



## ÉCOLE CENTRALE LYON

UE PRO  
RAPPORT DE FIN DE PROJET - PAR N° 107

# Analyse vitesse/rotation de la balle en sports de tables

*Élèves :*  
Guillaume PACZEK  
Baptiste PERREYON

*Enseignants :*  
Romain VUILLEMOT  
Louis GOSTIAUX

## Résumé

L'utilisation d'outils informatiques dans les sports est de nos jours beaucoup plus fréquente. En particulier, la vision par ordinateur permet de traiter un flux de données plus rapidement que l'humain et possède de nombreuses applications.

Ce projet propose un système de tracking performant dans deux sports aux problématiques différentes : le tennis de table et le billard. Dans le premier, la balle est plus difficile à détecter du fait de sa petite taille et de sa grande vitesse. Dans le deuxième, l'exigence sur la précision de la localisation est plus sévère car des mouvements de bien plus faible amplitude peuvent être observés lors du choc entre deux billes, élément déterminant des règles du billard. À cela s'ajoute que le mouvement de la balle de tennis de table est fondamentalement tridimensionnel, là où un mouvement plan constitue une bonne modélisation de la trajectoire de la bille de billard. L'analyse du mouvement diffère donc grandement dans ces deux sports.

C'est pourquoi le premier est accompagné d'une modélisation du mouvement qui permettrait d'exploiter les données obtenues par le tracking pour analyser le mouvement de la balle. Pour le billard s'ajoute un script permettant aux joueurs de rejouer leurs coups à volonté, phase importante dans l'entraînement des amateurs comme des professionnels.

## Abstract

Nowadays, the use of algorithms in sports is a common practice. For instance, computer vision aims at processing a data stream much faster than the human eye could. Therefore, it can be used in many fields.

This project presents an efficient ball tracking system for disciplines with different demands: table tennis and billiards. Table tennis features a smaller ball and higher velocities, which means a harder detection. Billiards' balls may be bigger, but the position detection must be much more precise in order to describe accurately bounces and slow trajectories. Indeed, bounces between balls make up a major part of the rules of billiards. Moreover, the trajectory of a table tennis ball is tridimensional, whereas plane motion is a sufficient model for billiards. As a result, motion analysis of these two sports will differ a lot.

Thus, the table tennis tracking system includes a trajectory modelling, which could supply the user with useful information about the detected motion. As for billiard, the system features a script which records the ball's trajectory and allows the user to freely play and replay their shots.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Localisation de la balle sur l'image</b>	<b>6</b>
2.1	Conditions de tracking . . . . .	6
2.2	Principe et hypothèses de l'algorithme . . . . .	7
2.3	Performances . . . . .	8
2.4	Améliorations possibles . . . . .	12
2.5	Conclusion sur le tracking . . . . .	13
<b>3</b>	<b>Modélisation du mouvement au tennis de table</b>	<b>14</b>
3.1	Trajectoire libre . . . . .	14
3.2	Rebond de la balle . . . . .	17
3.3	Conclusion sur la partie modélisation . . . . .	19
<b>4</b>	<b>Analyse du mouvement au billard</b>	<b>20</b>
4.1	Hypothèses et méthodologie . . . . .	20
4.2	Utilisation des données . . . . .	22
4.3	Fonctionnalité de rejet . . . . .	22
4.4	Déroulement du script pour le billard . . . . .	23
<b>5</b>	<b>Conclusion</b>	<b>26</b>
<b>6</b>	<b>Remerciement</b>	<b>28</b>

## Table des figures

1	Exemples d'applications de notre projet . . . . .	4
2	Placement des caméras en salle Amigo . . . . .	6
3	Masque binaire, barycentre (jaune), centre du cercle circonscrivant (vert) . . . . .	8
4	Image avec la position de la balle et image binaire. . . . .	9
5	Pointage manuel des positions de la bille à l'aide de <i>Tracker</i> . . . . .	9
6	Erreur de tracking sur un coup à rebonds . . . . .	10
7	Erreur de tracking sur un coup droit . . . . .	10
8	Erreur de tracking en fonction de la vitesse, prises sur plusieurs vidéos. . . . .	11
9	Erreur de tracking pour le tennis de table . . . . .	12
10	Axes du problèmes . . . . .	14
11	Trajectoire de la balle sans effet avec un effet top-spin et un effet back-spin	15
12	Différents effets au tennis de table . . . . .	16
13	Pointage manuel de la trajectoire de la balle . . . . .	16
14	Schéma général du rebond . . . . .	17
15	Schéma du rebond top-spin . . . . .	17
16	Pointage manuel du rebond d'une balle . . . . .	18
17	Systèmes d'axes de la caméra et du mouvement réel . . . . .	21
18	Répartition des zones de l'image projetée . . . . .	21
19	Le mode rejeu . . . . .	23
20	Modes de fonctionnement 1 et 2 . . . . .	25

# 1 Introduction

Dans la pratique de sport de haut niveau, il est de plus en plus courant d'utiliser des outils numériques dans l'analyse de données et des performances, afin de guider au mieux les athlètes dans leur progression. De ce fait, la nécessité d'analyser des vidéos sportives et de modéliser physiquement les phénomènes intervenant dans les sports s'est imposée. Le tennis de table, sport très populaire en Europe et en Asie, ne fait pas figure d'exception et de nombreux travaux ont déjà été menés. On pense en particulier aux travaux de Patrick Wong et Laurence Dooley [1] qui utilisent deux caméras pour obtenir une vision stéréo et qui se proposent de détecter la position de la balle dans l'espace à des fins d'arbitrage. On pense aussi au réseau neuronal TTNet [2] qui parvient à détecter la position de la balle au pixel près sur des vidéos full HD.

Un autre sport pour lequel l'analyse vidéo peut s'avérer utile est le billard. Cependant, du fait de sa moins grande popularité par rapport au tennis de table au niveau international, peu de recherches s'y sont attelées. On peut tout de même citer les travaux de l'université de Vienne de tracking vidéo automatique [3] et le projet de réalité augmentée qui détermine et affiche le meilleur coup possible à partir d'une image de la table provenant d'une caméra positionnée sur le joueur [4].

S'il peut être intéressant d'observer les mouvements des athlètes, ce projet se limite à l'analyse du mouvement de la balle mais s'intéressera donc à deux sports : le tennis de table et le billard. Il s'agit non seulement de simplifier et d'améliorer la qualité de l'entraînement des sportifs en proposant des analyses de données précises mais aussi d'avoir une vocation d'aide réelle pour les fédérations sportives.

Enfin le contexte des jeux olympiques de 2024 vient renforcer l'importance de moyens fiables pour l'analyse de données sportives. En effet le projet Sciences 2024, un programme de recherche collectif dédié à l'accompagnement des athlètes français dans leur quête de titres aux Jeux Olympiques et Paralympiques de Paris 2024, pourrait servir de cadre à ce projet car l'IA peut apporter un regard nouveau ou des outils d'analyse plus perfectionnés que l'œil humain ne le faisait jusqu'alors.



(a) Détection automatique de la position de la balle (cercle jaune) sur une vue latérale prise en salle Amigo



(b) Tracé en temps réel de la trajectoire d'une bille en transformant sa position détectée par la caméra en position à projeter sur le billard

FIGURE 1 – Exemples d'applications de notre projet

Dans les deux sports il s'agira de suivre le mouvement de la balle et de proposer une modélisation théorique (phénomène de rebond, d'effet ...) permettant d'analyser plus finement le mouvement réel. Dans cette optique, on se fixe l'objectif de développer un système automatique de tracking de balles avancé et déployable dans des lieux tels que des clubs ou l'INSEP. Selon le sport des contraintes très différentes s'imposent : la balle de ping-pong peut atteindre la centaine de kilomètre par heure, sa détection sur une image peut donc être difficile. Au billard, la vitesse est bien plus faible, mais la précision de la détection en position se doit d'être bien plus précise. En effet, la détection de chocs entre deux boules est un aspect déterminant des règles du billard et ces derniers peuvent entraîner de très petits déplacements, donc difficilement détectables, là où le rebond de la balle de ping-pong entraîne une modification plus nette de la trajectoire. De plus là où le billard peut facilement se ramener à un modèle bidimensionnel, le tennis de table constitue un problème fondamentalement 3D.

Les objectifs de ce projet sont de :

- Suivre la position de la balle avec précision.  
**L'écart entre la position réelle de la balle et la position donnée par le système lors de son mouvement libre ne diffère pas de plus de 2 cm au tennis de table et 1cm au billard dans 80% des cas.**
- Modéliser la trajectoire de la balle fidèlement à la réalité.  
**L'écart entre la trajectoire obtenue par modélisation et la réalité ne diffère pas de plus de 5cm au tennis de table et 1 cm au billard. Cet objectif sera validé à partir de pointage vidéo manuel sur un jeu de 30 vidéos présentant les différents coups classiques (avec ou sans effets).**

## 2 Localisation de la balle sur l'image

Cette section détaille la stratégie d'implémentation de l'algorithme de tracking de la balle au billard et au tennis de table, ainsi que ses résultats.

### 2.1 Conditions de tracking

Dans cette section, nous allons présenter les conditions et hypothèses du montage fait en salle Amigo.

Les caméras utilisées sont des Logitech Brio filmant en 640x480 pixels à 30 images par secondes. Ces dernières restent à une position fixe. Pour le billard, une seule caméra est utilisée tandis que deux sont nécessaires au tennis de table, afin de permettre de reconstituer la trajectoire de la balle en trois dimensions. La position des différentes caméras est visible sur la figure 2.

