

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



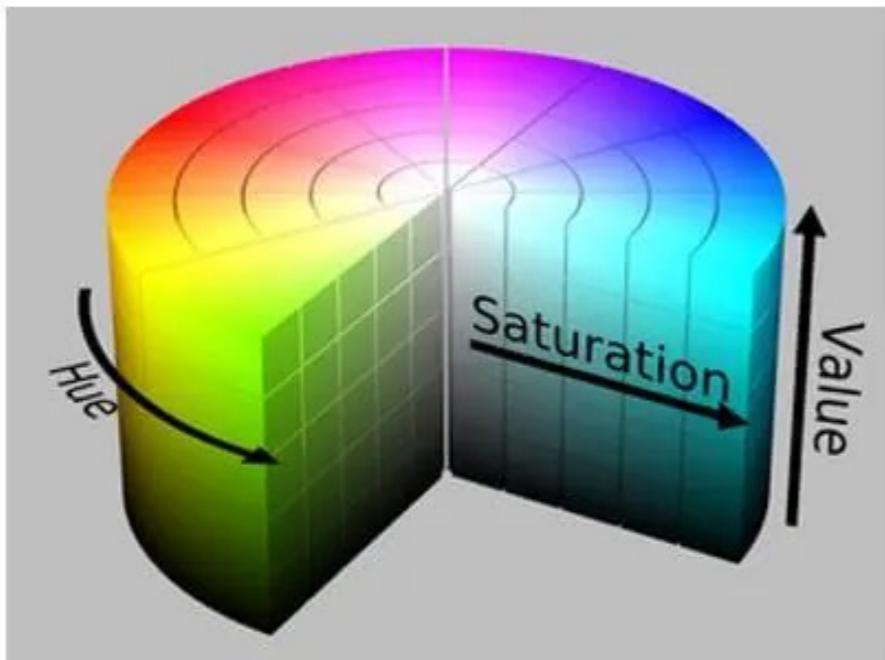
bebasket.fr



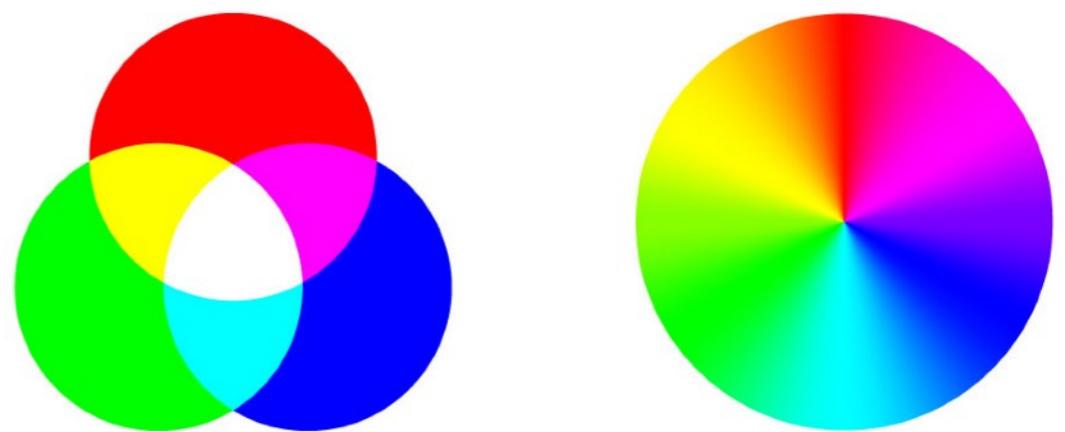
bodet-sport.com

