Analyse non intrusive du geste sportif dans des vidéos par apprentissage automatique

Jordan Calandre

La Rochelle Université

ÉCOLE DOCTORALE EUCLIDE

Laboratoire Mathématiques, Image et Applications

THÈSE présentée par :

Jordan CALANDRE

29 Septembre 2022

pour obtenir le grade de : **Docteur de La Rochelle Université**
Spécialité : **Automatique, Image et Signal**

Analyse non intrusive du geste sportif dans des vidéos par apprentissage automatique

---

Devant la commission d'examen composée de :

Chaabane DJERABA	Pr.	Université de Lille	Rapporteur
Philippe CARRE	Pr.	Université de Poitiers	Rapporteur
Jenny BENOIS-PINEAU	Pr.	Université de Bordeaux	Examinatrice
Catherine CHOQUET	Pr.	La Rochelle Université	Examinatrice
Laurent MASCARILLA	MCF-HDR	La Rochelle Université	Co-Directeur de thèse
Renaud PÉTERI	MCF-HDR	La Rochelle Université	Co-directeur de thèse
Julien MORLIER	Pr.	Université de Bordeaux	Invité

« 井の中の蛙大海を知らず (いのなかのかわず、たいかいをしらず) »

Une grenouille dans un puits ne connaît pas l'océan.

Table des matières

Table des figures

Liste des tableaux

List of Abbreviations

BoW	**B**ag of **W**ords
CNN	**C**onvolutional **N**eural **N**etwork
COR	**C**oefficient **O**f **R**estitution
CPU	**C**entral **P**rocessing **U**nit
CRISP	Compute**R** v**I**sion for **S**port **P**erformance
DVR	**D**igital **V**ideo **R**ecorder
FC	**F**ully **C**onnected
FMO	**F**ast **M**oving **O**bject
GPU	**G**raphics **P**rocessing **U**nit
HOF	**H**istogram of **O**riented Optical **F**low
HOG	**H**istogram of **O**riented **G**radients
IA	**I**ntelligence **A**rtificielle
ICP	**I**terative **C**losest **P**oint
IMU	**I**nertial **Me**asurement **U**nit
KCF	**K**ernelized **C**orrelation **F**ilter
KPI	**K**ey **P**erformance **I**ndicator
LIDAR	**L**ight **D**etection **a**nd **R**anging
OF	**O**ptical **F**low
RoI	**R**egions **o**f **I**nterest
TOF	**T**ime-**o**f-**F**light

Constantes Physiques

Coefficient de frottement de la raquette	$k = 1,9.10^{-3}$
COR de la raquette	$\epsilon_r = 0{,}81$
COR de la table	$\epsilon_t = 0{,}93$
Densité de l'air	$\rho = 1,225 kg/m^3$
Masse de la balle	$m = 27g$
Rayon de la balle	$r = 2cm$
Surface de frottement de la balle	$A = \pi * r^2$
Taille réelle de la balle	$H = 4cm$
Vecteur champ de pesanteur	$g = 9,8N/kg$