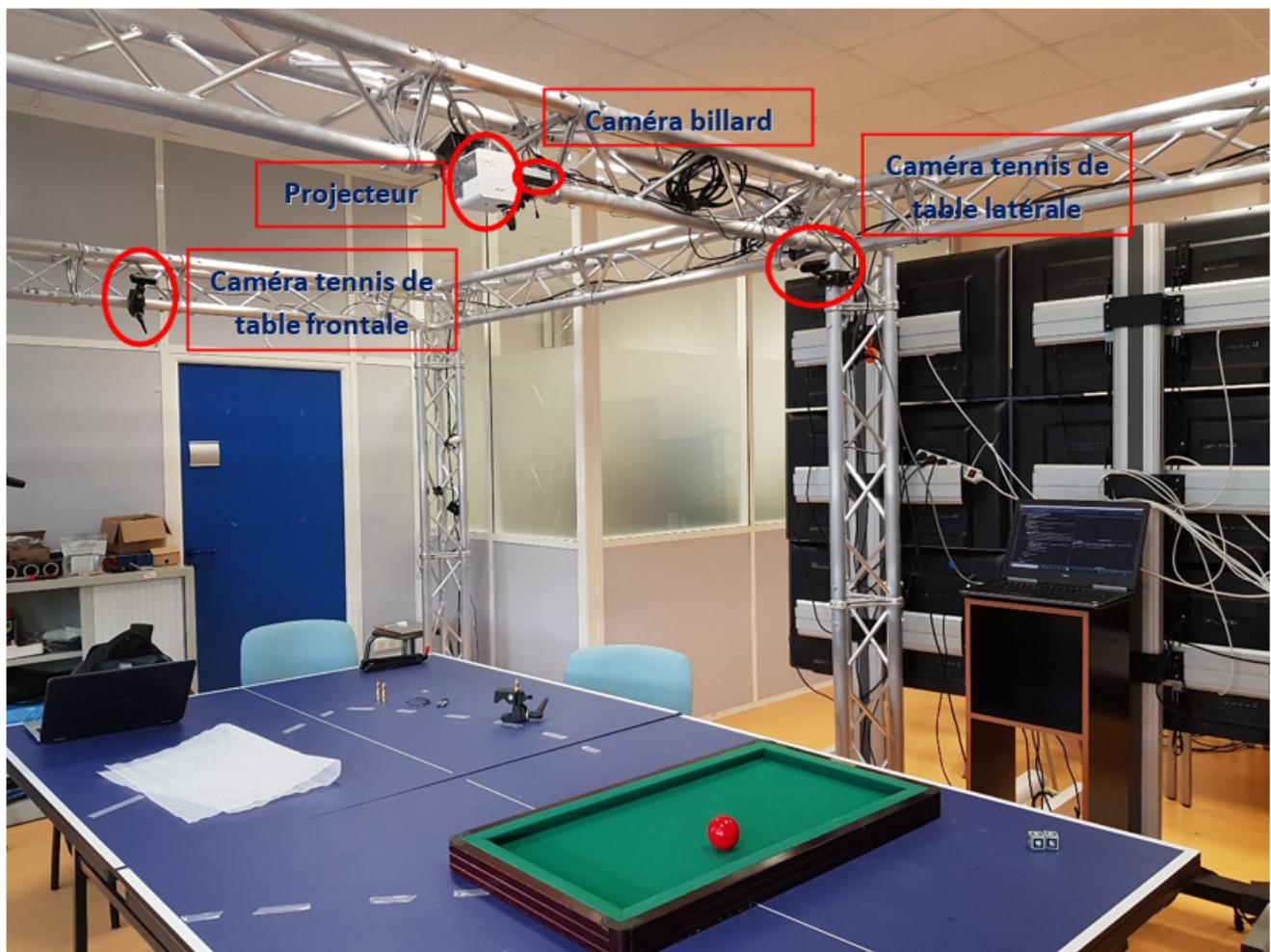


FIGURE 2 – Placement des caméras en salle Amigo

Comme visible sur la figure ci-dessus, un projecteur est également monté au-dessus de la table. Il permet de projeter en temps réel l'écran de l'ordinateur qui gère le programme. Cet élément n'est pas utile au tracking, il sera développé dans la partie 4.

Enfin, en plus de l'hypothèse de l'immobilité des caméras vient s'en ajouter une déterminante : l'algorithme de tracking ne détecte qu'un seul objet par image. Si cette hypothèse fait sens au tennis de table, elle restreint beaucoup le cas du billard où on ne s'intéressera donc qu'au tracking de la bille rouge.

Dernièrement, pour assurer le fonctionnement inconditionnel de l'algorithme, on suppose qu'il n'y a pas d'éléments du décor possédant une couleur trop proche de celle des objets à tracker. En pratique, ça n'est presque jamais le cas et cette hypothèse est vérifiée. On peut presque toujours trouver des bornes de couleurs (cf sous-section suivante) suffisamment serrées pour empêcher que cela n'arrive.

## 2.2 Principe et hypothèses de l'algorithme

On détaillera maintenant le fonctionnement de l'algorithme de tracking.

L'algorithme utilisé se base sur un blog [5] implémentant le suivi d'une balle verte beaucoup plus grande ( $\approx 20$  cm de diamètre). On en a retenu la méthode : c'est une recherche par couleur, puis par forme.

Avant de faire fonctionner l'algorithme pour la première fois, il faut déterminer deux paramètres importants : les bornes supérieures et inférieures de recherche de la balle dans l'espace de couleur HSV. On ne pourra détecter la balle que si sa couleur se situe entre ces deux bornes. Cet espace de couleur est plus adapté au tracking d'objet que le classique RGB car il dépend moins de l'éclairage et prend en compte la vraie couleur (hue = teinte en anglais). Ces paramètres permettent donc d'indiquer à l'algorithme des zones potentielles de présence de la bille. Leur détermination se fait à l'aide de l'outil interactif (*range-detector-video.py*) et ces paramètres sont supposés fixés par la suite.

L'algorithme de tracking consiste en une unique fonction prenant en argument :

- Une image RGB (issue par exemple d'une vidéo, ou directement de la caméra)
- Des bornes de couleurs dans l'espace HSV

Cette dernière renvoie les coordonnées du centre de la balle/bille, dans le cas où elle a été trouvée.

Partant de l'image dans laquelle nous souhaitons trouver la position de la balle, on doit d'abord effectuer une série de transformations (pré-processing) :

- Floutage par filtre gaussien
- Conversion RBG en HSV
- Binarisation selon les bornes HSV souhaitées

Le floutage permet la suppression du bruit. Après la binarisation, seuls les pixels de la couleur recherchée sont non nuls. On effectue ensuite une recherche de contours sur cette image binaire (fonction *cv2.find\_contours* de la bibliothèque OpenCV). Ensuite, il faut choisir le meilleur contour parmi ceux trouvés. On pourrait imaginer des heuristiques de choix complexes, mais dans le cas du billard, il n'y a souvent qu'un seul contour candidat, correspondant à la bille. On se contente de choisir le contour de plus grande aire et de vérifier que sa taille soit cohérente.

Une partie du pré-processing consistait en la soustraction du fond. En procédant ainsi, on peut espérer effacer les éléments du décor et avoir une meilleure détection. Plusieurs méthodes ont été proposées pour le calcul du fond : fond moyen calculé sur toutes les images de la vidéo (impossible à implémenter en temps réel), moyenne glissante autour de l'image actuelle ou encore fond statique obtenu lorsqu'aucun objet mobile ou joueur n'est présent sur l'image. Cette partie a finalement été abandonnée car les résultats étaient peu satisfaisants pour le tennis de table et la méthode semblait peu utile pour le billard, compte tenu des performances actuelles.

La dernière étape de l'algorithme est la détermination du centre du contour sélectionné. Initialement, on définissait la position du centre comme étant le barycentre des points du contour. Le problème de cette méthode est que le contour n'est jamais un cercle parfait, à cause de deux facteurs :

- À haute vitesse, la bille s'allonge sur l'image prise par la caméra.
- Le reflet des sources lumineuses sur la bille perturbe la captation du rouge de la bille.

Le centre est alors souvent détecté à côté de sa véritable position.

On utilise donc finalement une autre méthode de détermination du centre : le cercle circonscrivant aux pixels du contour. Il correspond au plus petit cercle tel que le contour détecté y soit inscrit. La position de la bille est alors définie comme le centre de ce cercle.

Cette méthode est plus robuste aux déformations de la balle ainsi qu'à son effacement partiel à cause de l'éclairage et fournit donc de meilleurs résultats en termes de précision. La figure 3 met en évidence la différence entre les deux méthodes de détermination du centre. On y observe que le masque n'est pas du tout circulaire et que le barycentre est par conséquent décalé sur la gauche. En revanche le cercle circonscrivant permet de bien prendre en compte la partie du masque qui a été effacée et détermine mieux la position réelle de la bille.



FIGURE 3 – Masque binaire, barycentre (jaune), centre du cercle circonscrivant (vert)

La figure 4 présente une visualisation possible du fonctionnement de l'algorithme. On y voit le masque ne laissant passer que 2 zones candidates et l'algorithme sélectionne celle correspondant à la bille.

### 2.3 Performances

Pour mesurer les performances de l'algorithme de tracking, on se munit d'un jeu de coups enregistrés dans la salle Amigo, supposé reproduire des conditions réelles de jeu.

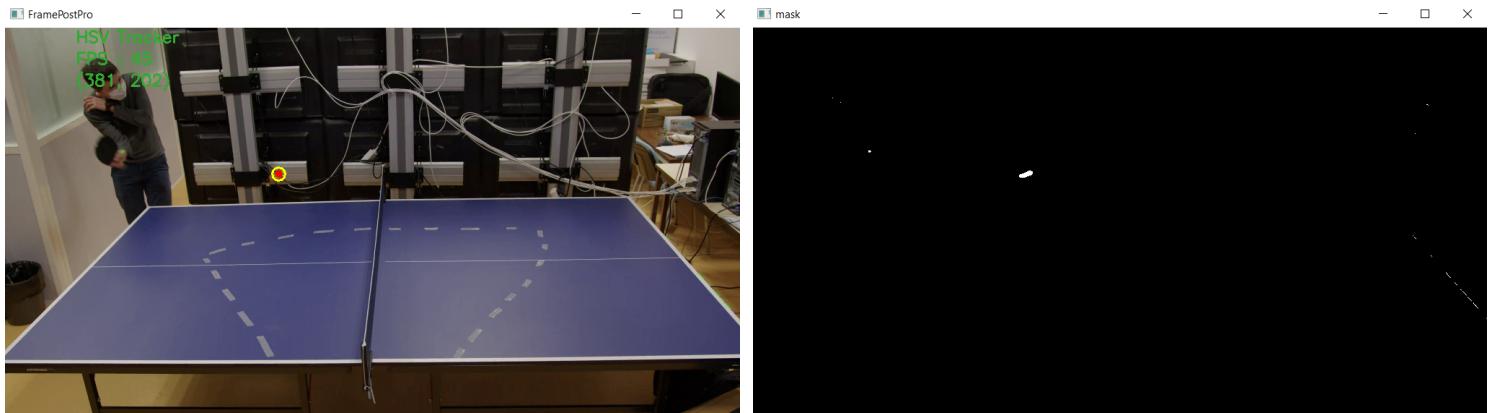


FIGURE 4 – Image avec la position de la balle et image binaire.