Suivi de la balle

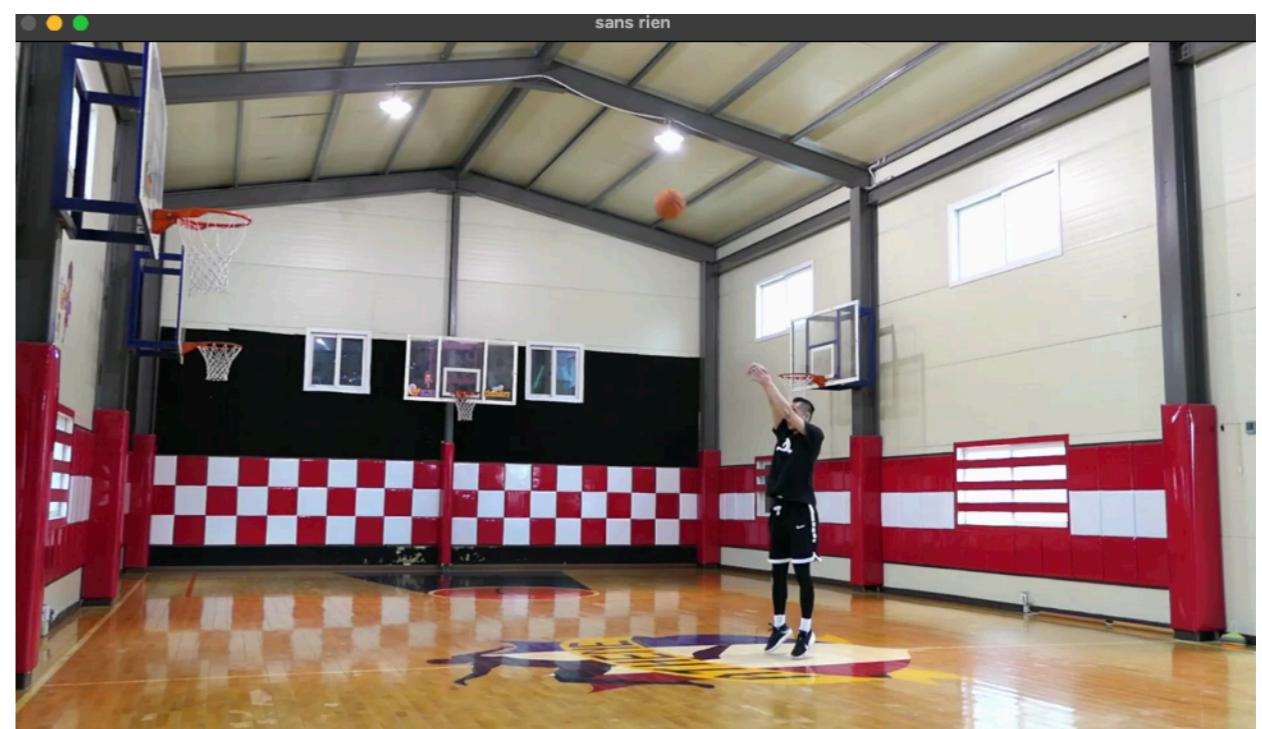
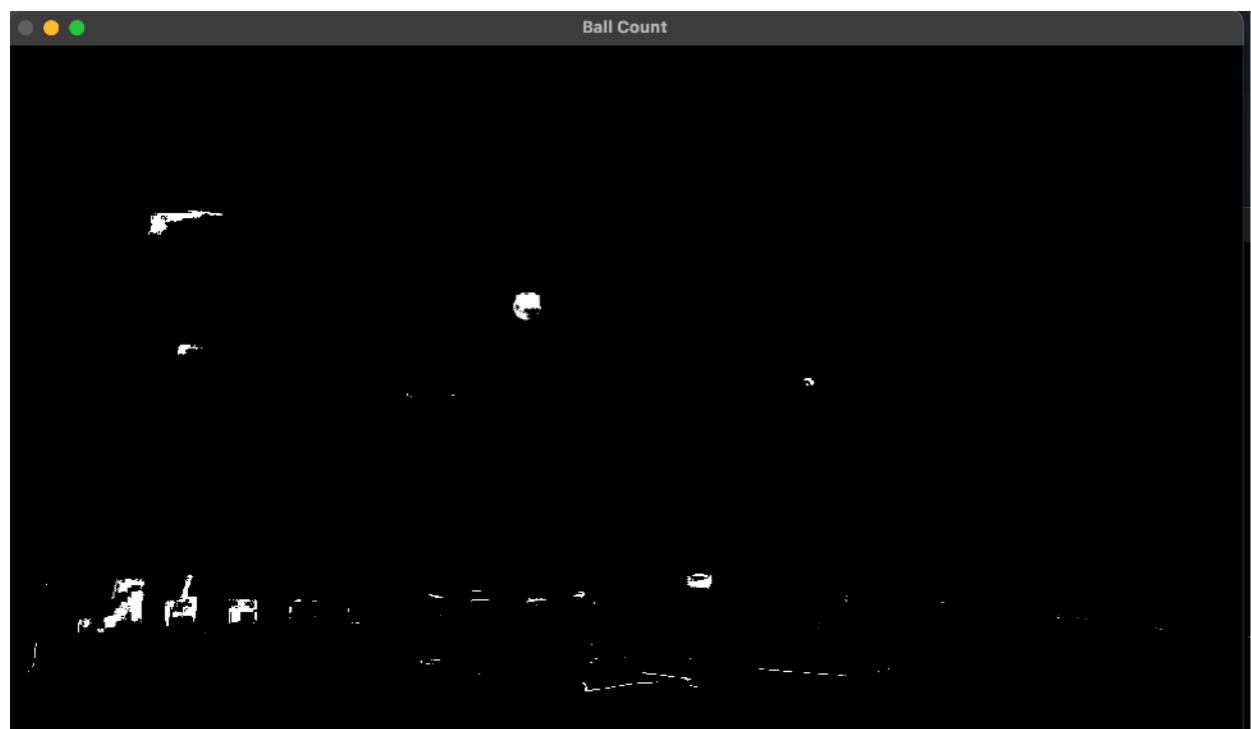
HSV : Hue Saturation Value
(Teinte saturation valeur)



