

# **Comptage de point automatisé au basketball**

Hugo Berthod  
Juin 2023

# Sommaire

- 1) Présentation du problème
- 2) Suivi de balle
- 3) Détections des paniers
- 4) Algorithme
- 5) Réponse à la problématique
- 6) Annexe

# Présentation du problème



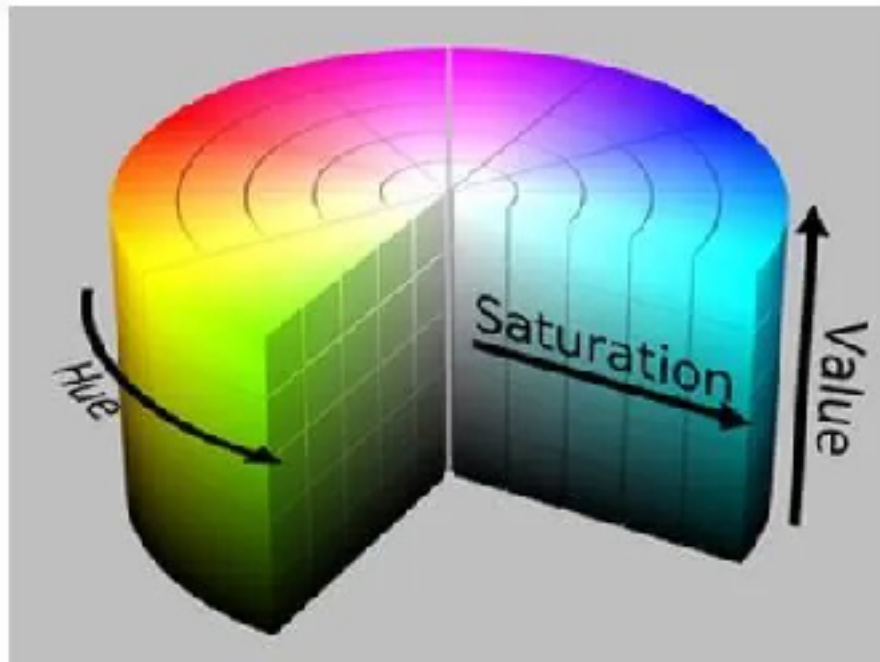
[bebasket.fr](http://bebasket.fr)



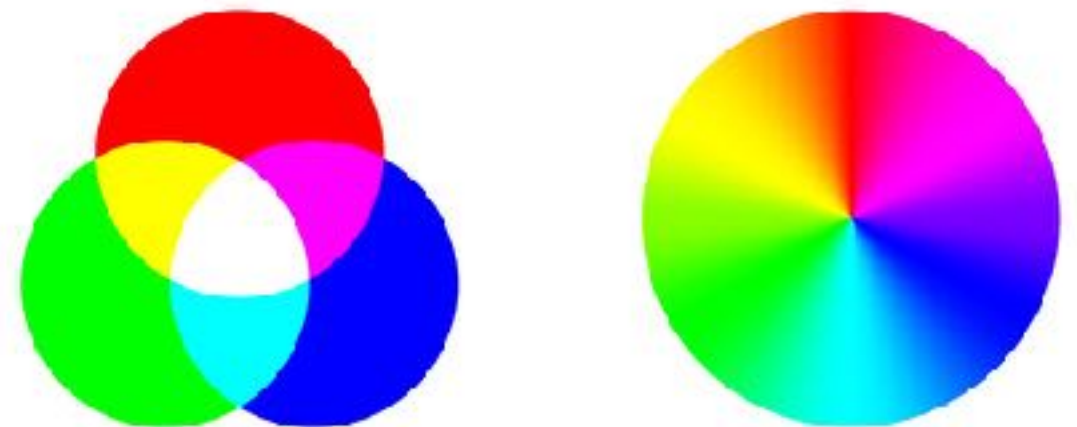
[bodet-sport.com](http://bodet-sport.com)