On suppose que le bon fonctionnement de l'algorithme sur ce jeu de données valide le fonctionnement de celui-ci. On mesure alors la précision de l'algorithme de tracking par la distance entre la position trackée et la position pointée à la main sur le jeu de vidéos. Un exemple de pointage manuel est donné figure 5

On s'intéresse à deux critères de performance : le taux de détection de la balle, c'est à dire le pourcentage d'images sur lesquelles on trouve la balle et l'erreur de tracking qui mesure la distance entre la trajectoire pointée à la main et la trajectoire trackée par l'algorithme. On essayera d'expliquer les sources d'erreur ainsi que de trouver des pistes d'amélioration.



FIGURE 5 – Pointage manuel des positions de la bille à l'aide de *Tracker*

On rappelle que le critère de précision pour le billard est : 1 cm d'écart maximum, dans 80% des cas. En tenant compte de la résolution des webcams utilisées, cela correspond à un écart maximum de 7,7 pixels. On trace l'erreur de tracking en fonction du temps, ainsi que l'erreur moyenne, pour plusieurs trajectoires du jeu de coups. On ajoute aussi un trait pointillé correspondant aux 7,7 pixels objectifs.

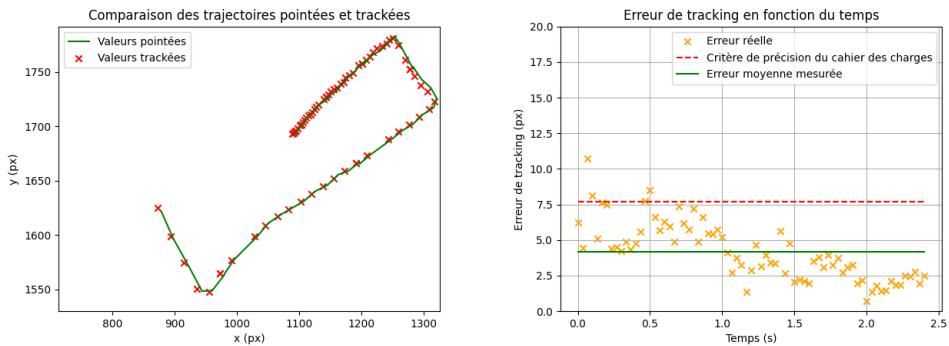


FIGURE 6 – Erreur de tracking sur un coup à rebonds

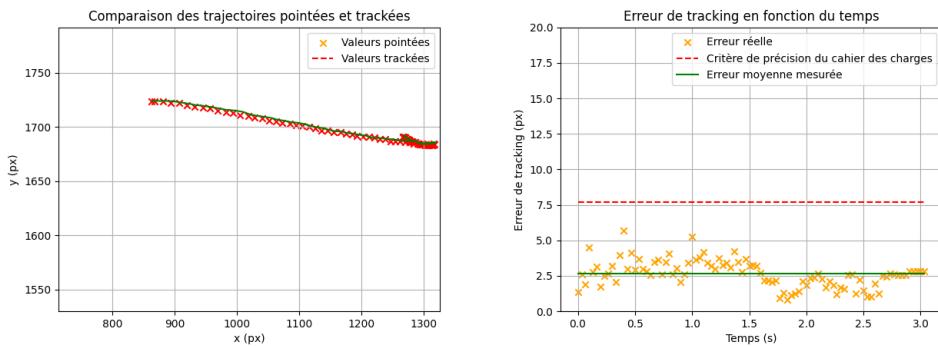


FIGURE 7 – Erreur de tracking sur un coup droit

Les figures 6 et 7 montrent l'erreur de tracking en fonction du temps, ainsi que les trajectoires pointées et trackées superposées. En particulier, on remarque que :

- Plus de 80% des écarts sont inférieures au critère imposé, on valide le cahier des charges
- L'erreur moyenne est presque 2 fois plus petite que le critère.
- Les plus grandes erreurs correspondent aux débuts de trajectoires, à haute vitesse. L'erreur semble ainsi décroître avec la vitesse.

Le critère de précision est ainsi validé haut la main. L'erreur semble osciller autour de l'erreur moyenne, que l'on peut interpréter comme du bruit ou de l'erreur de pointage. L'erreur moyenne sur toutes les trajectoires est autour de 4 pixels, ce qui signifie un écart moyen de 5 mm. La table mesurant 80 cm de longueur, cela représente une incertitude de position de 0.625%. De plus, on atteint un remarquable taux de détection de 100%, ce qui signifie que la balle est trouvée sur toutes les images de la vidéo.

La figure 8 montre pour le billard l'erreur de tracking en fonction de la vitesse de la balle. Il semble bien qu'on puisse perdre en précision pour des hautes vitesses. On peut identifier deux potentielles raisons, toutes liées aux déformations de la bille sur l'image à haute vitesse :

- La bille possède un diamètre de 3 cm et on recherche une précision de 5 mm, soit 1/6 de la dimension de l'objet. Lorsque la bille se déforme à haute vitesse, il est

d'autant plus difficile de pointer avec précision le centre de la balle. On s'attend donc à augmenter en moyenne l'erreur de pointage, qui s'apparentrait à une erreur de tracking.

- Mais l'erreur de tracking peut tout aussi bien venir de l'algorithme lui-même. Il se peut qu'à cause des déformations de la bille, la fonction de tracking soit moins performante.

Quoi qu'il en soit, l'erreur dépasse rarement les 7,7 pixels du cahier des charges. Dans le cadre d'utilisation de l'algorithme, les erreurs à haute vitesse ne sont pas gênantes. D'une part, la bille n'est rapide qu'en début de coup et ralentit rapidement, les mesures à haute vitesse ne représentent qu'une faible part des mesures. D'autre part, pour un tel algorithme la précision à basse vitesse est plus importante qu'à haute vitesse, notamment pour la détection de l'arrêt.

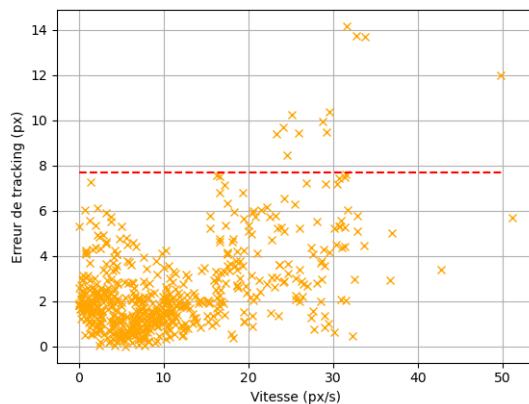


FIGURE 8 – Erreur de tracking en fonction de la vitesse, prises sur plusieurs vidéos.

En appliquant les mêmes graphes pour le montage du tennis de table, on trouve des résultats légèrement différents. Le critère de précision sur la position est aussi respecté : 1,1 cm d'erreur moyenne sur les 2 cm demandés. Cependant, le taux de détection de la balle ne dépasse pas 85%. Les graphes sont présentés en figure 9. On voit alors que lorsque la balle est détectée, elle l'est avec une faible erreur. Mais il arrive souvent qu'elle soit détectée au mauvais endroit (très grande erreur), voire pas du tout détectée (absence de point sur la trajectoire).

Notons que les problèmes de tennis de table et de billard ne sont pas de même nature. Pour le billard, la difficulté est l'estimation précise du centre. Pour le tennis de table, la difficulté est de détecter la balle, compte tenu de sa petite taille et de sa haute vitesse. Au billard s'ajouteraient éventuellement la difficulté de tracker plusieurs objets en même temps et de les suivre sans les mélanger. De plus, même si la zone de recherche est connue et possède un fond uniforme, le problème n'est pas évident si l'on s'impose une précision trop importante. Ceci explique les différences de performance de notre algorithme sur les deux montages.

Le cahier des charges initial ne spécifiait pas de critère de rapidité pour l'algorithme, si ce n'est qu'il puisse tourner en temps réel. Il se trouve qu'un appel à la fonction de tracking