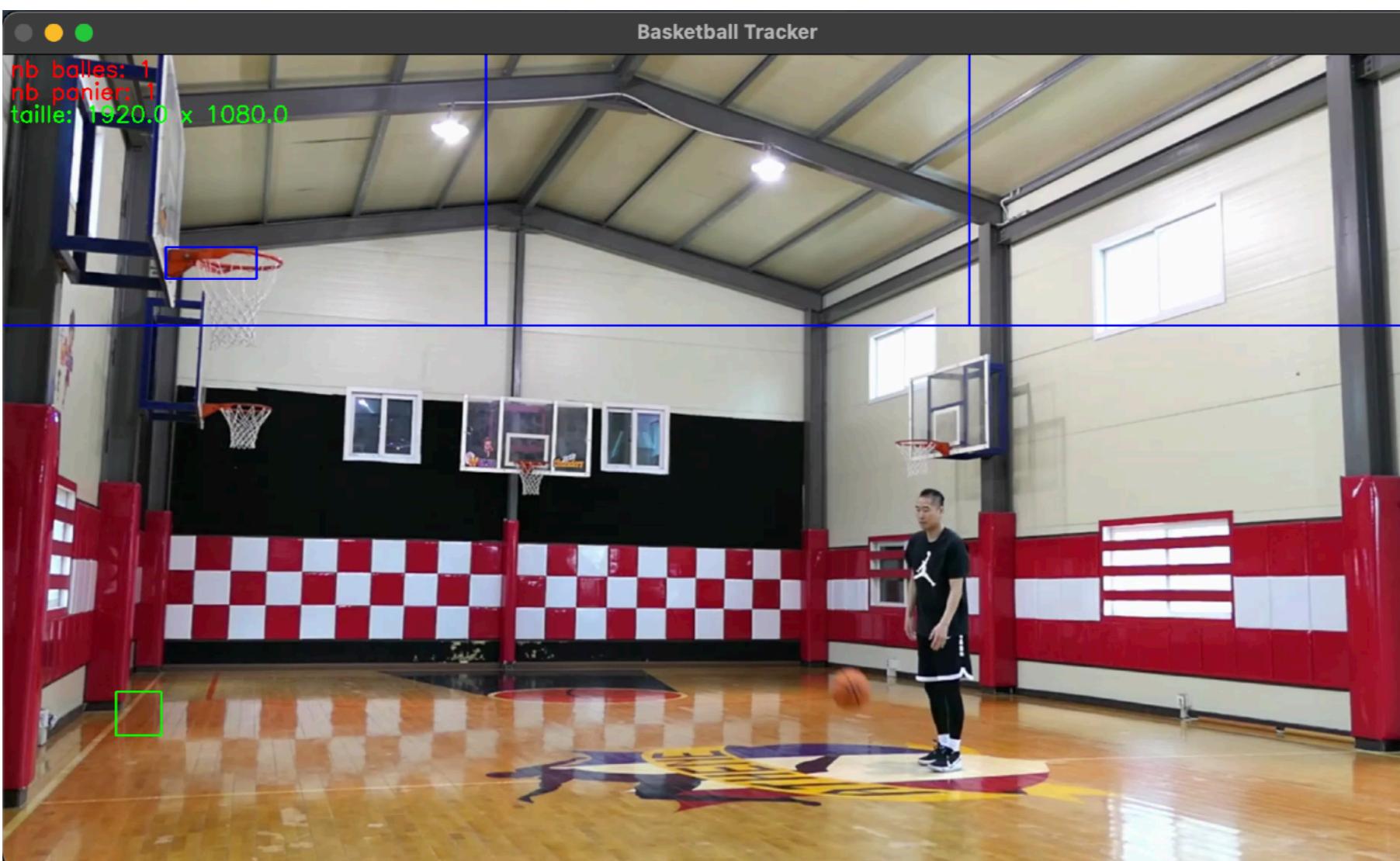
RVB : Rouge Vert Bleu



