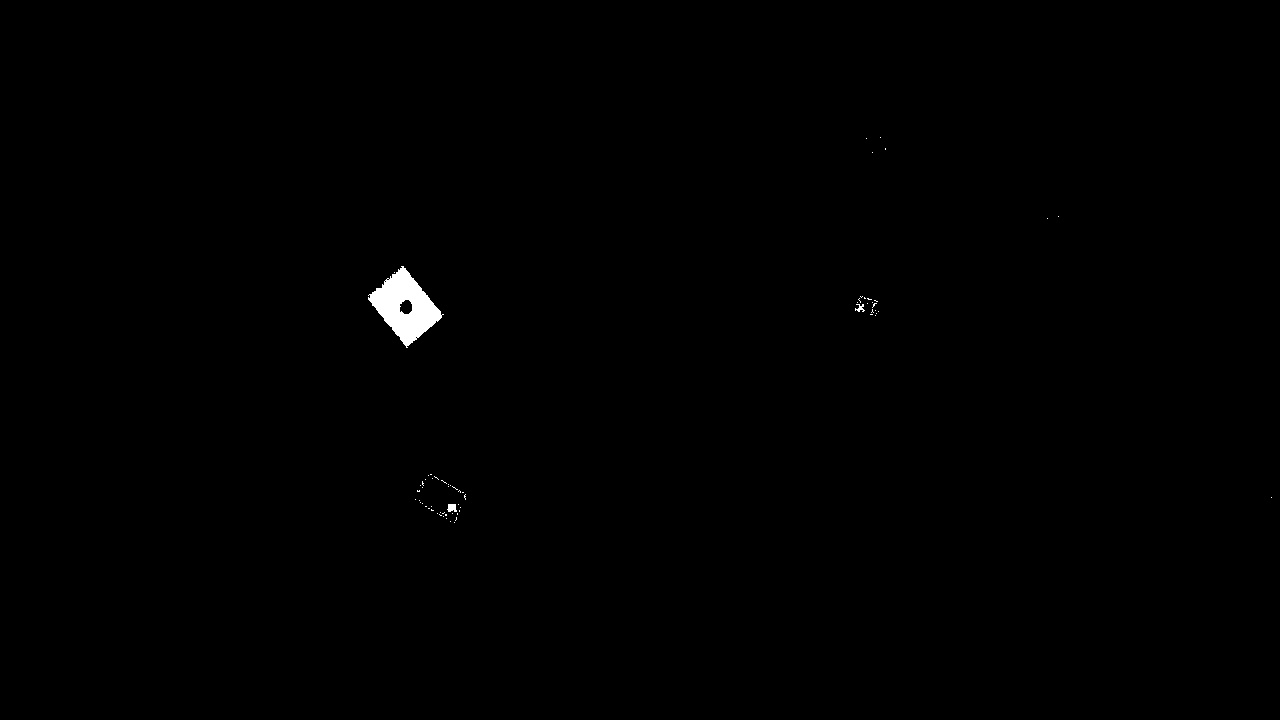
1. (binary\_fond + binary\_frame) – binary\_fond = image de l’objet choisit(img\_act)





binary\_fond + binary\_frame

(binary\_fond + binary\_frame) – binary\_fond



l’objet choisit

1. Après l’érosion, on obtient une image plus propre.