Liste des Symboles

(A_v, B_v)	Matrices de rebond table pour $\mathbf{V}^{in}$
(A_ω, B_ω)	Matrices de rebond table pour $\boldsymbol{\omega}_{in}$
(A''_v, B''_v)	Matrices de rebond raquette pour le $\mathbf{V}^{in''}$
(A''_ω, B''_ω)	Matrices de rebond raquette pour $\boldsymbol{\omega}_{in''}$
C_D	Coefficient de frottement
C_L	Coefficient de lift
(c_x, c_y)	Coordonnées en pixels du point focal
D	Distance entre la balle et la caméra
f	Distance Focale
f_{dev}	Distance en pixels entre le point focal et le centre de la balle
F_s	Fréquence d'acquisition
$\mathbf{F}_A$	Forces aérodynamiques
$\mathbf{F}_D$	Force de trainée
$\mathbf{F}_G$	Force grativationnelle
$\mathbf{F}_L$	Force de lift, ou Effet Magnus
h	Taille de la balle en pixels
$I = 2/3\ m.r^2$	Moment d'inertie
$\mathcal{I}_0$	Aller : Première impact, sur la raquette
$\mathcal{I}_1$	Aller : Deuxième impact, sur la table
$\mathcal{I}_2$	Aller : Troisième impact, sur la raquette adverse
$\mathcal{I}_3$	Retour : Quatrième impact, sur la table
$\mathcal{I}_4$	Retour : Cinquième impact, sur la raquette
$\mathbf{K}$	Matrice de paramètres intrinsèques
K_n	Coefficients de distorsions radiales d'une image
(k_x, k_y)	Coefficients d'échelle
$\mathbf{M}_{\mathcal{R},\mathcal{R}''}$	Matrice de changement de bases pour passer de $\mathcal{R}$ à $\mathcal{R}''$
$\mathbf{M}_{\mathcal{R}'',\mathcal{R}}$	Matrice de changement de bases pour passer de $\mathcal{R}''$ à $\mathcal{R}$
n	Nombre d'observations souhaitées pour ré-échantillonage temporel
n_e	Nombre d'échantillons entre $t = 0$ et $t = 1$
$\mathbf{P}$	Matrice de paramètres extrinsèque
$\mathbf{P}$	Quantité de mouvement

$\mathbf{P}_{in} = (x_{in}, y_{in}, z_{in})^t$	Position de l'origine du repère $\mathcal{R}''$ dans $\mathcal{R}$
P_n	Coefficients de distorsions tangentielles d'une image
(P_0, P_1)	Polynôme pour la régression de $\mathcal{S}_0$ et $\mathcal{S}_1$ dans $\mathcal{R}'$
$\mathbf{R}$	Matrice de rotation de la caméra
$\mathcal{R}$	Repère lié à la table
$\mathcal{R}'$	Repère lié au plan de frappe
$\mathcal{R}''$	Repère lié à la raquette
s_{xy}	Coefficient de déviation entre les axes x et y
$\mathcal{S}_.$	Portion de trajectoire quelconque
$\mathcal{S}_0$	Première portion de trajectoire $(\mathcal{I}_0, \mathcal{I}_1)$
$\mathcal{S}_1$	Deuxième portion de trajectoire $(\mathcal{I}_1, \mathcal{I}_2)$
$\mathcal{S}_2$	Troisième portion de trajectoire $(\mathcal{I}_2, \mathcal{I}_3)$
$\mathcal{S}_3$	Troisième portion de trajectoire $(\mathcal{I}_3, \mathcal{I}_4)$
S_0	Paramètre de lift
$t = 0$	Temps correspondant au point d'impact de la balle
$t = 1$	Temps correspondant au rebond table
T	Taille du cube temporel de notre réseau
$\mathbf{T}$	Vecteur de translation de la caméra
$[u, v, 1]$	Coordonnées de la projection d'un objet 3D dans l'image
$\mathbf{v_ø}$	Vélocité tangente à la table au moment de l'impact
v_s	Vitesse de glisse sur la table
$\mathbf{V}$	Vecteur vitesse de la balle dans $\mathcal{R}$
$\mathbf{V}'$	Vecteur vitesse de la balle dans $\mathcal{R}'$
$\mathbf{V_0}$	Vecteur initial de vitesse de translation dans $\mathcal{R}$
$\mathbf{V}^{in}$	Vecteur vitesse de la balle avant rebond dans $\mathcal{R}$
$\mathbf{V}_T^{in}$	Vitesse tangentielle à la table au moment de l'impact
$\mathbf{V}^{in''}$	Vecteur vitesse de la balle avant rebond dans $\mathcal{R}''$
$\mathbf{V}^{out}$	Vecteur vitesse de la balle après rebond dans $\mathcal{R}$
$\mathbf{V}^{out''}$	Vecteur vitesse de la balle après rebond dans $\mathcal{R}''$
$\mathbf{V}_r$	Vitesse de translation de la raquette dans $\mathcal{R}$
V_s	Décalage éventuel entre le point d'impact et le point de rebond
(x, y, z)	Axes de $\mathcal{R}$
(x', y', z')	Axes de $\mathcal{R}'$
(x'', y'', z'')	Axes de $\mathcal{R}''$
$(x_0, y_0, z_0)_{\mathcal{R}}$	Position initiale de la balle dans $\mathcal{R}$
(x_d, y_d)	Coordonnées d'un pixel dans l'image initiale avec déformations
(x_u, y_u)	Coordonnées d'un pixel dans l'image sans déformation
$\mathbf{W}$	Coordonnées homogènes 3D d'un objet dans la scéne