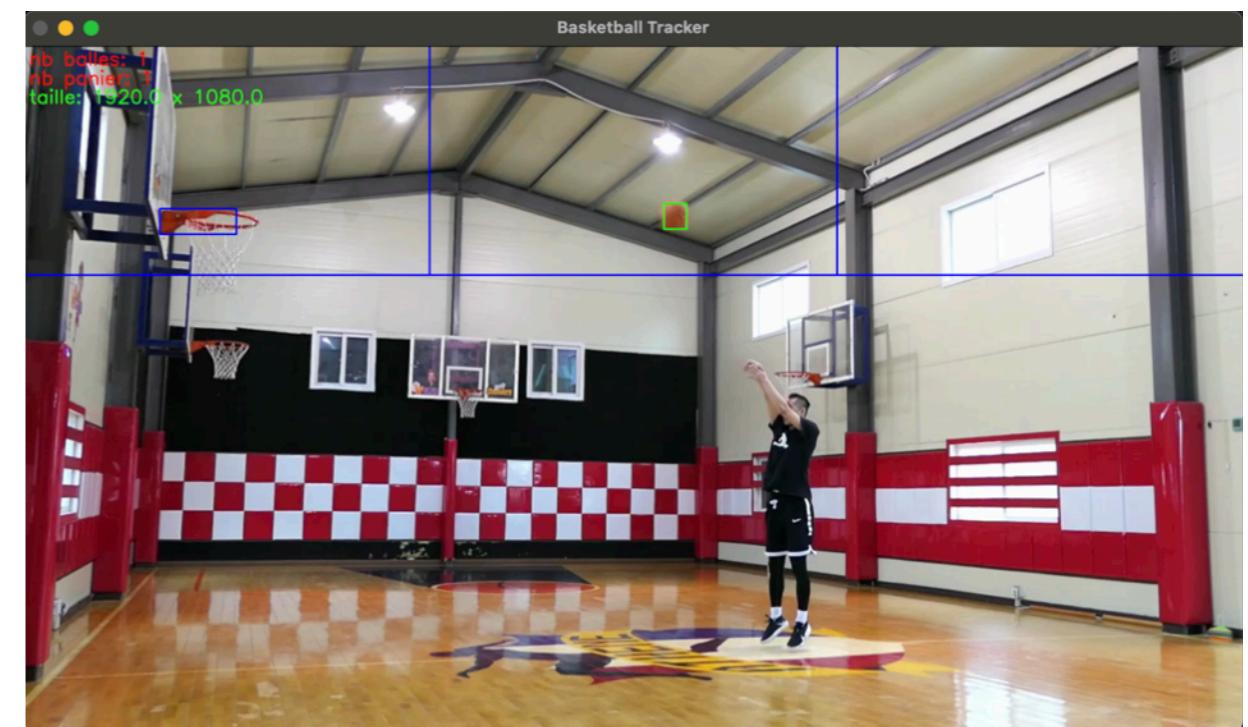
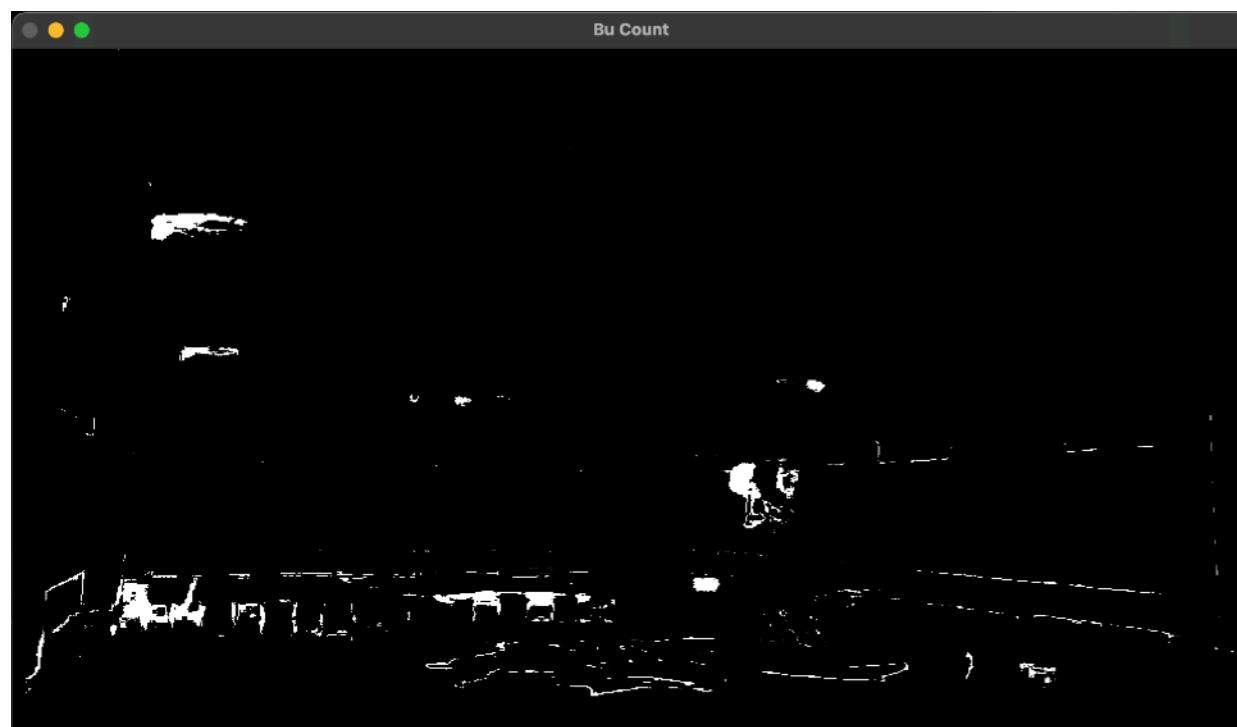
Suivi de la balle



