

BOUHELIER

Marius

Rapport Perception

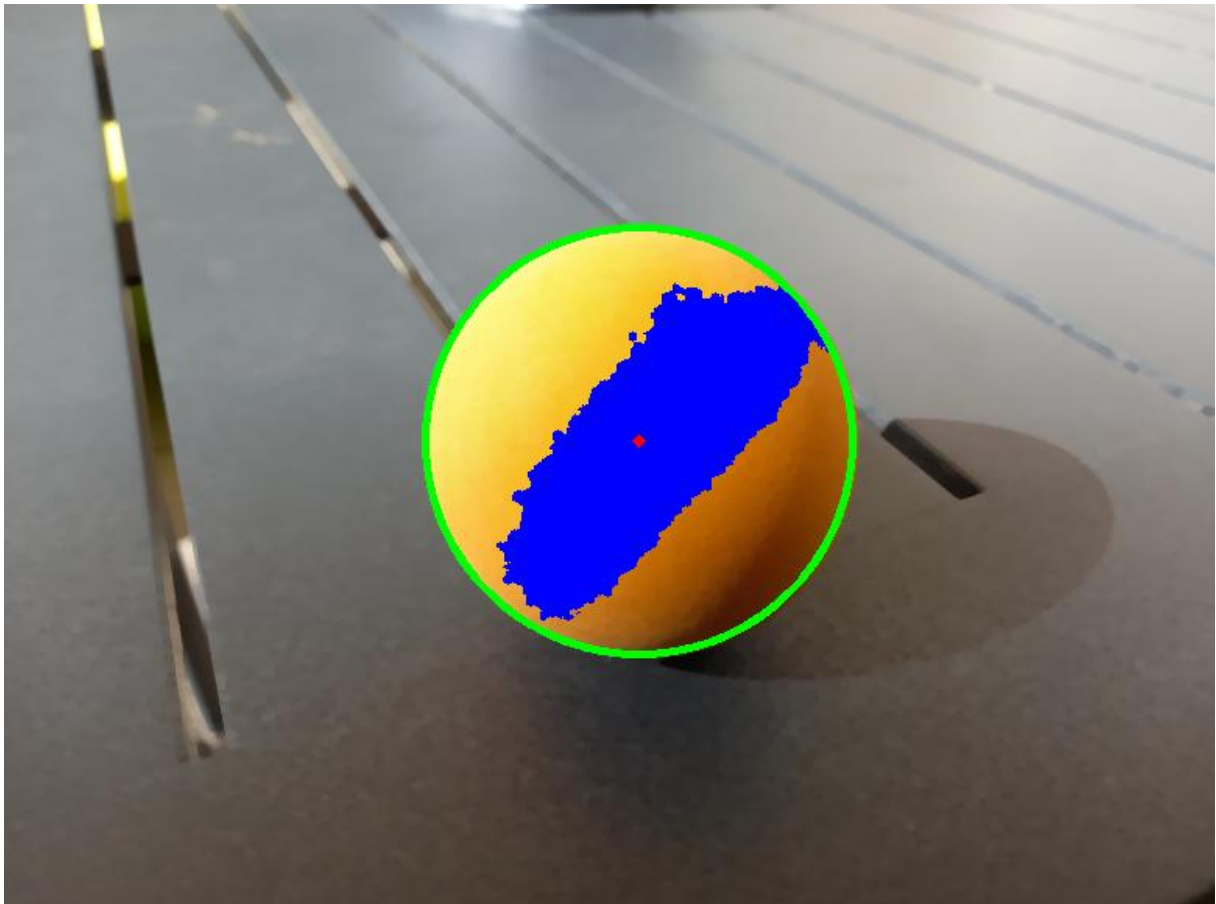
1/Introduction

L'objectif de ce projet est de repérer un objet, en l'occurrence une balle, sur une image dans un premier temps, puis sur une vidéo. Pour ce faire, nous utiliserons le langage Python couplé à la bibliothèque OpenCV. Nous allons expérimenter plusieurs méthodes, comme le repérage par la couleur (RGB), la teinte HSV et grâce à la transformée de Hough.

2/Méthode RGB

Dans un premier temps, nous allons essayer de repérer la balle dans l'image grâce à la valeur RGB de chaque pixel de cette dernière. L'objectif est de pouvoir cliquer sur la balle dans l'image et de pouvoir colorier entièrement cette dernière pour l'identifier.

Concrètement, le code réalisé prend la valeur RGB du pixel sur lequel nous venons de cliquer, calcule la distance euclidienne entre ce pixel et tous les autres pixels de l'image puis colorie grâce à un masque tous les pixels dont la valeur est proche de celle du pixel cliqué. La proximité de la valeur est arbitrairement choisie.