α	Orientation verticale de la raquette
γ	Orientation latérale de la raquette
δt	Pas temporel pour l'échantillonnage
Δt	Écart temporel, supposé petit pour la méthode d'Euler
$\Delta x'$	Décalage sur l'axe x'
$\Delta z'$	Décalage sur l'axe z'
$\boldsymbol{\theta} = (\alpha, 0, \gamma)$	Angle polaire du vecteur de translation de la balle
ω	Vecteur de vitesse rotation angulaire
μ	Coefficient de friction sur la table
$\boldsymbol{\omega_0}$	Vitesse de rotation supposée constante sur $\mathcal{S}$.
$\boldsymbol{\omega}_{in}$	Vecteur de rotation de la balle avant impact dans $\mathcal{R}$
$\boldsymbol{\omega}_{in''}$	Vecteur de rotation de la balle avant impact dans $\mathcal{R}''$
$\boldsymbol{\omega}_{out}$	Vecteur de rotation de la balle après impact dans $\mathcal{R}$
$\boldsymbol{\omega}_{out''}$	Vecteur de rotation de la balle après impact dans $\mathcal{R}''$

Chapitre 1

Introduction Générale

Sommaire

Dans la première partie de ce chapitre, nous commençons par présenter le contexte général de ces travaux, à savoir un intérêt croissant, dans la population générale comme dans les clubs et associations sportives, pour l'aide automatisée à l'activité sportive. La mise à disposition, pour le plus grand nombre, de données vidéo et de puissance de calcul, rend réalisable des logiciels d'intelligence artificielle pour analyser les pratiques sportives, en dehors des laboratoires, en condition d'entraînement ou de compétition. Le projet régional CRISP (ComputeR vIsion for Sport Performance), se place dans cette perspective et sert de cadre à notre travail.

Dans la deuxième partie, nous présentons le sujet principal de cette thèse : l'analyse de mouvements humains à grain fin dans des vidéos acquises avec un matériel compatible avec la pratique sportive, que nous qualifions de « non-intrusif». Cette notion, comme celle de granularité dans l'analyse vidéo, sera précisée.

Enfin, nous présentons dans la troisième partie nos différentes contributions dans ce domaine, qui seront détaillées tout au long de ce manuscrit.

1.1 Contexte

Le sport est un enjeu social et de santé publique, reconnu au niveau européen [1], très important pour la société, et il existe actuellement la volonté politique de développer pour tous les pratiques physiques. Un rapport [2] de l'INSERM a été effectué pour le ministère de la Ville, de la Jeunesse et des Sports afin d'évaluer l'impact de l'activité physique sur la santé. Ce rapport met l'accent sur le fait que pour quantifier réellement l'impact d'une activité sportive sur la santé, il est nécessaire de disposer d'outils pertinents de mesure des niveaux de cette activité.

La démocratisation actuelle de l'accès aux nouvelles technologies permettant la collecte de données individuelles relatives à l'activité physique (applications pour smartphones, montres et vêtements connectés...) se doit d'être considérée comme une source d'information potentielle pour la recherche et l'innovation dans les domaines de la santé et du sport.