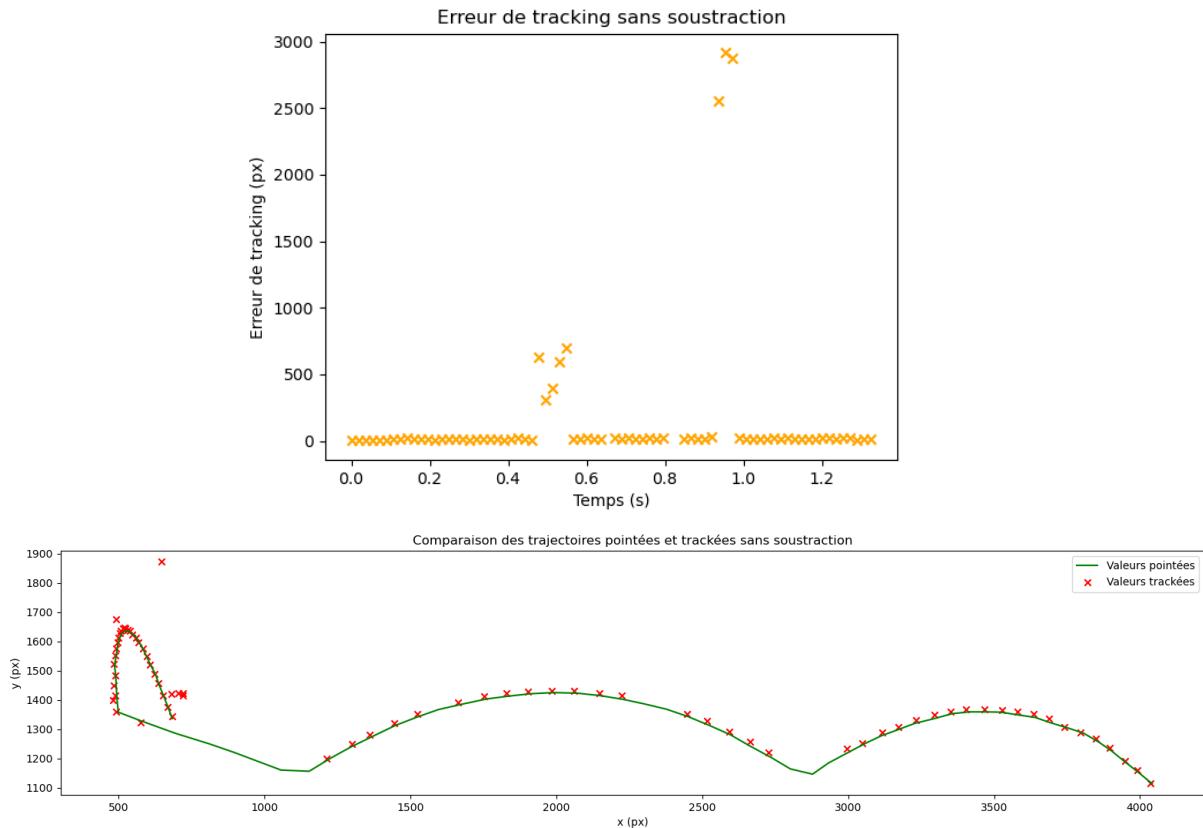


FIGURE 9 – Erreur de tracking pour le tennis de table

dure en moyenne 1 ms. C'est un bon score car cela signifie qu'un algorithme de tracking en temps réel pourrait très bien suivre une vidéo à 30, 60, voire 240 images par seconde, tant que les autres opérations de la boucle ne sont pas trop longues. En pratique, nous avons réussi à implémenter sans problème cette fonction dans des algorithmes de suivi en temps réel et même à coupler le tracking à des fonctionnalités visant à améliorer l'expérience de jeu de l'utilisateur (cf partie 4).

## 2.4 Améliorations possibles

La vitesse de la fonction de tracking pourrait encore être améliorée. En effet, l'image est lue telle quelle alors qu'on pourrait la redimensionner avant de la traiter. On pourrait aussi sélectionner une région d'intérêt, pour éviter d'avoir à traiter toute l'image (particulièrement vrai au billard où l'on connaît la position de la table). Ces deux opérations réduiraient déjà considérablement le nombre de pixels à traiter et donc le temps d'exécution de la fonction de tracking. On pourrait aussi améliorer la précision et le taux de détection des deux algorithmes en prédisant les positions suivantes. Cela serait particulièrement utile en tennis de table où la balle est difficile à détecter et à distinguer des autres contours du fait de sa taille.

## 2.5 Conclusion sur le tracking

Ainsi, on a pu implémenter une méthode de tracking donnant des résultats très satisfaisants en terme de précision et rapidité. Les hypothèses sont raisonnables et permettent déjà de rejouer des coups en billard (cf partie 4). Quelques optimisations sont encore possibles, mais non nécessaires car les critères du cahier des charges sont déjà largement validés.

Pour compléter totalement l'algorithme de tracking, il faudrait ajouter le suivi de plusieurs billes simultanément au billard. Ceci n'a pas encore été abordé. Une fois l'algorithme de tracking complètement implémenté et fonctionnel, sa rapidité et facilité d'utilisation lui donneraient l'avantage de pouvoir être utilisé dans n'importe quel contexte. Les possibilités sont alors larges : analyse de données, prédiction de trajectoires, entraînement de sportifs et amateurs.

### 3 Modélisation du mouvement au tennis de table

Modéliser le mouvement de la balle dans un sport a plusieurs intérêts. On peut citer la prédiction de la trajectoire ou l'analyse des effets qui sont utiles à l'entraînement des sportifs. Mais la modélisation du mouvement peut également aider à la reconstruction de la trajectoire à partir des positions trackées et palier d'éventuels manques de l'algorithme de tracking en indiquant les positions de la balle manquées par la caméra. Cela est particulièrement vrai dans le cadre du tennis de table où la balle est bien plus petite et surtout bien plus rapide qu'au billard. C'est donc pourquoi seul le tennis de table a été étudié.

Dans la suite du rapport, on entend par trajectoire libre la trajectoire de la balle entre deux rebonds. Ces derniers étant vus comme la cause de nouvelles conditions initiales à une nouvelle trajectoire libre. Dans un premier temps, on va présenter l'étude théorique de la trajectoire libre au tennis de table puis celle du rebond.

#### 3.1 Trajectoire libre

On fait l'hypothèse que la vitesse de rotation de la balle reste constante au cours de la trajectoire et que la balle est soumise à trois forces : son poids, les frottements de l'air dûs à son déplacement et l'effet Magnus dû à sa rotation. La rotation de la balle sur elle-même entraîne une différence de vitesse locale en dessous et au-dessus de la balle. Cela induit donc une différence de pression en accord avec la loi de Bernoulli qui se traduit par une force supplémentaire sur la balle. On considère le repère présenté sur la figure 10.

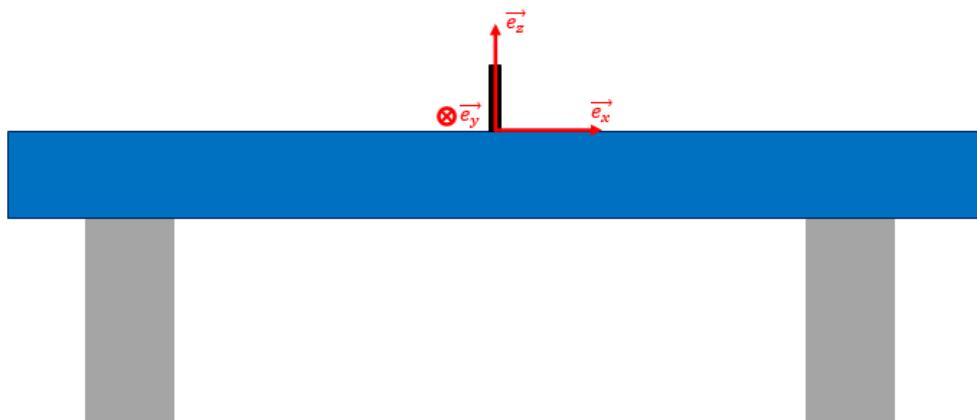


FIGURE 10 – Axes du problèmes

Le poids de la balle, noté  $\vec{P}$  vaut :  $\vec{P} = m\vec{g} = -mg\vec{z}$ .

Les frottements de l'air sur la balle, compte tenu de sa grande vitesse, peuvent être modélisés par :  $\vec{F}_{frict} = -\frac{1}{2}\rho C_x S v \vec{v} = -hv\vec{v}$ . On a  $h \approx 3 \cdot 10^{-4} \text{ kg.m}^{-1}$ .

L'effet Magnus induit une force  $\vec{F}_{magnus} = \alpha(\vec{\omega} \wedge \vec{v})$ . On a  $\alpha \approx 1,8 \cdot 10^{-5} \text{ kg}$ .

La deuxième loi de Newton appliquée à la balle donne lieu à un système de trois équations non linéaires :

$$\begin{cases} m\ddot{x} = -h\dot{x}\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} + \alpha(\omega_y\dot{z} - \omega_z\dot{y}) \\ m\ddot{y} = -h\dot{y}\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} + \alpha(\omega_z\dot{x} - \omega_x\dot{z}) \\ m\ddot{z} = -h\dot{z}\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} + \alpha(\omega_x\dot{y} - \omega_y\dot{x}) - mg \end{cases} \quad (1)$$

Ces équations non linéaires ont été résolues numériquement par la méthode d'Euler. Elles permettent de prédire la trajectoire de la balle une fois les conditions initiales de position, de vitesse et de vitesse de rotation données. La figure 11 présente les trajectoires prévues pour une balle lancée de la même position avec la même vitesse mais en y appliquant un effet (c'est à dire une vitesse de rotation) différent.

Pour rappel, avec les axes choisis sur la figure 10, un effet top-spin correspond par exemple à une rotation dans le plan  $(\vec{e}_x, \vec{e}_z)$  où la balle avance suivant  $\vec{e}_x$  et tourne vers l'avant et un effet back-spin correspond au cas où elle tourne vers l'arrière. Cela est présenté sur la figure 12.