# Suivi de la balle

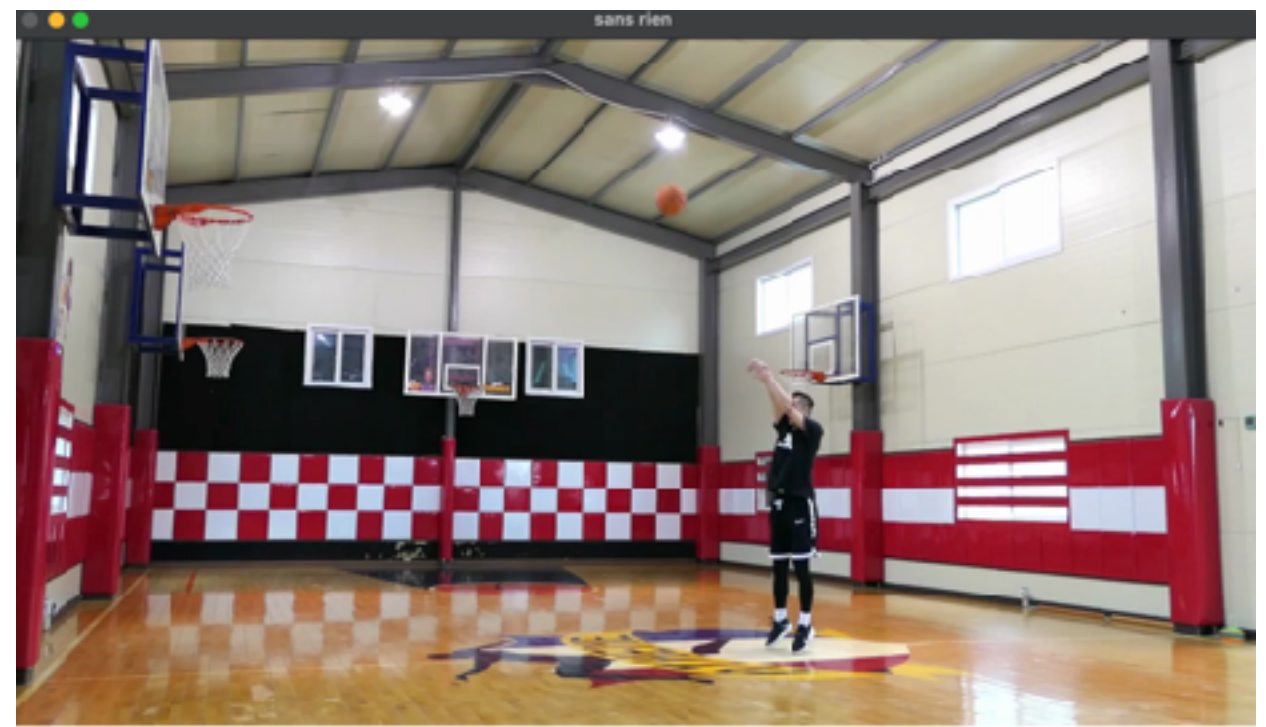
**HSV : Hue Saturation Value**  
(Teinte saturation valeur)