L'analyse du geste sportif en laboratoire permet de combiner des mesures cinématiques 3D avec des mesures d'efforts extérieurs et des mesures d'activités musculaires. Il est possible *via* des marqueurs disposés à des endroits pertinents sur le corps du sportif, généralement les articulations, de reconstruire un squelette. L'analyse dynamique de ces traceurs (cellules photosensibles, marqueurs actifs) permet d'effectuer des analyses biomécaniques du sportif pendant l'effort. Une technique alternative est celle des « exosquelettes » de dernières générations qui permettent, en plus de leur fonctionnalité d'assistance à l'effort, d'obtenir des mesures dynamiques du corps (hanche, genou, pied) avec une grande précision.

Cependant, cette analyse du geste sportif est souvent cantonnée à des études en laboratoire et les données obtenues sont en général précises, mais souvent déconnectées d'une réalité évidente qui est celle du terrain. Les pratiques physiques de pleine nature, contexte dit écologique (dispositifs sportifs – salle de sport, parcours de santé – ou non) rendent naturelles des solutions d'acquisition sans marqueurs ou appareillages pouvant gêner le sportif dans sa performance et sa pratique. Le domaine de la vision par ordinateur, branche de l'intelligence artificielle, peut permettre de répondre à cette problématique en utilisant l'information provenant d'images ou de vidéos.

La vidéo est souvent utilisée par les entraîneurs et les sportifs pour analyser certaines séquences de jeux ou certains gestes techniques. Il existe des

1. https://rm.coe.int/12rev3-draft-5-fr-charte-europeenne-sport-revisee-\2021-epas-master-276/1680a3c545
2. http://www.sports.gouv.fr/IMG/pdf/1-inserm.pdf

logiciels d'analyse vidéo permettant d'aider au séquençage temporel d'actions (par exemple au rugby : les touches, les mêlées...) mais les traitements avancés, comme la reconnaissance de ces actions ou l'analyse de tactiques de jeux, restent faits manuellement par l'utilisateur.

Projet CRISP

Le projet CRISP (ComputeR vIsion for Sport Performance) était un projet multidisciplinaire et inter-établissements, financé, de janvier 2017 à novembre 2021, par la région Nouvelle-Aquitaine. Il a réuni des équipes spécialisées en vision par ordinateur et en traitement du signal et des images (le MIA et le L3i de pour La Rochelle Université ; le XLIM-ICONES de l'Université de Poitiers), en analyse et indexation de vidéos (le LABRI de l'Université de Bordeaux). Du côté applicatif, l'IMS (Université de Bordeaux), a apporté ses compétences notamment en acquisition et évaluation de la performance sportive, en analyse du mouvement ou en biomécanique humaine. Enfin, l'intégration du STAPS Bordeaux dans le projet a permis de solliciter les étudiants spécialistes des activités physiques et sportives dont des sportifs de haut niveau. En particulier, les séquences de tennis de table pour le chapitre 5 ont été acquises dans ce cadre.

L'objectif initial du projet CRISP était de créer une synergie forte entre les équipes de la Nouvelle Aquitaine autour de la Vision par Ordinateur et de ses applications dans un contexte Sport & Santé. Il avait pour finalité de rendre les caméras « intelligentes » afin d'analyser en situation dite « écologique » la pratique sportive. Dans ce contexte, le terme « écologique » est un équivalent de «non-intrusif » et qualifie les méthodes d'acquisition et d'analyse qui ne perturbe pas les sportifs. La vision a naturellement été notre choix pour respecter ces contraintes, c'est en effet le premier indicateur de l'entraîneur afin de qualifier l'efficience du geste sportif. L'objectif était donc de développer des méthodes en vision par ordinateur pour permettre :

1. l'acquisition de gestes sportifs,
2. leur reconnaissance,
3. leur analyse. Le but étant ainsi d'optimiser l'apprentissage et l'entraînement d'étudiants en faculté des sports ou en clubs.
4. L'analyse de tactiques de jeux. Ce point plus prospectif, doit permettre, si l'on considère un sport d'opposition (tennis de table, rugby...), d'analyser automatiquement les actions qui émergent lors d'une partie.