Suivi de la balle



Détection des paniers



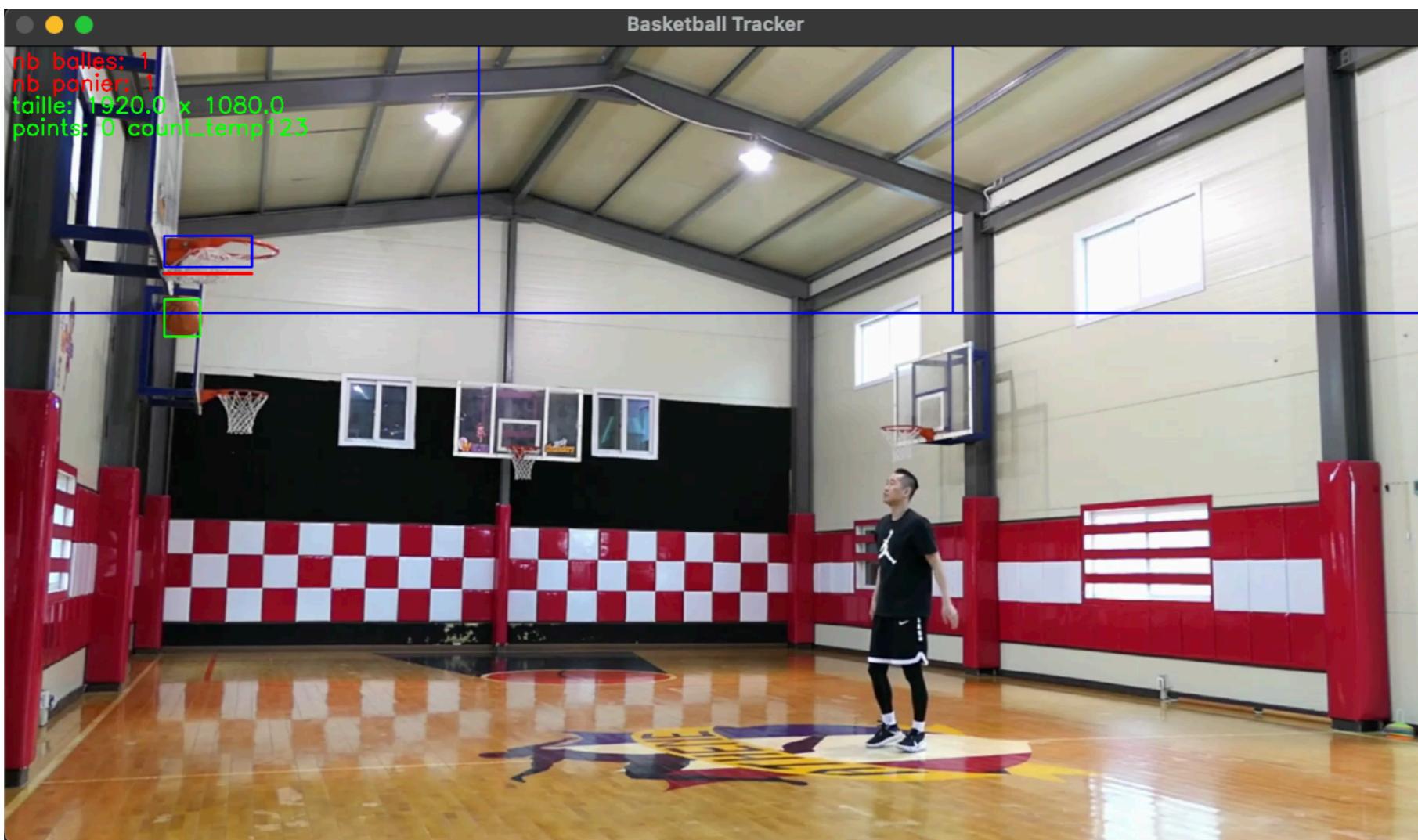
Algorithme

- ▶ Valeurs de teinte pour la balle
- ▶ Valeurs de teinte pour le 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
 - ▶ Si la balle est dans la zone rouge sous le panier
 - ▶ On ajoute 2 points au score
 - ▶ 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
 - ▶ Détection de la balle grâce un filtre de mouvement
 - ▶ Utiliser le centre de la balle
- ▶ Chose encore à implémenter :
 - ▶ Affiner les valeurs pour la balle et le panier
 - ▶ Déetecter quelle équipe marque

Réponse à la problématique



Anhexe

```
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
33 count_temp = 0
34
35 def scored(x,y,x2,y2,x3,y3):
36     if x <x3 and x3 <x2 and y< y2 and y2< y3 :
37         return True
38     else :
39         return False
40
41
42
43 while True:
44     # Capture frame from the video
45     isclosed = 0
46     point_count = 0
47
48     # Initialisation du flux vidéo, pour pouvoir le réutiliser facilement
49     cap = cv2.VideoCapture('/Users/hugo/Documents/Cours/Prepa/TIPE/TIPE_Basketball/script/IA_assistef/01.mp4')
50     #Récupération de la hauteur, de la largeur et de leurs moitiés
```

Annexe

```
50     #Récuperation de la hauteur, de la largeur et de leurs moitiés
51     height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
52     width = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
53
54     mid_w = round((width/3))
55     mid_h = round((height/3))
56
57     kernel = np.ones((5,5), np.uint8)
58
59
60     fst_ret,fst_frame = cap.read()
61
62     fst_hsv = cv2.cvtColor(fst_frame, cv2.COLOR_BGR2HSV)
63