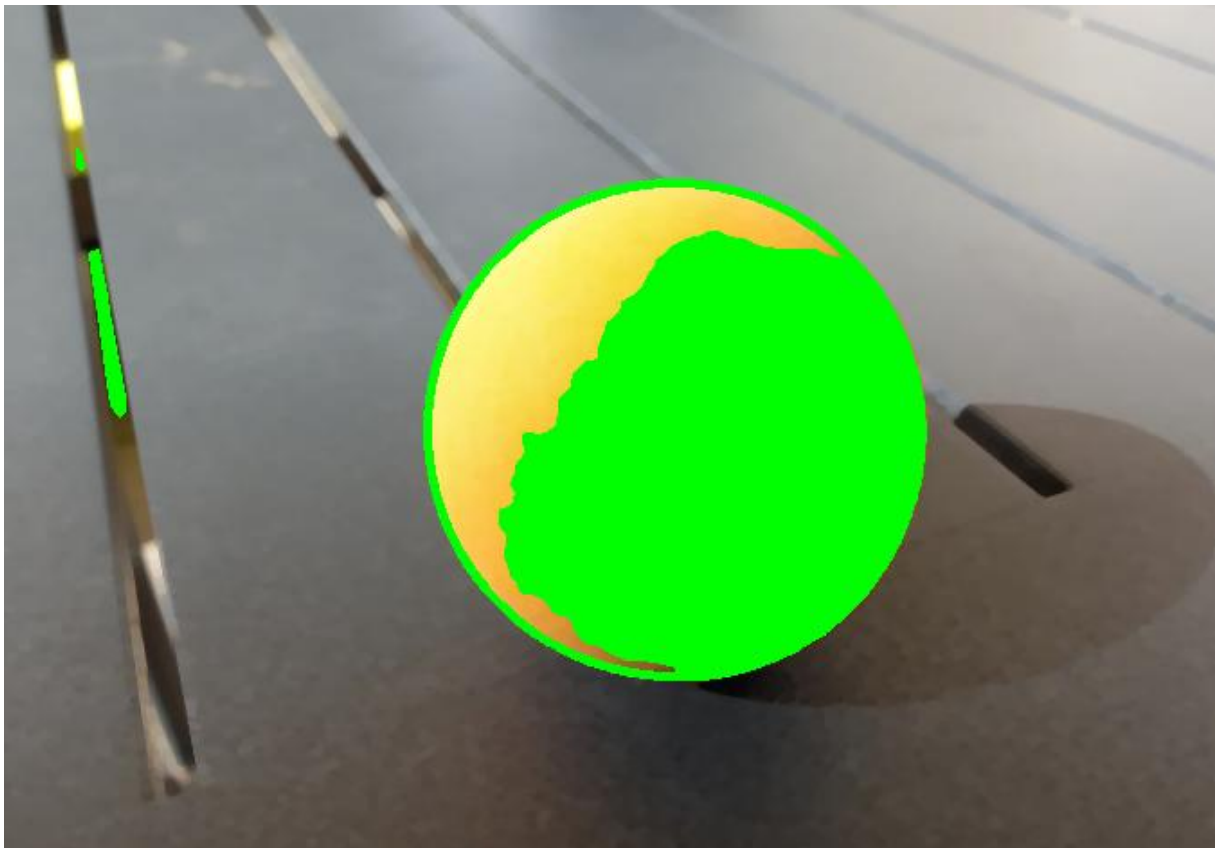


Après plusieurs tests sur différentes images, la conclusion est que cette méthode est loin d'être optimale pour différentes raisons. Premièrement, il n'y a pas une valeur de proximité « parfaite », soit elle est trop basse et n'est reconnue qu'une petite partie de la balle soit elle est trop élevée et la balle est relativement bien repérée mais d'autres éléments de l'image sont aussi coloriés. De plus sur une image comprenant des parties de la même couleur que la balle que nous voulons repérer la méthode ne fait aucunement la différence. Et finalement suivant l'exposition de la balle certains pixels à l'ombre auront une valeur complètement différente de pixels surexposés par exemple.

3/Méthode HSV

Cette méthode fonctionne de la même manière que la première (comparaisons de la valeur d'un pixel de la balle à tous les autres pixels de