C'est naturellement dans le point 3) que notre travail trouve toute sa place. En effet, l'application type de l'étude du projet CRISP était le tennis de table. Pour ce sport, la région de Nouvelle-Aquitaine est un centre d'excellence : plusieurs clubs évoluent au niveau national (Villeneuve sur Lot, Cestas, Agen, Mont de Marsan, CAM Bordeaux) et possèdent en leur sein des joueurs de niveau international. L'équipe universitaire de tennis de table féminine de l'Université de Bordeaux participe régulièrement aux championnats d'Europe. La faculté des STAPS du collège Sciences de l'Homme de l'Université de Bordeaux joue un rôle majeur dans l'animation de cette équipe et a initié depuis de nombreuses années des actions de recherche (colloque ITTF INSEP 2013) et d'innovation pédagogique (application ScolpingTAB et colloque EducPing). Le projet CRISP bénéficie également du soutien de la Ligue de Tennis de Table de Nouvelle-Aquitaine.

1.2 Analyse d'actions sportives

La reconnaissance et l'analyse automatique d'actions et d'activités humaines dans des vidéos a reçu ces dernières années une attention particulière dans la communauté de la vision par ordinateur et de la reconnaissance de formes. Cet intérêt est motivé par une grande variété d'applications telles que l'annotation automatique de vidéos (Stein et al., 2018), la vidéo-surveillance (Arunnehru, Chamundeeswari et Bharathi, 2018), ou encore l'assistance aux personnes âgées.

Certains travaux ont porté sur la reconnaissance dans des vidéos de sports (Laptev et al., 2008 ; Tang, Fei-Fei et Koller, 2012 ; W. Li et al., 2013), avec pour objectif de reconnaître un sport parmi plusieurs et non de reconnaître et de caractériser finement une action d'un sport donné comme c'est notre cas.

Ces approches ont été testées sur de nombreux jeux de données standards comme UCF-101 (Soomro, Zamir et Shah, 2012) ou AVA (Gu et al., 2018) qui sont dédiés à la reconnaissance d'actions humaine dans des vidéos. Toutefois, leur objectif principal est de faire le lien entre une séquence vidéo et une activité au sens large. Elles peuvent être qualifiées de méthodes à gros-grain.

Les paramètres considérés dans ces méthodes comme le nombre de joueurs présents, ou encore les couleurs et textures présentes dans la scène ne sont d'aucune utilité pour une analyse précise du geste sportif que nous qualifierons d'analyse "à grain fin".

Le problème est également plus complexe puisque la variabilité intra-classe (vidéos d'un même coup) peut-être très forte lors d'un changement de point de vue, et la variabilité inter-classe (vidéos de coups différents) peut être très faible. Deux coups différents d'un même joueur peuvent avoir plus de similarités visuellement que le même coup effectué par deux joueurs différents. De plus, la quantité d'annotations nécessaire s'avère relativement importante, avec un coût en temps humain élevé et nécessitent des connaissances parfois expertes pour lever les ambiguïtés.

Analyse à grain fin

L'extraction de mouvement humain, ou Motion Capture (MoCap), est souvent utilisée dans les jeux-vidéos et dans la création d'animations cinématographiques (XIA et al., 2017). Les deux familles de systèmes les plus populaires pour cette capture de mouvements humains sont ceux qui utilisent des dispositifs optiques, le plus souvent des caméras couleurs et les autres dits-« non-optiques», qui utilisent des capteurs inertiels ou mécaniques.