64     bu_mask = cv2.inRange(fst_hsv, lower_bu, upper_ball)
65
66     opening_bu = cv2.morphologyEx(bu_mask, cv2.MORPH_OPEN, kernel)
67     closing_bu = cv2.morphologyEx(opening_bu, cv2.MORPH_CLOSE, kernel)
68
69     contours_bu , hierarchy_bu = cv2.findContours(closing_bu, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
70
71     print("hauteur : ", height, "largeur : ",width)
72     print("1/2 hauteur : ", mid_h, "1/2 largeur : ",mid_w)
73     while True:
74
75         ret, frame = cap.read()
76         rat , vide = cap.read()
77         if not ret or not rat:
78             isclosed
79             break
80
81         # Conversion des images de RGB à HSV
82         hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
83
84         # Création du masque pour isolé la balle et les paniers
85         ball_mask = cv2.inRange(hsv, lower_ball, upper_ball)
86
87         # Apply morphological transformations to the mask
88         opening_ball = cv2.morphologyEx(ball_mask, cv2.MORPH_OPEN, kernel)
89         closing_ball = cv2.morphologyEx(opening_ball, cv2.MORPH_CLOSE, kernel)
90
91         # Find contours of the basketball
92         contours_ball, hierarchy_ball = cv2.findContours(closing_ball, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
93         contours_bu , hierarchy_bu = cv2.findContours(closing_bu, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
94         # Draw bounding boxes around the basketball and count them
95         ball_count = 0
96         bu_count = 0
97
98         c_line = (255,0,0)
99         tick_line = 2
100        cv2.line(img=frame,pt1=(0,mid_h), pt2=(1919,mid_h), color=c_line, thickness=tick_line)
```