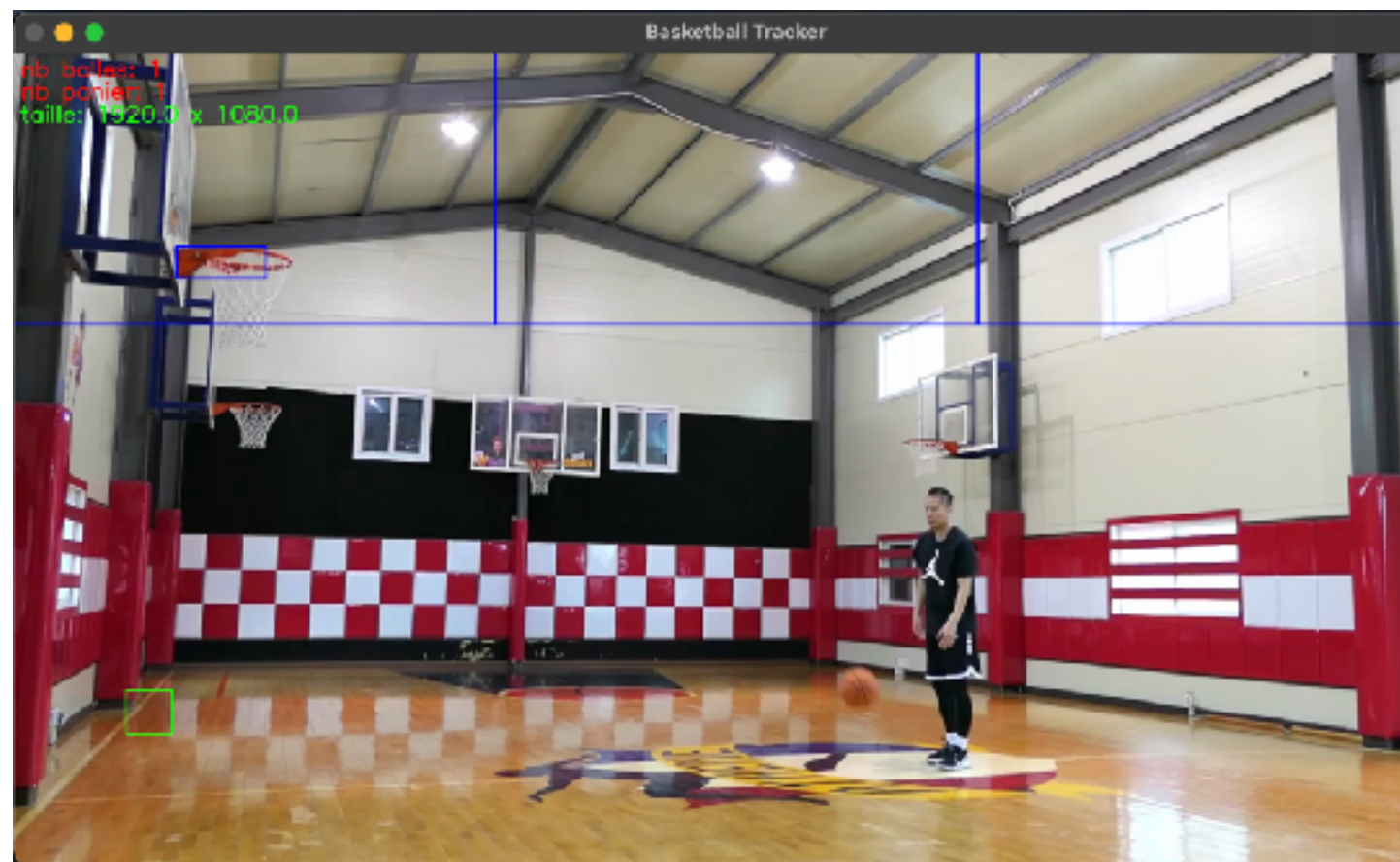
**RVB : Rouge Vert Bleu**



# Suivi de la