Acquisition par systèmes optiques

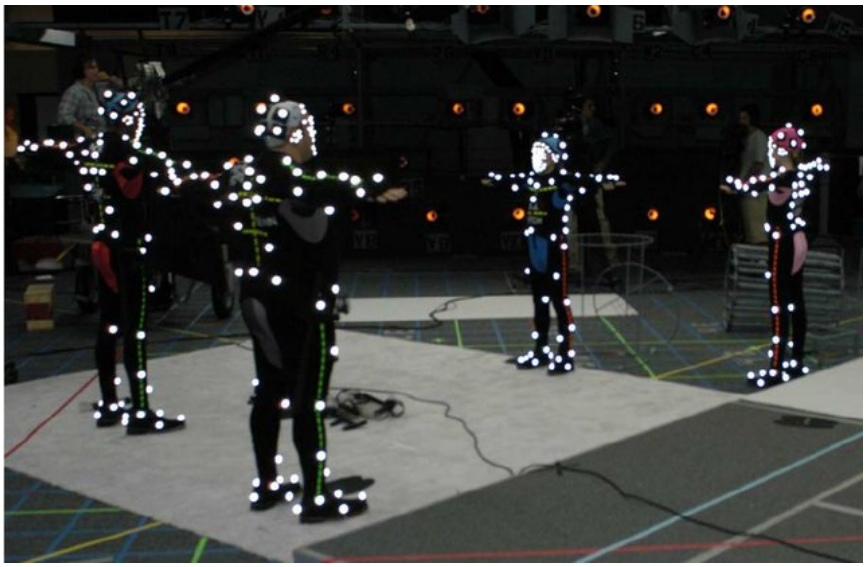


FIGURE 1.1 – Acteurs avec combinaison pour la capture de mouvements (HAVALDAR, 2006)

Les systèmes optiques utilisent des données images issues d'un ensemble de caméras synchronisées pour obtenir par triangulation la position 3D d'une

ou plusieurs personnes. On ne se limite pas dans ce contexte à repérer la personne par son centre de gravité mais c'est un ensemble de points remarquables, choisis selon l'application qui sera traitée. Classiquement, ce sont les membres, le torse, la tête, voire des éléments du visage qui sont retenus. Le plus souvent, ces éléments sont repérés par des marqueurs réfléchissants dits marqueurs `passifs` placés à des positions remarquables comme les articulations ou les lignes du visage. Ces marqueurs sont ensuite détectés sur chaque image, et mis en correspondance entre au moins deux images pour effectuer une triangulation et obtenir leurs coordonnées spatiales dans la scène 3D. Un exemple d'un ensemble de marqueurs de ce type est présenté sur la figure 1.1. La surface capturée est cependant restreinte à la vue commune d'au moins deux des caméras. Afin d'étendre la surface pouvant être reconstruite, il est nécessaire d'augmenter le nombre de caméras, ce qui alourdit la mise en place du système et augmente son coût.

Ces marqueurs `passifs` ont ensuite évolué vers des marqueurs dits-`actifs` équipés de leds. Ceux-ci émettent souvent dans l'infrarouge et par un système de codage sont identifiables de façon unique. Par exemple, les leds s'éclairent très brièvement et à tour de rôle, ce qui permet de les apparier sans ambiguïté. Si la précision de ces systèmes est très élevée, mais les contraintes pratiques sont nombreuses, ce qui réserve ce type de techniques à des utilisations ponctuelles comme le tournage d'un film ou l'analyse fine d'un geste en condition de laboratoire.

Acquisition par systèmes non optiques

Pour les systèmes dits non-optiques, l'obtention des positions des différents marqueurs ne repose pas sur l'exploitation de données image, mais sur l'utilisation de capteurs récupérant des paramètres physiques. Les deux principaux systèmes non-optiques utilisent des capteurs de mouvement inertiels ou des capteurs de mouvement mécaniques.

Les capteurs de mouvements inertiels (IMU), contiennent une combinaison de gyroscope, de magnétomètre et d'accéléromètre. Ainsi, les mouvements et rotations des différents capteurs sont détectés, puis transmis à un ordinateur comme le montre la Figure 1.2.

Le deuxième type de système non-optique est le mouvement mécanique, ou capture par exosquelette. L'objectif est de suivre directement les mouvements relatifs de la structure fixée sur le corps. Un exemple est donné à la Figure 1.3.

FIGURE 1.2 – IMUs pour extraire les informations de mouvement et de rotation (BLESER et al., 2017).