Annexe

```
100     cv2.line(img=frame, pt1=(0,mid_h), pt2=(1919,mid_h), color=c_line, thickness= tick_line)
101     cv2.line(img=frame, pt1=(mid_w,0), pt2=(mid_w,mid_h),color=c_line, thickness= tick_line)
102     cv2.line(img=frame, pt1=(2*mid_w,0), pt2=(2*mid_w,mid_h),color=c_line, thickness= tick_line)
103     #Boucle du traitement du resultat du filtre pour les paniers , affichage en bleur
104     for countour in contours_bu :
105         area_bu = cv2.contourArea(countour)
106         xb,yb,wb,hb = cv2.boundingRect(countour)
107         if area_bu > 1500 and yb < mid_h and xb < mid_w :
108             cv2.rectangle(frame, (xb,yb),(xb+wb,yb+hb), (255,0,0),2)
109             y_area = yb + 50
110             x_area = xb
111             y_area_2 = y_area +2
112             x_area_2 = xb + wb
113             cv2.rectangle(frame, (x_area,y_area),(x_area_2,y_area_2),(0,0,255), 2 )
114             bu_count += 1
115         if area_bu > 1500 and yb < mid_h and 2*mid_w<xb < width :
116             cv2.rectangle(frame, (xb,yb),(xb+wb,yb+hb), (255,0,0),2)
117             bu_count += 1
118
119         if bu_count > 2 :
120             print("Ya un probleme qlq part : on detecte 2 paniers")
121     #Boucle du traitement du resultat du filtre pour la balle, affichage en vert
122     for element in contours_ball:
123         area = cv2.contourArea(element)
124         x, y, w, h = cv2.boundingRect(element)
125         dif = abs(w -h)
126         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
127             x, y, w, h = cv2.boundingRect(element)
128             cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)
129             ball_count += 1
130         if scored(x_area,y_area,x_area_2,y_area_2,x,y) and count_temp > 10 :
131             point_count += 2
132             count_temp = 0
133         count_temp+=1
134
135         print("nb de points marqué : ", point_count)
136     if ball_count > 1 :
137         print("on a un problème quelque part, on detecte 2 balles")
138
139     # Display the tracking result on the screen
140     cv2.putText(frame, "nb balles: " + str(ball_count), (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
141     cv2.putText(frame, "nb panier: " + str(bu_count), (10,60), cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255),2)
142
143     cv2.putText(frame, "taille: "+str(width)+" x "+str(height), (10, 90), cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0),2)
144     cv2.putText(frame, "points: "+str(point_count) + " count_temp"+str(count_temp), (10, 120), cv2.FONT_HERSHEY_SIMPLEX,1,(0,255,0),2)
145
146     cv2.imshow("Ball Count", ball_mask)
147     cv2.imshow("Bu Count", bu_mask)
148     cv2.imshow("Basketball Tracker", frame)
149
150     #cv2.imshow("sans rien",vide)
```

Ahnexe

```
150     #cv2.imshow("sans rien",vide)
151     # Exit the program if the 'q' key is pressed
152     if cv2.waitKey(1) & 0xFF == ord('q'):
153         isclosed= 1
154         break
155
156     # Update the previous count
157     prev_ball_count = ball_count
158     prev_bu_count = bu_count
159     if isclosed :
160         break
161
162     # Release the video capture and close all windows
163     cap.release()
164     cv2.destroyAllWindows()
```