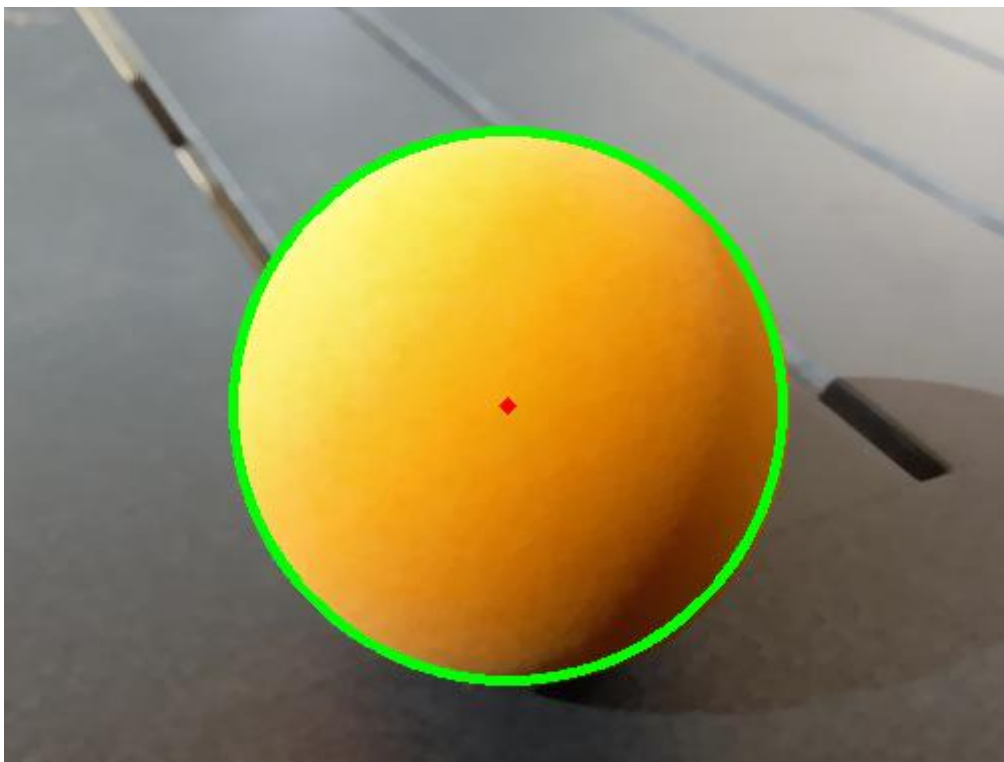
l'image) cependant nous utilisons 3 autres paramètres pour définir un pixel, Hue, Saturation, Value (HSV). Ici, pour calculer la proximité entre deux pixels on utilise aussi la distance euclidienne seulement, la valeur H étant un angle il faut rendre sa plage de valeur circulaire (un angle de 360° et un angle de 1° sont en fait très proche). Ainsi fait, on réalise les mêmes opérations que la méthode RGB. On obtient :



Pour conclure s'agissant de la méthode HSV, nous remarquons que la balles est en plus grande partie colorié par rapport à la méthode RGB mais cela reste imparfait pour les mêmes raisons évoquées précédemment hormis que la teinte (paramètre H) reste stable face à un changement de lumière.

4/ Transformée de Hough

La méthode de Hough pour la détection des cercles est une technique que nous utilisons pour identifier des formes circulaires dans une image en nous basant sur les contours et les gradients d'intensité. Avant d'appliquer cette méthode, nous effectuons un pré-traitement de l'image en lui appliquant un flou gaussien afin de réduire le bruit et de lisser les bords, ce qui rend la détection plus précise. Nous utilisons ensuite la fonction `cv.HoughCircles`, qui transforme les points de l'image dans un espace de paramètres où chaque point représente un cercle possible défini par son centre et son rayon ; les cercles sont alors détectés là où plusieurs votes se regroupent dans cet espace. Nous appliquons également des opérations morphologiques d'ouverture et de fermeture avant la détection : la fermeture, qui correspond à une dilatation suivie d'une érosion, nous permet de combler les petits trous et d'unifier les régions disjointes, tandis que l'ouverture, une érosion suivie d'une dilatation, sert à éliminer les petites taches et imperfections dues au bruit. En combinant ces deux traitements, nous nettoyons et renforçons les contours des objets, ce qui facilite une détection plus fiable et plus précise des cercles par la méthode de Hough.



5/ Vidéo

Dans ce programme, nous effectuons une détection de cercles en temps réel à partir d'un flux vidéo. À chaque image capturée par la caméra, nous la convertissons en niveaux de gris, puis nous lui appliquons un flou gaussien pour réduire le bruit. Nous utilisons ensuite des opérations morphologiques d'ouverture et de fermeture afin de nettoyer l'image et de renforcer les contours. Une fois cette préparation terminée, nous appliquons la méthode de Hough pour repérer les formes circulaires présentes dans la scène. Enfin, nous traçons ces cercles directement sur la vidéo, ce qui nous permet d'observer en direct la détection des objets ronds, même dans des conditions visuelles imparfaites.



6/ Conclusion

En conclusion, nous avons expérimenté plusieurs méthodes de détection afin de repérer une balle dans une image puis dans une vidéo. Les approches basées sur la couleur, bien qu'intuitives, se sont révélées sensibles aux variations de luminosité et aux couleurs proches dans le

fond. En revanche, la transformée de Hough, associée à un pré-traitement par ouverture et fermeture morphologiques, s'est montrée bien plus robuste et précise pour détecter les formes circulaires. L'application de cette méthode sur un flux vidéo en temps réel a permis d'obtenir un repérage efficace et stable, validant ainsi son intérêt pour la détection automatique d'objets de forme simple.