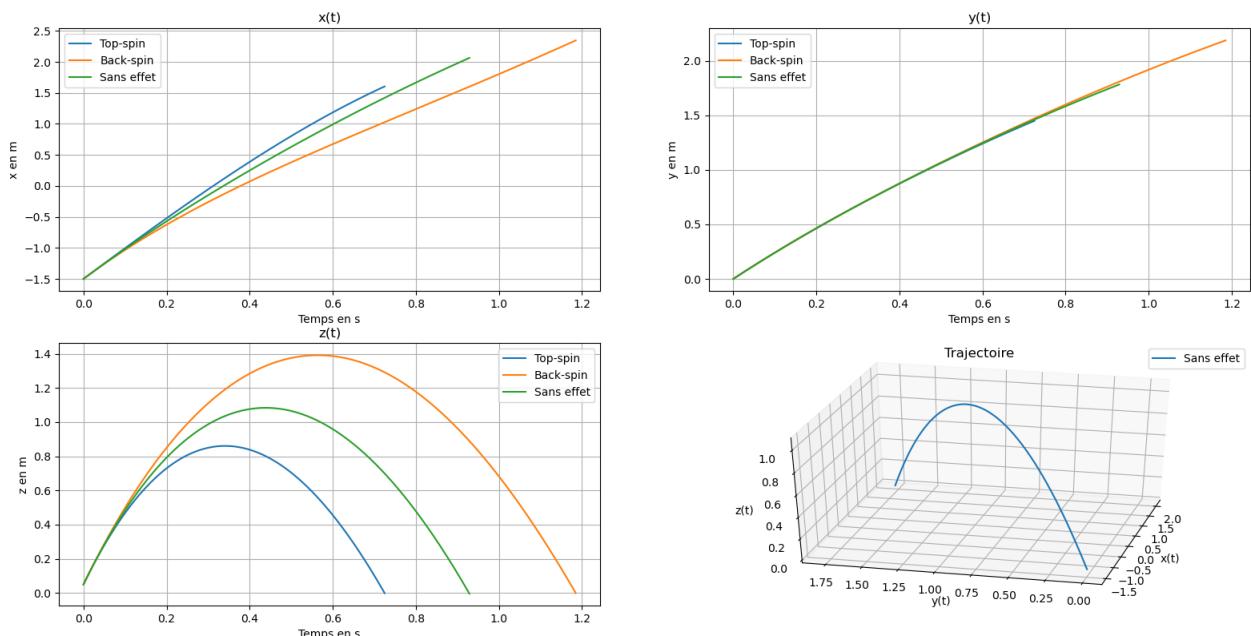


FIGURE 11 – Trajectoire de la balle sans effet avec un effet top-spin et un effet back-spin

Les trajectoires sont cohérentes avec ce l'on sait empiriquement : une balle liftée (effet top-spin) retombe plus vite alors qu'une balle coupée (back-spin) retombe plus lentement et va donc plus loin.

L'hypothèse d'une vitesse de rotation constante étant assez forte, il nous paraît intéressant de la vérifier expérimentalement. L'objectif est de mesurer la variation de vitesse angulaire  $\Delta\omega$  et de comparer les trajectoires obtenues par résolution numérique des équations avec les deux vitesses  $\omega_{min}$  et  $\omega_{max}$ . Ces trajectoires ne doivent pas différer de plus de 2,5 cm. Il était prévu de vérifier cela pour tous les effets décrits sur la figure 12. Avant de réaliser ces expérimentations, il a fallu s'assurer qu'il soit possible de mesurer la vitesse de rotation de la balle sur un pointage vidéo. On a donc tout d'abord pris une série de vidéos avec une caméra filmant l'ensemble de la table et effectué un pointage manuel comme visible figure 13. Malgré la petite taille de la balle sur l'image, il est possible de déterminer sa vitesse de rotation mais avec une précision plus faible qu'espérée. Le principal

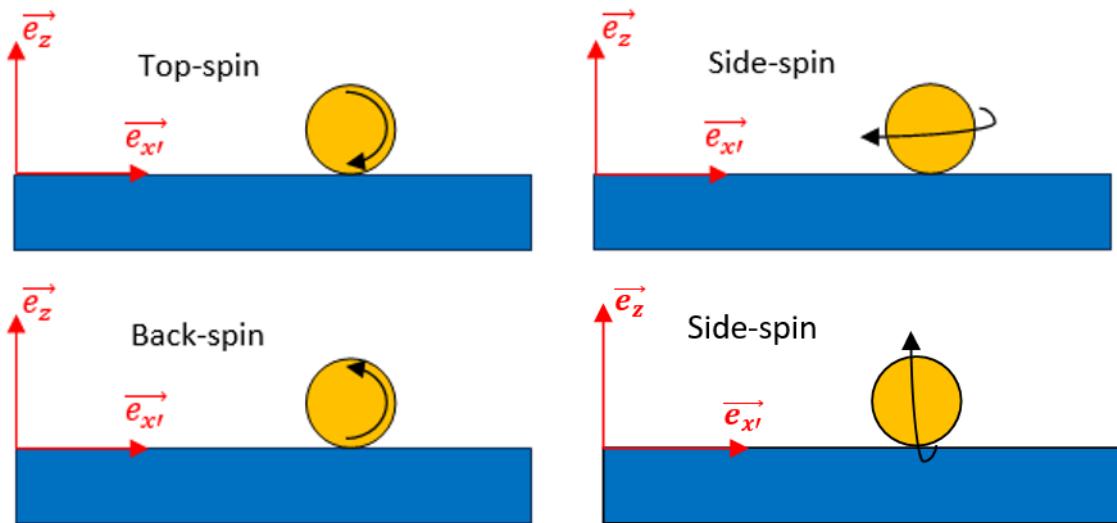


FIGURE 12 – Différents effets au tennis de table

problème réside dans le fait que la rotation de la balle se fait fondamentalement sur trois axes là où une vidéo, en 2D, ne peut en capter qu'une sur un seul axe, perpendiculaire au plan de l'image, avec précision.

Les points rouges représentent les différentes positions du centre de la balle tandis que les bleus sont ceux d'un point d'un axe quelconque de la balle. On a utilisé une balle bicolore pour mieux repérer cet axe.

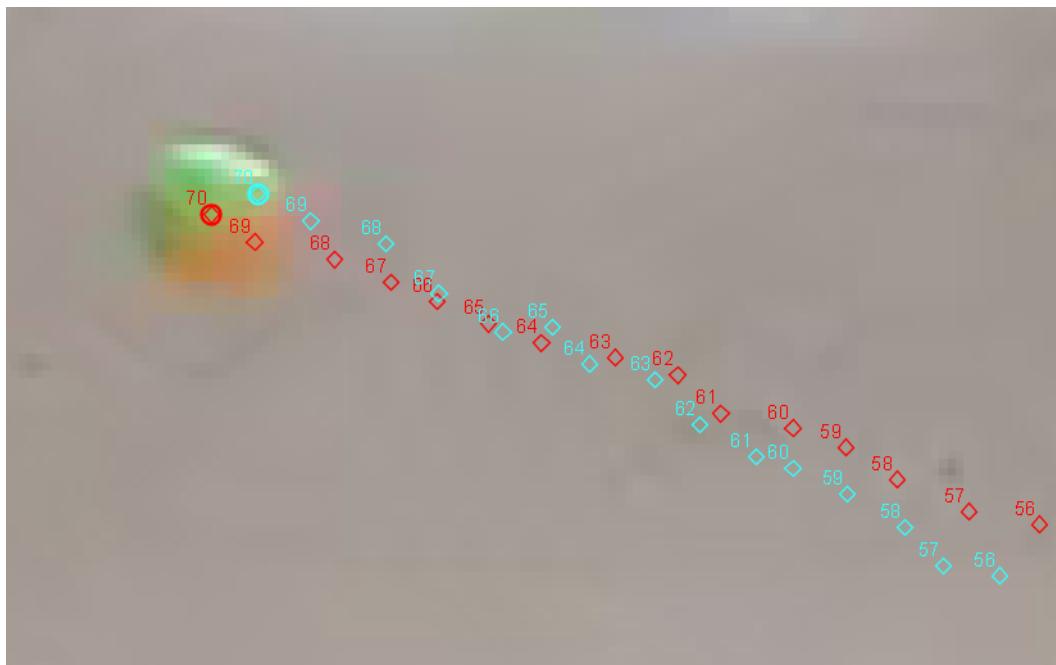


FIGURE 13 – Pointage manuel de la trajectoire de la balle  
En rouge le centre de gravité, en bleu un point d'un axe de la balle

### 3.2 Rebond de la balle

Dans notre approche de la modélisation de la trajectoire, le rebond de la balle de tennis de table sur la table peut être vu comme la création de nouvelles conditions initiales au mouvement de la balle. Le cas du rebond sur la raquette n'est pas traité.

Ainsi l'objectif de la modélisation du rebond au tennis de table est le suivant : compte tenu de données connues avant le rebond sur la vitesse de son centre d'inertie (norme, direction et sens) et sur sa vitesse de rotation (décomposée selon 3 axes), il s'agit de déterminer ces mêmes informations après le rebond.

Au rebond, la vitesse de la balle et sa vitesse de rotation peuvent se décomposer sur les trois axes du repère ( $\vec{u}_x$ ,  $\vec{u}_y$ ,  $\vec{u}_z$ ). On note  $\vec{u}_{x'}$  l'axe du plan ( $\vec{u}_x$ ,  $\vec{u}_y$ ) tel que  $v_x \vec{u}_x + v_y \vec{u}_y = v_{x'} \vec{u}_{x'}$ . La vitesse de rotation de la balle peut également être décomposée suivant trois axes, mais on ne s'intéressera qu'au cas où la vitesse de rotation de la balle est parallèle à  $\vec{u}_{y'}$ . Pour traiter tous les types de rebond possibles, on fait l'hypothèse que les conséquences des effets de rotation de la balle sont indépendantes entre elles. Cela permet de traiter séparément les trois cas se présentant : lorsque la vitesse de rotation est autour de l'axe  $\vec{u}_{y'}$  – ce qui correspond aux effets top-spin et back-spin – et lorsqu'elle est orthogonale à cet axe – ce qui correspond à un effet side-spin (selon l'axe  $\vec{e}_x'$  ou  $\vec{e}_z'$ ).

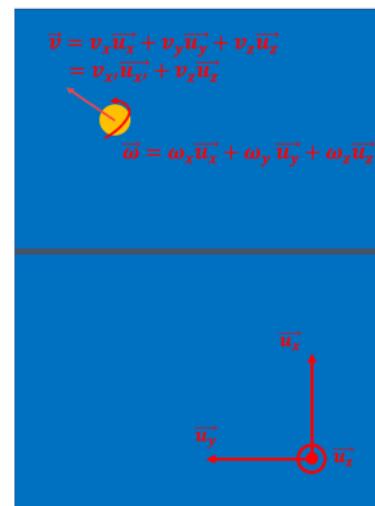


FIGURE 14 – Schéma général du rebond

#### Cas de la rotation autour de l'axe $\vec{u}_{y'}$ : effet top-spin et back-spin

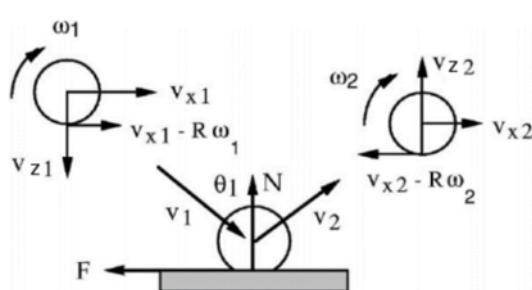


FIGURE 15 – Schéma du rebond top-spin

Des études, notamment celle de Rod Cross [6] permettent d'obtenir ces relations en posant deux coefficients :  $e_z = -\frac{v_{z2}}{v_{z1}}$  et  $e_x = -\frac{v_{x2} - R\omega_2}{v_{x1} - R\omega_1}$  et en utilisant la conservation du moment cinétique au point de contact entre la table et la balle  $I\omega_1 + mRv_{x1} = I\omega_2 + mRv_{x2}$ . On a alors :

Le cas ici présent est plus simple à traiter car le problème peut se ramener à un problème plan. La balle arrive vers la table avec une vitesse  $\vec{v}_1 = v_{x1} \vec{u}_{x'} + v_{z1} \vec{u}_z$  et une vitesse de rotation  $\vec{\omega}_1 = \omega_1 \vec{u}_{y'}$ . On utilise la même notation pour les vitesses après le rebond.