balle

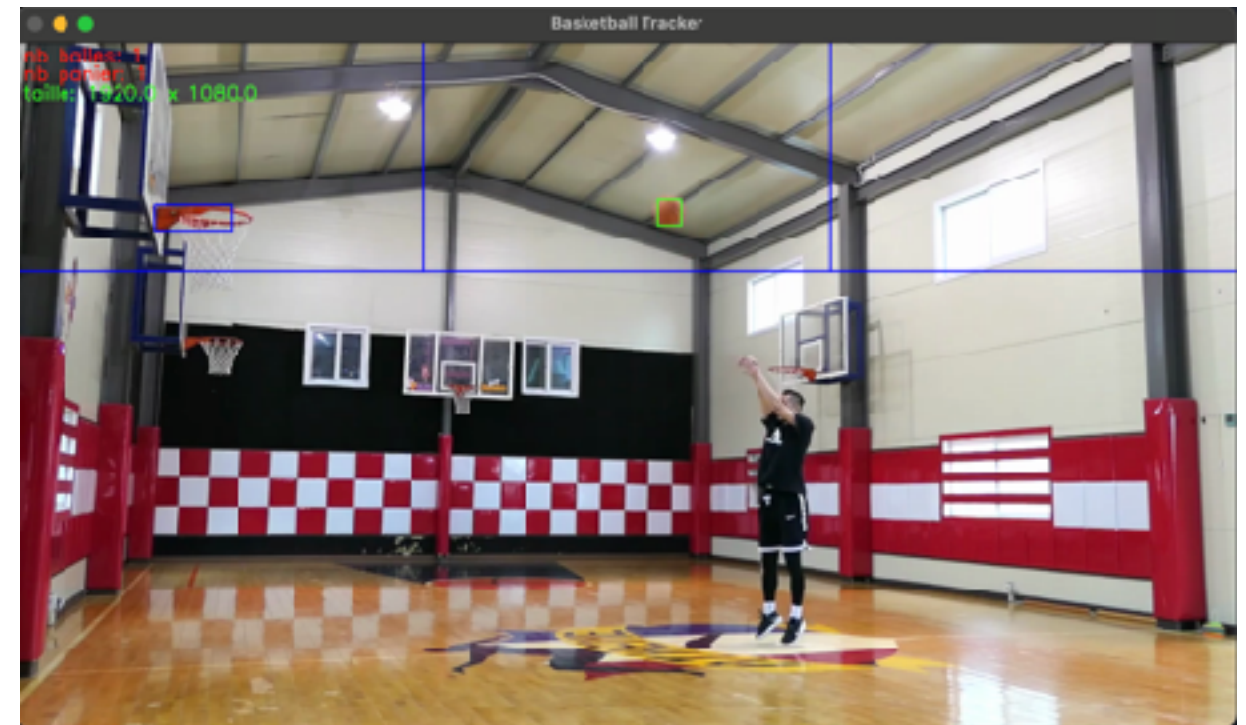
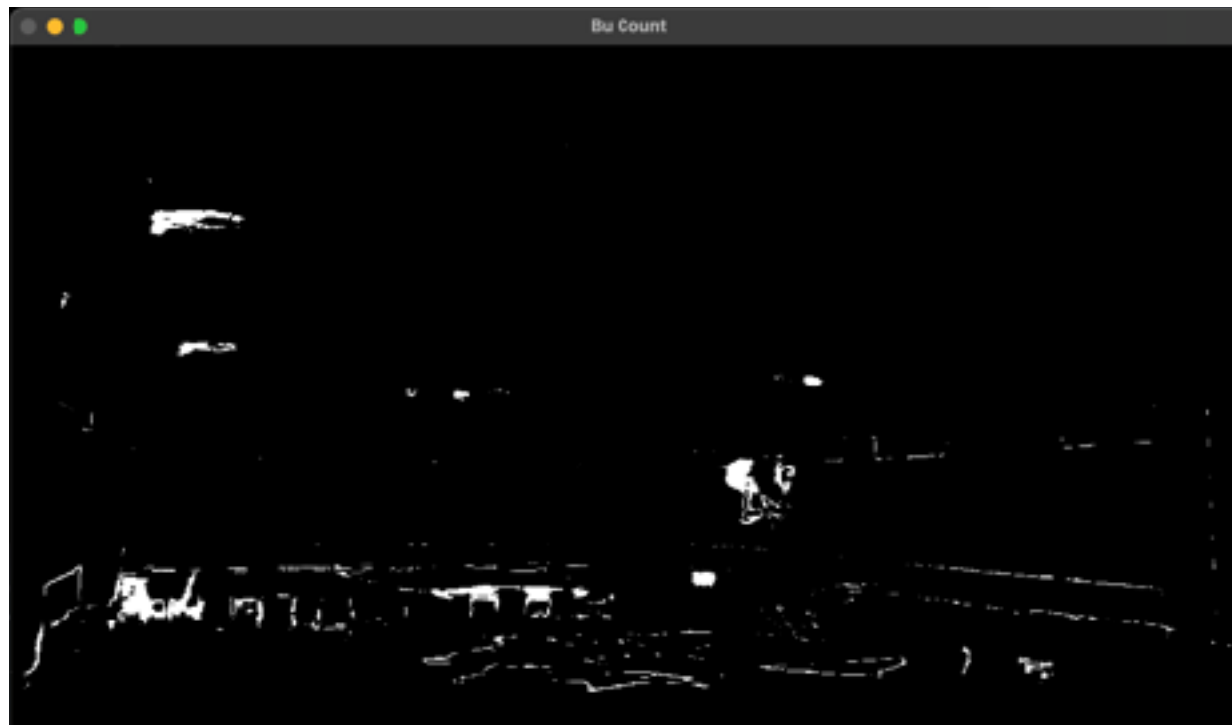