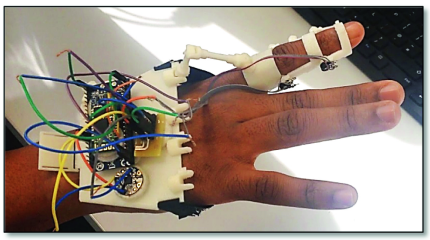


FIGURE 1.3 – Suivi de la main avec mouvement mécanique (SECCO et TADESSE, 2020)

Dans les deux cas présentés ci-dessus, l'analyse est faite dans des conditions loin d'être adaptée à une utilisation sportive de haut niveau. Que ce soit pour un système optique ou non-optique, l'utilisation de capteurs fixés sur le corps peut gêner les sportifs lors de leurs déplacements, lors de l'exécution des gestes techniques et réduire leurs performances. À cette gène s'ajoute le coût généralement élevé des dispositifs acquisition, conjuguée à des contraintes d'acquisitions fortes nécessitant des locaux spécialisés. Cette approche est donc peu adaptée à un entraînement quotidien.

L'approche non-intrusive, comme celle que nous avons développée, semble donc la plus naturelle et la mieux adaptée dans ce contexte.

1.3 Contributions

Les contributions sont divisées en deux parties qui suivent le plan du manuscrit. Pour la première partie, les contributions sont les suivantes :

— Acquisition vidéos de séquences de jeu effectuées par des joueurs de niveau régional et national. La captation a été faite en stéréoscopie grâce à deux caméras rapides (240 fps), calibrées et synchronisées ce qui a permis de créer le jeu de données `avec sportifs`. Trois types de coups sont identifiés dans cette base : les `Top Spins`, les `Contre-Attaques`, et les `Poussettes`.

— Création de deux jeux de données synthétiques de trajectoires de balles physiquement réalistes. Chaque séquence du premier jeu correspond à une trajectoire entre deux frappes avec un seul rebond sur la table. Le second jeu y ajoute la frappe par une raquette : il comprend donc une trajectoire avec un rebond sur la table, une frappe et une seconde trajectoire avec un rebond.
Ces jeux de données, nommés respectivement `Synthétique` et `Synthétique avec frappe`, sont constitués de trajectoires simulées dont nous connaissons les paramètres cinématiques et qui permettent d'évaluer nos méthodes.

— Proposition d'une méthode de reconstruction des positions 3D successives de balles par une approche monocaméra. La reconstruction 3D des positions de la balle est possible en utilisant un réseau convolutif entraîné sur des images synthétiques, puis ré-entraîné sur des séquences réelles dont la vérité-terrain est connue grâce par stéréovision.

— Proposition d'une méthode de calcul des paramètres cinématiques de la balle : vitesse de translation et de rotation. À partir des positions successives de la balle, cette extraction est faite en utilisant des modèles physiquement réalistes de trajectoire et de rebond sur la table.

— Proposition d'une méthode de calcul des paramètres de la raquette : vitesse et angle de frappe. Là encore, un modèle physiquement réaliste du comportement de la balle lors de la frappe est utilisé.

L'essentiel de cette thèse s'étant déroulée durant la pandémie de Covid-19, et notamment durant les périodes de confinement, les phases d'acquisition des vidéos en condition réelles et de constitution des bases de données a été compliquée à mener à bien. Notamment, les complexes sportifs ont longtemps été inaccessibles. C'est une des raisons qui nous ont poussés à créer et

à utiliser largement des jeux de données de synthèse.

Pour la seconde partie, les principales contributions de ce manuscrit correspondent à des travaux menés dans le cadre du workshop MediaEval [3] :

— Organisation d'une tâche de classification d'actions humaines au défi MediaEval avec l'équipe des membres du projet CRISP.

— Participation à ce défi en utilisant les singularités du flot optique et des Images Dynamiques pour encoder des séquences vidéos comme entrée d'un réseau convolutif.

3. https://multimediaeval.github.io/

1.4 Structure du manuscrit

Ce manuscrit est composé de deux parties ainsi que d'annexes.