L'objectif est d'obtenir des relations entre ces vitesses permettant de déterminer les vitesses après rebond.

$$\frac{v_{x_2}}{v_{x_1}} = \frac{1 - \alpha e_x}{1 + \alpha} + \frac{\alpha(1 + e_x)}{1 + \alpha} \cdot \frac{R\omega_1}{v_{x_1}} \quad (2)$$

$$\frac{\omega_2}{\omega_1} = \frac{\alpha - e_x}{1 + \alpha} + \frac{1 + e_x}{1 + \alpha} \cdot \frac{R\omega_1}{v_{x_1}} \quad (3)$$

$$\frac{v_{z_2}}{v_{z_1}} = -e_z \quad (4)$$

Le coefficient  $\alpha$  défini par  $I = \alpha m R^2$  vaut  $2/5$  dans le cas d'une sphère creuse. Le coefficient  $e_z$  reste globalement constant et on garde la valeur de  $0,78$ . En revanche  $e_x$  est variable, notamment en fonction de la vitesse de rotation, de la vitesse de la balle et de l'angle d'incidence. La difficulté est donc de le déterminer. On fait ici l'hypothèse suivante :  **$e_x$  ne dépend que de ces trois paramètres et des matériaux du contact. Chacun de ces paramètres agit de manière indépendante sur  $e_x$ .** Ainsi on a  $e_x = \lambda_{dm} \cdot f(v_1) \cdot g(\theta_1) \cdot h(\omega_1)$ .

Pour déterminer ces inconnues, nous avions prévu de mener des expérimentations. Elles consistaient en une série de lancers où un seul paramètre parmi  $v_1, \theta_1, \omega_1$  évoluerait à la fois, les autres étant donc fixés à une valeur connue. En filmant la balle au rebond, on pourrait alors mesurer par pointage vidéo manuel  $v_2, \theta_2, \omega_2$  pour déterminer  $e_x$ . En plus de déterminer les fonctions recherchées, on testerait l'hypothèse d'indépendance des paramètres.

Mais avant toute chose, nous avons voulu nous assurer que l'on puisse mesurer la vitesse et la vitesse de rotation de la balle aux instants proches du rebond. L'idée consiste donc à poser une caméra directement sur la table, assez proche du rebond. Après un premier essai non concluant, nous avons utilisé une caméra GoPro 8 à 240 fps et des images de 1920 par 1080 pixels. Un pointage de la vidéo réalisé avec le logiciel Tracker est visible figure 16

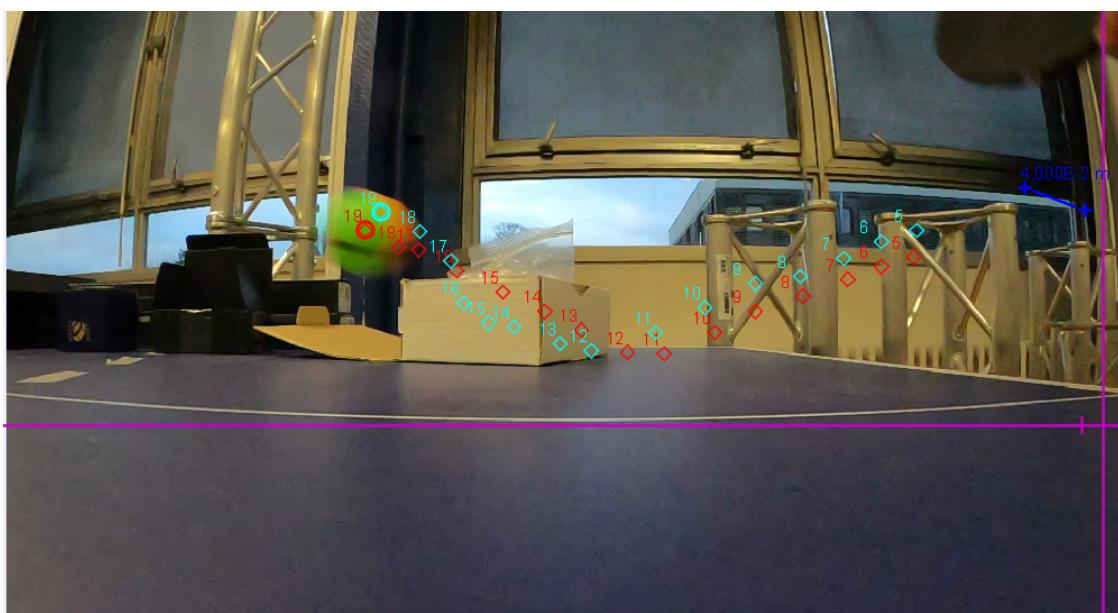


FIGURE 16 – Pointage manuel du rebond d'une balle

Les points rouges représentent les différentes positions du centre de la balle tandis que les bleus sont ceux d'un point d'un axe de la balle. On a utilisé une balle bicolore pour mieux repérer cet axe. Le pointage de deux points est nécessaire pour déterminer la vitesse de rotation de la balle, mais le centre de gravité de la balle seul suffit à déterminer sa vitesse de translation.

### 3.3 Conclusion sur la partie modélisation

La modélisation du mouvement peut s'avérer très utile dans le suivi de la trajectoire comme dans l'exploitation des données récoltées et nous avons pu proposer une première étape de cette modélisation. La suite nécessite cependant des expérimentations que nous n'avons pas pu mener. En revanche, nous avons montré qu'il était possible de les mener avec le matériel adéquat. Au tennis de table, c'est donc toute la partie du rebond qui doit être approfondie.

On peut de plus en garder la méthode pour la modélisation de la trajectoire au billard. En travaillant sous l'hypothèse des solides indéformables, on peut séparer trajectoire libre où la bille est soumise à son poids et à la réaction de la table, composée d'une force verticale annulant le poids et de frottements secs réduisant sa vitesse, et le rebond contre le bord de la table pouvant se caractériser par un coefficient de restitution de vitesse. Après quoi il faudra caractériser le choc entre deux billes et, si l'on veut affiner la modélisation faite, prendre en compte les effets que le joueur applique à la balle.

## 4 Analyse du mouvement au billard

Dans la partie suivante, on parlera uniquement de ce qui a été fait pour le billard, cas plus simple car la trajectoire de la bille est bi-dimensionnelle. Il s'agit donc de reconstituer la trajectoire de la bille dans le plan (en mm) à partir de ses positions sur les images prises par la caméra (en pixels), puis d'en tirer un certain nombre de données telles que la vitesse de la bille ou la détection d'évènements comme les rebonds et de les afficher à l'adresse de l'utilisateur. Cela a été implémenté dans un script présentant deux usages : le suivi en temps réel de la bille en proposant quelques données à l'utilisateur sur son coup et une fonctionnalité permettant de rejouer un coup enregistré.

### 4.1 Hypothèses et méthodologie

Dans toutes les sections suivantes les hypothèses suivantes sont valables. La table de billard, la caméra et le vidéo projecteur restent fixes. Le projecteur est situé au-dessus de la table de manière à ce que l'image projetée englobe complètement cette dernière et il ne déforme pas l'image par rapport à celle visible sur l'écran de l'ordinateur (un rectangle sur l'ordinateur sera donc visible comme un rectangle sur la table de billard). La table est entièrement visible par la caméra.

**Rq :** En salle Amigo, la caméra était également au-dessus du billard, mais cela n'est pas obligatoire.

Une première étape est de corriger la position de la balle obtenue par l'algorithme de tracking. En effet, la caméra utilisée induit une légère distorsion de l'image qui nuit à la précision de la détection effectuée. Heureusement le module OpenCV propose des outils de dé-distorsion de l'image mais ceux-ci nécessitent la connaissance de paramètres intrinsèques à la caméra. Nous avons donc tenté de les déterminer empiriquement à l'aide d'une méthode connue utilisant la rectitude du damier d'un échiquier pour mesurer la déformation induite par la caméra. Cependant, malgré deux expérimentations, les résultats obtenus n'ont pas été concluants. Il a donc été décidé de négliger cet aspect du traitement de la position détectée par l'algorithme de tracking.

Ensuite, il convient d'obtenir les coordonnées réelles de la bille dans un plan ortho-normé afin de pouvoir utiliser les lois de bases de la mécanique pour analyser son mouvement. On rappelle que la caméra n'a pas de position définie mais qu'elle reste cependant fixe. Pour atteindre l'objectif voulu, il faut trouver une relation homographique entre les deux plans visibles figure 17 : le plan de la caméra et une vue de dessus de la table. L'homographie est une transformation géométrique qui peut être mise sous forme de matrice, de sorte qu'une fois déterminée, elle peut n'être appliquée qu'à un seul point (les coordonnées en pixels de la bille) au lieu de toute l'image, ce qui permet d'économiser un grand temps de calcul.

Ainsi, au début de l'exécution du script, on va déterminer la matrice d'homographie recherchée. Pour cela, on place successivement la bille rouge dans les 4 coins du billard et on relève ses positions dans l'image. Ses positions réelles étant alors connues grâce aux dimensions du billard, on peut trouver la relation entre les deux plans grâce à une fonction d'OpenCV.