# Suivi de la balle





# Détection des paniers



# Algorithme

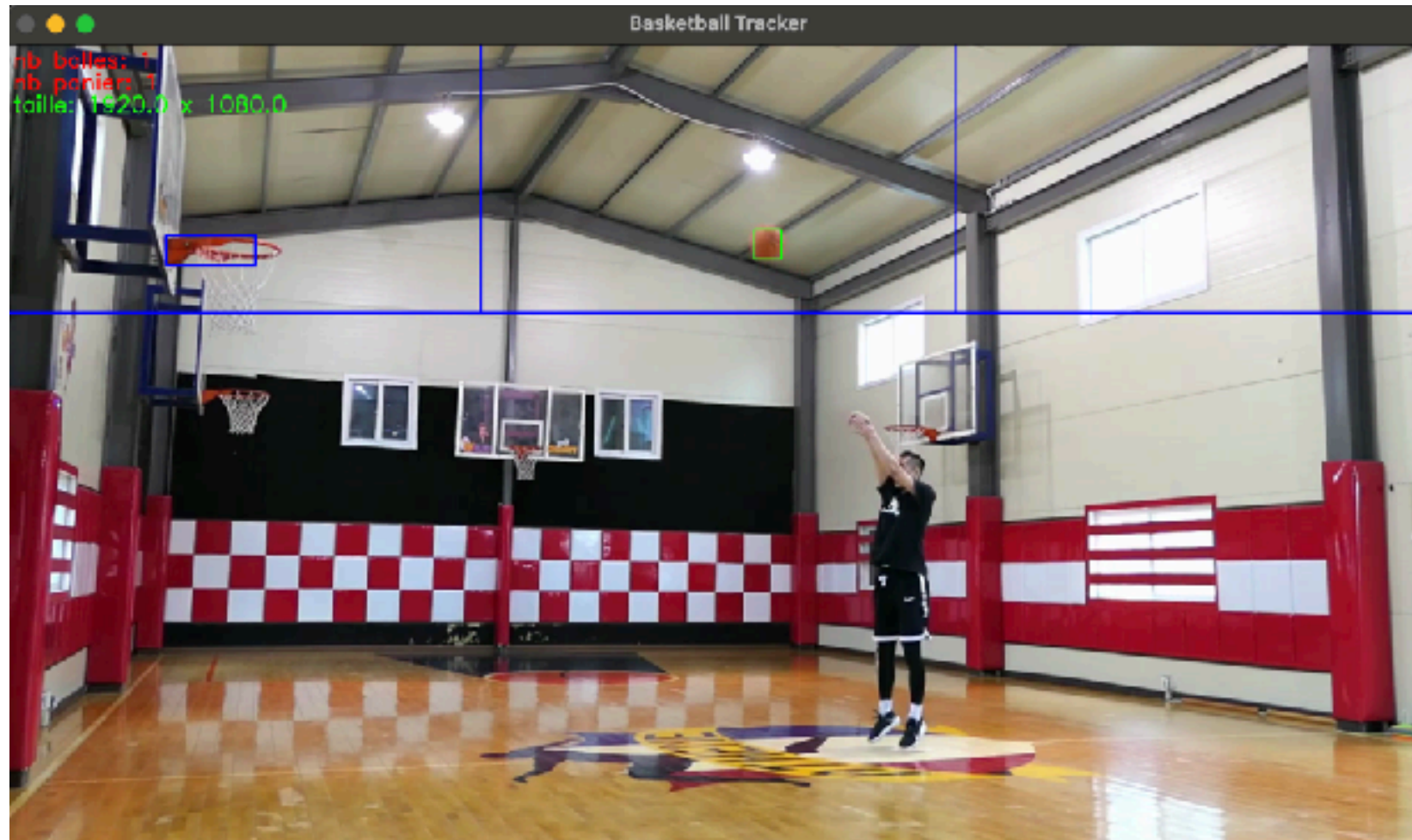
- ▶ Valeurs balle
- ▶ Valeurs panier
- ▶ Cap = récupération du flux vidéo
- ▶ Tant que la vidéo a encore des images
  - ▶ HSV = conversion de cap de RGB à HSV
  - ▶ On applique le filtre avec les valeurs pour les balles et le paniers
  - ▶ On vérifie la taille des balles
    - ▶ On dessine un carré autour
  - ▶ On vérifie la taille des paniers
    - ▶ On vérifie la position des paniers
    - ▶ On dessine un rectangle autour
  - ▶ On affiche le nombre de balles et de paniers
  - ▶ On affiche l'image avec les contours



