Léo Vampouille

**Rapport de Perception**

**Introduction**

Dans le cadre des séances de TD de Perception, l’objectif était de développer un programme informatique capable de détecter automatiquement une balle sur une image puis d’en afficher la position. Pour ce faire, nous avons utilisé le langage Python ainsi que la bibliothèque OpenCV, spécialisée dans l’acquisition, la manipulation et le traitement d’images et de vidéos.

Notre démarche a été itérative :

1. Segmentation par couleur en RGB,
2. Puis en HSV,
3. Amélioration par morphologie mathématique,
4. Enfin détection géométrique par transformée de Hough pour les cercles.

Cette progression nous a permis d’augmenter progressivement la robustesse et la précision de la détection.

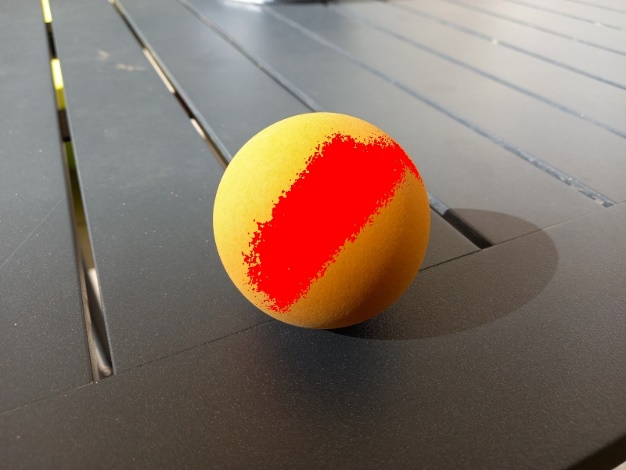
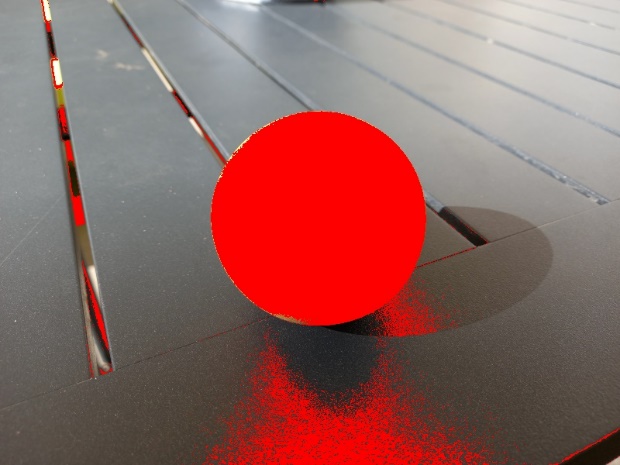
**Approche par segmentation RGB**

Nous avons d'abord travaillé directement dans l’espace RGB, en sélectionnant les pixels proches d’une couleur cliquée dans l’image :

Le script 1\_mask\_RGB permettait de cliquer sur la balle et de recolorier les pixels ayant une couleur similaire.

Limites observées

* La couleur de la balle n’est pas uniforme → plusieurs clics nécessaires
* Forte sensibilité aux variations d’éclairage → segmentation incomplète

Conséquence : la détection est peu fiable.

*Balle coloriée selon RGB après 1 clic Balle coloriée selon RGB après plusieurs clic*

Conclusion : l’espace RGB est trop dépendant des variations de lumière

**Passage en espace HSV**

Pour contourner ces limites, nous avons transformé l’image dans l’espace HSV (Teinte–Saturation–Valeur).  
Seule la teinte H a été utilisée pour segmenter la balle.

Dans le code :

* Conversion en HSV
* Récupération de la teinte au clic
* Création d’un masque binaire selon une plage de teinte

Extrait du script 3\_mask\_H\_et\_morpho.py :

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Résultat

* La plupart de la balle est détectée en un seul clic
* **Mais la teinte est présente ailleurs → bruit dans l’image

*Balle coloriée selon H après 1 clic*

Conclusion : mieux, mais nécessite un filtrage spatial.

**Amélioration par opérations morphologiques**

Nous utilisons une ouverture :

* Érosion → suppression des petites zones isolées
* Dilatation → restauration de la forme principale

Code utilisé dans le masque HSV :

Une image contenant texte, capture d’écran, Police, ligne

Le contenu généré par l’IA peut être incorrect.

Résultat : formes plus propres et contours lissés ; mais possible détection d’autres objets ayant une teinte proche.

Une image contenant monochrome, noir et blanc

Le contenu généré par l’IA peut être incorrect.

*Une image contenant vert, dessin humoristique, Dessin d’enfant, Graphique

Le contenu généré par l’IA peut être incorrect.Une image contenant capture d’écran, noir et blanc

Le contenu généré par l’IA peut être incorrect.Masque selon H*

*Masque selon H après ouverture balle coloriée grâce au masque*

**Détection géométrique : Transformée de Hough**

Nous avons ensuite utilisé une approche géométrique.  
Objectif : détecter la balle par sa forme circulaire, indépendamment de la couleur.