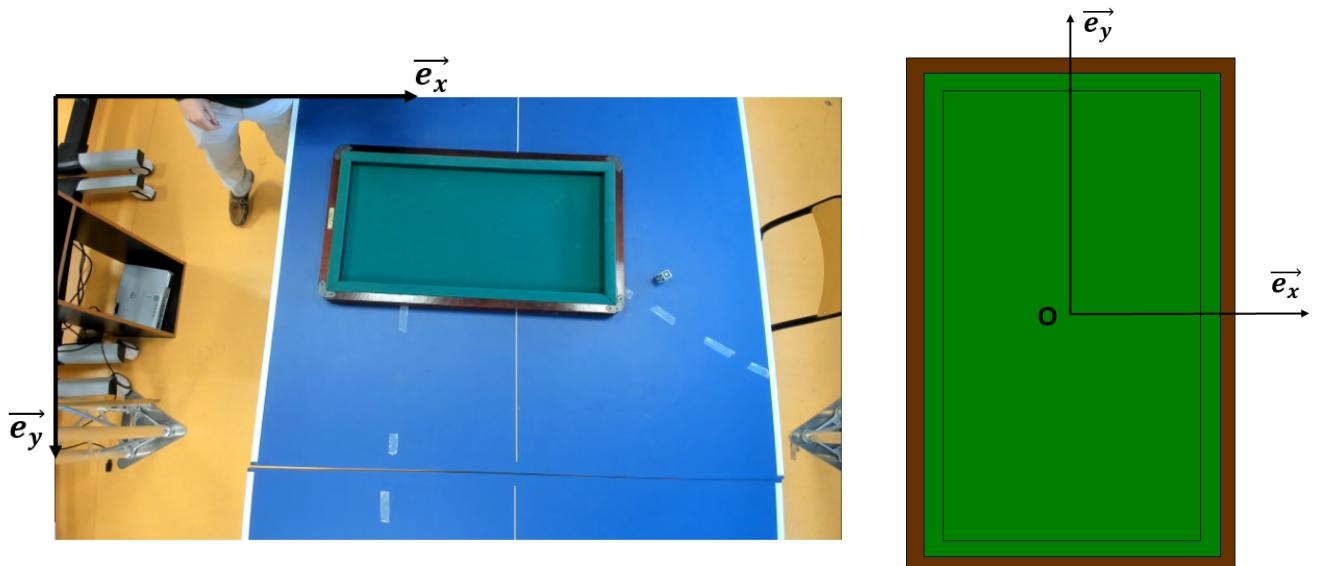


FIGURE 17 – Systèmes d’axes de la caméra et du mouvement réel

Reste enfin une dernière partie : l’exploitation et la communication des données obtenues. Le système est équipé d’un vidéoprojecteur permettant d’afficher en temps réel des informations sur et autour de la table. La table doit donc avoir une position bien définie par rapport au vidéoprojecteur (autrement les informations ne seraient pas affichées au bon endroit). C’est pourquoi, au démarrage du script, il est demandé à l’utilisateur de positionner la table à un endroit précis. Pour communiquer avec l’utilisateur, l’idée principale consiste à utiliser une image modèle sur laquelle on apporte des modifications pour afficher les informations désirées. Cette dernière est construite ainsi :

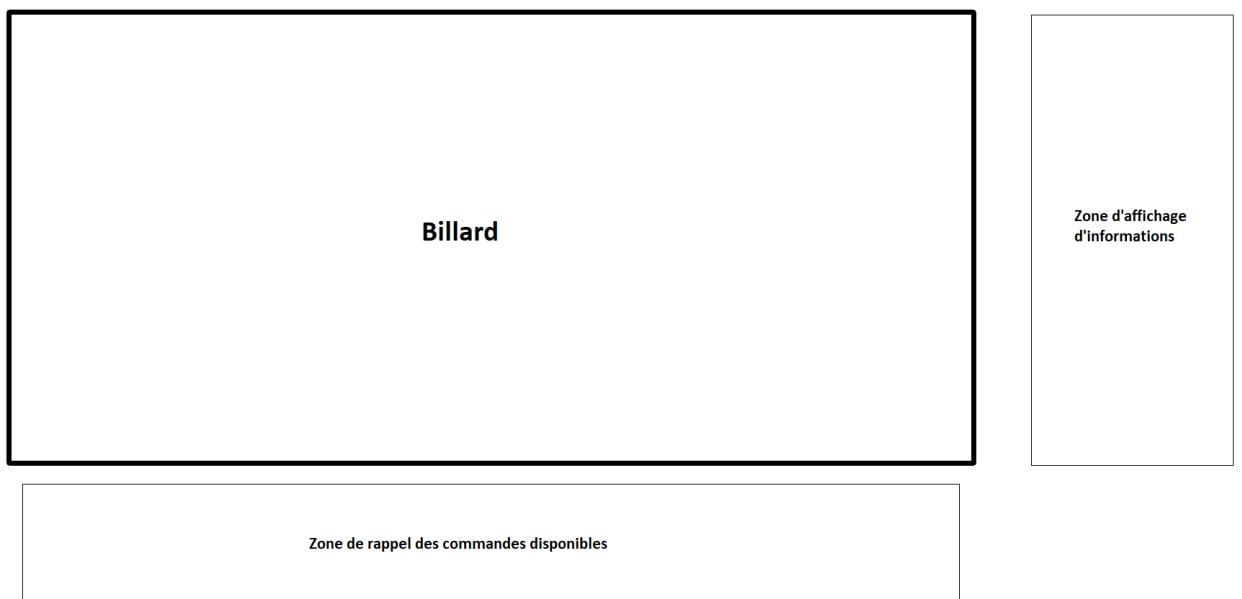


FIGURE 18 – Répartition des zones de l’image projetée

L'image à projeter étant directement affichée sur l'écran de l'ordinateur, elle est caractérisée par un nouveau système d'axes. Dans un souci de simplicité, le billard a une position horizontale sur l'image projetée ce qui rend la relation entre la position réelle de la bille et sa position sur l'image à afficher plus simple à déterminer mais justifie la nécessité d'avoir placé la table de billard à un endroit précis.

## 4.2 Utilisation des données

Dans le premier type de fonctionnement, l'utilisateur joue librement et on souhaite afficher des informations en temps réel sur le coup qu'il réalise. Nous avons choisi de concentrer nos efforts sur trois d'entre elles :

- Indiquer si la bille est en mouvement ou à l'arrêt. Pour cela, on compare l'écart maximum entre les positions des deux dernières secondes à un seuil fixé.
- Indiquer en temps réel la vitesse de la bille. Cette information, plus utile au joueur, est simplement calculée par le déplacement de la bille entre deux images enregistrées avec un pas régulier.
- Déetecter les rebonds de la bille sur le bord. Ici l'objectif est de détecter le rebond de la bille sur un bord de la table et de colorer différemment la partie de ce bord où le rebond a eu lieu jusqu'au début d'un nouveau coup. La détection du rebond est basée sur le changement de signe d'une des composantes de la vitesse de la bille.

## 4.3 Fonctionnalité de rejeu

Le deuxième type de fonctionnement est celui qui sera le plus utile au joueur. Lorsqu'il le souhaite, il peut enregistrer une trajectoire réalisée lors du premier mode de fonctionnement et la fonctionnalité de rejeu lui permet de retravailler son coup s'il veut le maîtriser ou de tenter une nouvelle approche s'il n'était pas satisfait du coup qu'il avait réalisé.

Cette fonctionnalité est divisée en trois étapes. Premièrement, le choix du coup à rejouer durant lequel les différentes trajectoires enregistrées sont affichées pour permettre au joueur de sélectionner le bon coup. Dans un deuxième temps, le programme affiche la position de la bille au début du coup sélectionné et attend que cette dernière soit positionnée correctement pour entrer automatiquement dans la dernière étape : la phase de rejeu proprement dite. La trajectoire enregistrée est affichée en pointillé et on trace en parallèle la nouvelle trajectoire réalisée par le joueur, qui pourra alors les comparer à la fin de son coup, comme visible sur la figure 19. Il lui est possible d'enregistrer la nouvelle trajectoire, de réitérer son essai, de choisir un nouveau coup à travailler ou encore de retourner au mode normal de fonctionnement.



FIGURE 19 – Le mode rejeu

#### 4.4 Déroulement du script pour le billard

Lorsque le script se lance, l'algorithme projette une image blanche avec un rectangle noir et demande à l'utilisateur de placer l'intérieur de la table de billard (c'est à dire la partie où les billes se déplacent) dans le rectangle. Les dimensions de ce rectangle ont été calibrées pour correspondre à la taille de la table et à la position du vidéoprojecteur en salle Amigo mais il est possible de le redimensionner en appuyant sur « r » puis en sélectionnant avec la souris le coin supérieur gauche puis le coin inférieur droit. Le placement de la table est primordial car il permet de garantir que les coordonnées réelles de la bille et du point la représentant sur l'image affichée par le vidéoprojecteur coïncident. Une fois que la table est placée, l'utilisateur doit appuyer sur « q » pour fermer l'image affichée et que le script continue. A partir de cet instant, la table ne doit plus bouger.