# Algorithme

- ▶ Améliorations possibles :
  - ▶ Utilisation d'un filtre de Kalman
  - ▶ Éviter la confusion entre balles et panier avec la detection de mouvement
    - ▶ Soit avec la différence de position entre 2 images
    - ▶ Soit avec un filtre déjà implémenté
- ▶ Chose à finir :
  - ▶ Détection du panier marqué
  - ▶ Affiner les valeurs pour la balle et le panier

# Réponse à la problématique



# Annexes

```
1  #importation des librairies
2  import cv2
3  import numpy as np
4
5  #Définition des valeurs maximale et minimale prise par le filtre HSV pour la balle
6  lower_ball = np.array([5, 120, 70])
7  upper_ball = np.array([10, 255, 255])
8
9  #Définition des valeurs maximale et minimale prise par le filtre HSV pour les paniers
10 lower_bu = np.array([0, 50, 50])
11 upper_bu = np.array([0, 100, 100])
12
13 # Initialisation des variables permettant de savoir le nombre de balles détecté
14 prev_ball_count = 0
15 ball_count = 0
16
17 prev_bu_count = 0
18 bu_count = 0
19
20 # initialisation du compteur pour les points
21 point_count = 0
22
23 #Initialisation des variables servant à compartimenter le flux vidéo
24 width = 0
25 height = 0
26
27 mid_w = 0
28 mid_h = 0
29
30 bu_pos_1 = np.array([]) # initialisation des tableaux pour la position
31 bu_pos_2 = np.array([])
32
```