Une image contenant texte, capture d’écran, Police, logiciel

Le contenu généré par l’IA peut être incorrect.Extrait du fichier 4\_mask\_H\_morpho\_et\_cercles.py :

Avec :

Dp 🡪 facteur de réduction de résolution

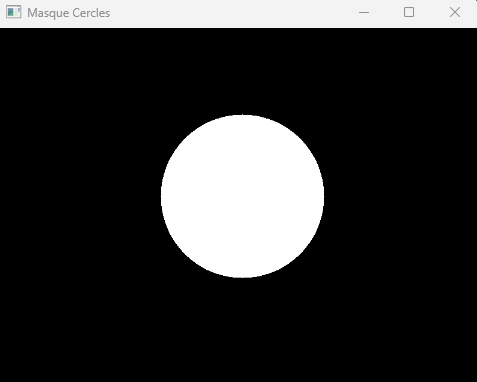
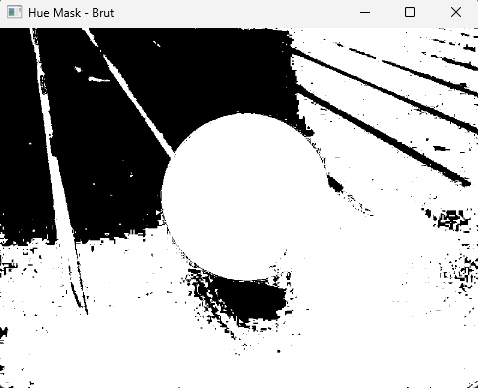
minDist 🡪 écarte les fausses détections multiples

param1 🡪 seuil du détecteur de Canny (contours)

param2 🡪 seuil de validation des cercles (sensibilité)

min/maxRadius🡪 contraintes géométriques → évite les erreurs

Retourne le centre (cx,cy)du cercle et son rayon r.

 *Masque cercles Masque H*

**Fusion couleur + forme : solution finale**

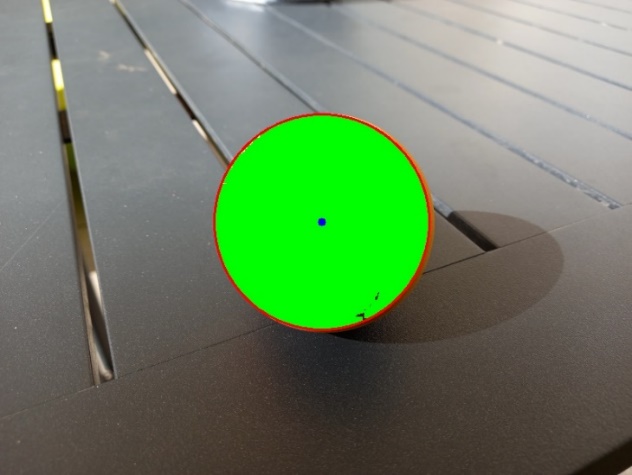
Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.Nous combinons les deux masques :

Ce filtrage logique garde uniquement la zone :  
- ayant la bonne teinte  
- appartenant à un cercle détecté

Résultat final :

* Balle détectée avec une précision élevée
* Centre marqué par un point bleu
* Bordure circulaire tracée en rouge

La balle est parfaitement localisée, segmentée et centrée.

*Balle coloriée sur le masque combiné des 2 précedents*

**Extension à la vidéo : détection en temps réel image par image**

Après avoir validé la détection de balle sur image fixe, nous avons prolongé notre approche sur une séquence vidéo contenant une balle en mouvement. L’objectif a été de détecter la balle à chaque frame, afin d’obtenir un suivi visuel fluide.

Pour cela, nous réutilisons les deux étapes précédemment validées :

1. Masque de couleur (segmenter les pixels proches de la teinte de la balle dans l’espace HSV)
2. Détection de forme circulaire (Transformée de Hough)

Une image contenant texte, Police, capture d’écran, nombre

Le contenu généré par l’IA peut être incorrect.Ces opérations sont appliquées à chaque image de la vidéo grâce à la boucle de lecture image-par-image :

a) Segmentation par couleur dynamique

Nous convertissons chaque frame en HSV, puis nous appliquons un filtrage sur la teinte.

Une ouverture morphologique est ensuite appliquée pour supprimer le bruit

b) Détection du cercle dans chaque frame

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.Comme précédemment, la balle est identifiée par HoughCircles dans l’image convertie en niveaux de gris et légèrement floutée :

Les cercles validés fournissent le centre et le rayon du cercle et ceux-ci sont affichés en temps réel

c) Fusion et affichage du résultat

Une image contenant texte, Police, capture d’écran

Le contenu généré par l’IA peut être incorrect.La fusion logique entre le masque couleur et le masque cercle garantit que seul le bon objet est suivi :

Résultats et analyse :

La balle est détectée dans une grande partie de la séquence, mais la qualité de suivi peut varier selon les conditions visuelles de la scène.

Observations principales :

Bonne luminosité, balle bien visible🡪 Détection correcte avec affichage du cercle

Balle floue en mouvement🡪 Perte de détection temporaire

Arrière-plan comportant des couleurs proches🡪 Pixels parasites visibles, faux positifs possibles

**Conclusion**

Nous avons construit progressivement une méthode robuste de détection :

|  |  |  |
| --- | --- | --- |
| Méthode | Avantages | Limites |
| RGB | Simple | Très sensible à la lumière |
| HSV | Bonne segmentation couleur | Pixel parasites |
| Morphologie | Contours plus propres | Encore du bruit |
| HoughCircles | Détection du cercle fiable | Peut détecter de mauvais cercles |
| Fusion | Résultat robuste et précis | Dépend des paramètres |