Une fois la table placée, la correspondance entre le deuxième système de coordonnées et le troisième est effectué. On va maintenant s'intéresser à la correspondance entre les deux premiers. Le but de cette partie est de trouver une homographie entre le plan de la caméra et celui de la vue de dessus de la table. À cet instant un choix s'impose à l'utilisateur : déterminer la matrice d'homographie à l'aide d'une fonction que nous avons créé ou utiliser celle sauvegardée à l'extérieur du programme dans le fichier « AllData.json » (par exemple si il a déjà effectué la procédure pour la déterminer la dernière fois qu'il a lancé le script, il n'a pas besoin de la déterminer à nouveau). La fonction permettant de déterminer l'homographie nécessite l'usage de la caméra qui, étant externe au PC, met un certain temps à se connecter, il faut donc être patient ! Lorsqu'une image venant de la caméra s'affiche, c'est qu'il est possible de commencer. L'algorithme demande à l'utilisateur de placer la boule rouge dans le coin supérieur gauche de la table puis de valider avec la barre d'espace ou la touche entrée. Une fois le premier angle détecté, l'algorithme demande à l'utilisateur de placer la boule dans les trois derniers angles. Connaissant ainsi la position de chacun des angles de la table dans le plan de la caméra et dans le système de coordonnées « réelles », on peut déterminer une relation géométrique entre les deux plans. Si le programme ne parvient pas à trouver d'homographie, c'est que la position des

angles a été mal détectée, on doit alors recommencer la procédure de localisation des 4 coins. Pour vérifier que tout a bien fonctionné, on demande à l'utilisateur de retirer la boule et on prend une nouvelle image de la table (après validation avec espace ou entrée) à laquelle on applique l'homographie déterminée ce qui doit afficher une vue de dessus de la table. L'utilisateur peut alors confirmer si l'homographie trouvée est correcte avec « y » ou l'infirmier avec « n » auquel cas on recommence la procédure de localisation des 4 coins.

Arrivé à cette étape, toutes les procédures préliminaires ont été effectuées, on peut rentrer dans le cœur du programme. Le programme comporte deux modes principaux, le jeu libre et la capacité de rejouer des coups (séparé en 3 étapes). On commence par le mode jeu libre. Sur l'écran du PC (et donc sur la table de billard grâce au projecteur) est affichée la fenêtre du fond sur laquelle on peut tracer et écrire des informations, elle est donc en permanence modifiée mais s'appuie sur le modèle de base de la figure 18. C'est une sorte d'interface graphique avec l'utilisateur. Dans le premier mode (appelé mode 0 dans la suite), on affiche la trajectoire de la balle en temps réel (elle s'estompe en 3 secondes). Sur la zone d'informations à droite, on affiche si la balle est en mouvement ou à l'arrêt. Sur la zone en dessous de la table, on indique à l'utilisateur les commandes disponibles, à savoir :

- Sauvegarder la dernière trajectoire de la bille en appuyant sur « s ». Il faut pour cela qu'elle soit à l'arrêt.
- Entrer dans le fonctionnement type rejou, en commençant par le mode 1, en appuyant sur « r ».
- Quitter l'application avec « q ».

Le mode 1 permet de choisir la trajectoire à rejouer. On affiche donc sur l'interface graphique les numéros de l'ensemble des coups enregistrés. Sélectionner un numéro affiche la trajectoire sélectionnée et le joueur peut valider son choix avec « entrée ». Il est possible de sauvegarder jusqu'à 9 coups. Lorsque le choix du joueur est fait, on entre directement dans le mode 2. Il est sinon possible de revenir au mode 0 avec « échap » et de quitter l'application avec « q ».

Le mode 2 attend de l'utilisateur qu'il replace la bille à sa position initiale. Pour cela, on affiche directement sur le billard la position où la bille doit être placée par un cercle. En parallèle, on tracke la position actuelle de la balle. Lorsque elle est bien placée, on affiche un deuxième cercle plus grand centré sur la même position. Au bout de 3 secondes, si la bille n'a pas été déplacée, on entre automatiquement dans le mode 3. Il est encore possible de revenir au mode 1 avec « échap » ou de quitter



(a) Affichage de la trajectoire pour permettre le choix du coup à rejouer



(b) Affichage des deux cercles lorsque la bille est correctement positionnée

FIGURE 20 – Modes de fonctionnement 1 et 2

Le mode 3 constitue la véritable phase de rejeu. La trajectoire enregistrée est affichée en pointillés. L'utilisateur peut alors essayer de refaire le même coup pour le travailler et comparer l'écart entre les deux trajectoires ou tenter un tout autre coup si celui qu'il avait fait ne le satisfaisait pas. La position de la bille est trackée pour afficher en temps réel sa nouvelle trajectoire en traits continus, sans qu'elle ne s'efface. Lorsque la bille s'arrête, le programme est en stand-by et attend la prochaine demande de l'utilisateur. Ce dernier peut :

- Etudier les deux trajectoires qui sont toujours affichées.
- Retourner au mode 0 s'il souhaite poursuivre le jeu. Il lui faut pour cela cliquer sur « 0 ».
- Choisir une nouvelle trajectoire à travailler, c'est à dire retourner au mode 1. Il doit donc appuyer sur « 1 ».
- Rejouer ce même coup, ce qui nécessite de replacer la bille à sa position initiale (mode 2). Cette commande est accessible par la touche « 2 ».
- Sauvegarder la nouvelle trajectoire dans la liste des trajectoires sauvegardées en appuyant sur « s ».
- Sauvegarder la nouvelle trajectoire en écrasant la précédente. Ceci est possible avec la commande « Maj + s ».

A la fin du programme les trajectoires sauvegardées ainsi que les paramètres potentiellement modifiés (taille du rectangle pour positionner la table et matrice d'homographie) sont enregistrés dans des fichiers annexes.

## 5 Conclusion

L'objectif de ce projet était de mettre au point un outil de tracking polyvalent et performant ainsi qu'une étude de modélisation, qui puissent représenter un réel intérêt pour les sportifs. Dans les deux disciplines étudiées, tennis de table et billard, les problématiques sont différentes mais notre outil a pu fournir des résultats satisfaisants.

Au billard, nous avons pu implémenter un algorithme qui suit en temps réel et de manière précise la trajectoire de la bille tout en affichant des données de jeu et propose une fonctionnalité de rejeu véritablement utile pour les joueurs.

Toutefois, on peut dégager plusieurs pistes d'améliorations. Par exemple, prévoir la trajectoire et le point d'arrêt de la bille en mesurant sa vitesse à partir des premières images de sa trajectoire (ou alors en traquant le mouvement de la canne pour indiquer au joueur le résultat de son coup avant même qu'il le joue). Cela nécessiterait une modélisation de la trajectoire de la bille qui, dans un premier temps, serait caractérisée par des frottements sur la table pour la trajectoire libre et un changement de direction et un coefficient de restitution de vitesse en partant de l'hypothèse de l'indéformabilité des solides.

Un autre aspect d'amélioration est bien sûr la possibilité de tracker plusieurs billes en même temps. Selon le type de bille : américaine (avec des numéros) ou classique (de couleur uniforme) l'algorithme de tracking sera confronté à des problèmes bien différents. En effet, d'un côté il est conçu pour détecter une balle de couleur uniforme (ce qui n'est pas le cas des billes de numéro supérieur à 8) et de l'autre il faudra pouvoir faire la distinction entre les billes de la même couleur. C'est notamment lors du choc entre deux billes qu'il faudra voir comment l'algorithme réagit et qu'il sera sans doute nécessaire de modéliser la trajectoire des billes après un choc pour indiquer à l'algorithme un moyen de les différencier. Il faudra également redéfinir la manière dont les trajectoires sont stockées et le placement des billes au début de la phase de rejeu.

Pour améliorer les performances temporelles de l'algorithme, il serait également intéressant d'utiliser les coordonnées sur l'image de la caméra des 4 coins déterminés lors du calcul de l'homographie. On pourrait ainsi réduire l'espace de recherche des billes sur l'image, permettant d'une part de réduire le temps de calcul et d'autre part de limiter les fausses détections dues à un objet de couleur similaire à la bille en dehors de la table.

Enfin, d'un point de vue davantage orienté vers l'utilisateur, il serait bon de pouvoir lui laisser la possibilité de régler certains paramètres comme les bornes de recherche de la bille dans l'espace HSV. En effet, ces dernières ont été spécifiquement calibrées pour le montage de la salle Amigo et avec une caméra précise, il n'est donc pas certain que les paramètres déterminés soient optimaux dans toutes les situations.

Une autre idée est de permettre à l'utilisateur de communiquer avec le programme sans avoir besoin de se déplacer jusqu'à l'ordinateur pour appuyer sur une touche. Les joueurs de billard ont en effet avec eux un objet cubique appelé le bleu. Ce dernier n'étant normalement jamais posé sur la table, on peut imaginer que sa détection sur cette dernière engendrerait le début de la communication avec l'utilisateur. Un quadrillage proposant différentes options à l'utilisateur serait alors affiché et le joueur devra alors positionner le bleu dans la case qui correspond à son choix pour que l'algorithme interprète la position de ce dernier comme un choix.

Au tennis de table, on a pu montrer l'efficacité de notre outil de tracking mais la reconstruction de la trajectoire en trois dimensions à partir des flux vidéo des deux caméras n'a pas pu être abordée. Il pourra alors être envisagé d'utiliser dans cette optique.

## 6 Remerciement

Nous voulons remercier les différents acteurs qui nous ont permis de mener ce projet à bien.

Tout d'abord, nous deux tuteurs Romain Vuillemot et Louis Gostiaux pour leurs retours au long du projet ainsi que le LIRIS pour l'accès à la salle Amigo et son matériel.

Nous remercions également notre conseiller de gestion de projet et en communication Arnaud Bréard pour ses suggestions dans la présentation de notre travail.

Enfin, nous remercions le Bazar de Centrale Lyon pour le prêt de l'échiquier qui a servi aux études de dé-distortion de l'image.

## Références

- [1] Patrick Wong, Laurence Dooley, and Adrian Hopgood. Tracking a table tennis ball for umpiring purposes. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 170–173, May 2015.
- [2] Roman Voeikov, Nikolay Falaleev, and Ruslan Baikulov. TTNet : Real-time temporal and spatial video analysis of table tennis. *arXiv :2004.09927 [cs]*, April 2020. arXiv : 2004.09927.
- [3] Development of a video-based position tracking system for automatic acquisition of the ball trajectories in caromarom billiard.
- [4] T. Jebara, C. Eyster, J. Weaver, T. Starner, and A. Pentland. Stochasticks : augmenting the billiards experience with probabilistic vision and wearable computers. In *Digest of Papers. First International Symposium on Wearable Computers*, pages 138–145, Cambridge, MA, USA, 1997. IEEE Comput. Soc.
- [5] Adrian Rosebrock. Ball tracking with opencv. 2015.
- [6] Rod Cross. Bounce of a spinning ball near normal incidence. *American Journal of Physics*, 73(10) :914–920, October 2005.