# Annexes

```
33 while True:
34     # Capture frame from the video
35     isclosed = 0
36     # Initialisation du flux vidéo, pour pouvoir le réutiliser facilement
37     cap = cv2.VideoCapture('/Users/hugo/Documents/Cours/Prepa/TIPE/TIPE_Basketball/script/IA_assistef/01.mp4')
38     #Récupération de la hauteur, de la largeur et de leurs moitiés
39     height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
40     width = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
41
42     mid_w = round((width/3))
43     mid_h = round((height/3))
44
45     print("hauteur : ", height, "largeur : ",width)
46     print("1/2 hauteur : ", mid_h, "1/2 largeur : ",mid_w)
47     while True:
48
49         ret, frame = cap.read()
50         rat , vide = cap.read()
51         if not ret or not rat:
52             isclosed
53             break
54
55         # Conversion des images de RGB à HSV
56         hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
57
58         # Création du masque pour isolé la balle et les paniers
59         ball_mask = cv2.inRange(hsv, lower_ball, upper_ball)
60         bu_mask = cv2.inRange(hsv, lower_bu, upper_ball)
61
62         # Apply morphological transformations to the mask
63         kernel = np.ones((5,5), np.uint8)
64         opening_ball = cv2.morphologyEx(ball_mask, cv2.MORPH_OPEN, kernel)
65         closing_ball = cv2.morphologyEx(opening_ball, cv2.MORPH_CLOSE, kernel)
66
67         opening_bu = cv2.morphologyEx(bu_mask, cv2.MORPH_OPEN, kernel)
68         closing_bu = cv2.morphologyEx(opening_bu, cv2.MORPH_CLOSE, kernel)
69         # Find contours of the basketball
70         contours_ball, hierarchy_ball = cv2.findContours(closing_ball, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
71         contours_bu , hierarchy_bu = cv2.findContours(closing_bu,cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
72         # Draw bounding boxes around the basketball and count them
73         ball_count = 0
74         bu_count = 0
75
76         c_line = (255,0,0)
77         tick_line = 2
78         cv2.line(img=frame,pt1=(0,mid_h), pt2=(1919,mid_h), color=c_line, thickness= tick_line)
79         cv2.line(img=frame, pt1=(mid_w,0), pt2=(mid_w,mid_h),color=c_line, thickness= tick_line)
80         cv2.line(img=frame, pt1=(2*mid_w,0), pt2=(2*mid_w,mid_h),color=c_line, thickness= tick_line)
```

# Annexes

```
81     #Boucle du traitement du resultat du filtre pour la balle, affichage en vert
82     for element in contours_ball:
83         area = cv2.contourArea(element)
84         x, y, w, h = cv2.boundingRect(element)
85         dif = abs(w-h)
86         if area > 900 and dif < 10: #permet de s'assurer que les petites taches ne sont pas prises en compte et que le contour est proche d'un carré
87             x, y, w, h = cv2.boundingRect(element)
88             cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)
89             ball_count += 1
90
91     #Boucle du traitement du resultat du filtre pour les paniers , affichage en bleur
92     for countour in contours_bu :
93         area_bu = cv2.contourArea(countour)
94         xb,yb,wb,hb = cv2.boundingRect(countour)
95         if area_bu > 1500 and yb < mid_h and xb < mid_w :
96             cv2.rectangle(frame, (xb,yb),(xb+wb,yb+hb), (255,0,0),2)
97             bu_count += 1
98         if area_bu > 1500 and yb < mid_h and 2*mid_w<xb < width :
99             cv2.rectangle(frame, (xb,yb),(xb+wb,yb+hb), (255,0,0),2)
100             bu_count += 1
101
102     # Display the tracking result on the screen
103     cv2.putText(frame, "nb balles: " + str(ball_count), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
104     cv2.putText(frame, "nb panier: " + str(bu_count), (10,60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255),2)
105
106     cv2.putText(frame, "taille: "+str(width)+" x "+str(height), (10, 90), cv2.FONT_HERSHEY_SIMPLEX,1, (0,255,0),2)
107
108     cv2.imshow("Ball Count", ball_mask)
109     cv2.imshow("Bu Count", bu_mask)
110     cv2.imshow("Basketball Tracker", frame)
111     cv2.imshow("sans rien",vide)
112     # Exit the program if the 'q' key is pressed
113     if cv2.waitKey(1) & 0xFF == ord('q'):
114         isclosed= 1
115         break
116
117     # Update the previous count
118     prev_ball_count = ball_count
119     prev_bu_count = bu_count
120 if isclosed :
121     break
122
123 # Release the video capture and close all windows
124 cap.release()
125 cv2.destroyAllWindows()
```