Dans le présent chapitre 1, nous avons abordé les motivations et les problèmes inhérents à l'analyse fine de mouvements sportifs, ainsi que les raisons pour lesquelles une approche non-intrusive est à privilégier.

Dans la première partie de ce manuscrit, nous nous focalisons sur l'étude des trajectoires en utilisant un modèle physique.

Nous commençons par aborder, dans le chapitre 2, le problème de l'extraction en vision mono-caméra des positions 3D des centres des balles, ainsi que des trajectoires qu'elles constituent. Cette reconstruction en 3D, à partir d'images 2D, est rendue possible grâce à la calibration de la caméra et à l'existence d'objets dont la géométrie est parfaitement standardisée : la table et balle. La qualité de la reconstruction est validée grâce aux paires stéréoscopiques du jeu de données `avec sportifs`.

Dans le chapitre 3, nous présenterons différentes méthodes de représentation et d'analyse de la trajectoire d'une balle de tennis de table depuis la frappe du joueur jusqu'au premier rebond sur la table. À partir des informations obtenues au chapitre précédent et grâce à la connaissance d'un modèle physique qui prend en compte les forces s'exerçant sur la balle lors de son déplacement, nous en déduisons ses paramètres cinématiques. Des expérimentations sur nos jeux de données valident l'approche.

Le chapitre 4 complète notre étude en introduisant la modélisation des rebonds sur la table puis sur la raquette. Dans la section 4.2, l'utilisation des rebonds de la balle sur la table permet d'améliorer les estimations des vitesses de rotation obtenues au chapitre précédent. La section 4.3 est dédiée à l'étude des paramètres vitesse et orientation caractérisant le comportement de la raquette lors d'une frappe. Ces paramètres sont difficiles à obtenir à partir de l'observation de la raquette sur une vidéo : la forme de la raquette n'est pas standardisée, et n'est pas toujours visible. Nous avons donc choisi de ne pas l'observer directement. En effet, son impact sur la balle entraîne une modification des paramètres de vitesse de cette dernière, leur connaissance avant et après impact permet de déduire vitesse et orientation de la raquette. Cette information est d'autant plus intéressante qu'elle fait le lien avec le geste du joueur et constitue un bon indicateur de performance.

La seconde partie de ce manuscrit porte sur la reconnaissance à grain fin d'actions sportives. Dans le chapitre 5 nous présentons brièvement nos travaux en tant que participant comme en tant que membre de l'équipe organisatrice de la tâche de classification d'actions sportives au workshop MediaEval.

À la suite de ces deux parties, la conclusion et les perspectives de nos travaux sont données dans le chapitre 6.

Première partie

Partie 1 - Analyse des forces cinématiques

Chapitre 2

Reconstruction de séquence de positions dans l'espace 3D à partir d'une vidéo

Sommaire

Introduction L'objectif de ce chapitre est de reconstruire la position 3D d'une balle à partir d'une séquence vidéo obtenue par une unique caméra. La qualité de cette reconstruction est évaluée de deux façons. La première repose sur l'utilisation d'une seconde caméra qui, synchronisée avec la première, fournit une séquence de couples stéréoscopiques, et nous permet par la suite d'obtenir des estimations précises des positions 3D de la balle. La seconde utilise des séquences vidéo de synthèse.
Dans ce manuscrit, nous distinguerons la *séquence de positions 3D* d'une balle de sa *trajectoire*, qui intègre un modèle physique de la dynamique du mouvement, et qui sera abordé dans le chapitre 3.

FIGURE 2.1 – Étapes pour la reconstruction de séquence de positions 3D d'une balle

Les principales contributions de ce chapitre sont :
— La mise en place d'un processus d'acquisition pour l'enregistrement de scènes de tennis de table ayant pour objectif une étude trajectographique de la balle.
— La création d'un jeu de données `synthétiques` reproduisant la scène réelle (position de caméras, paramètres de caméra et flou de mouvement de la balle).
— La mise en place d'un réseau convolutif qui aide à la reconstruction de la séquence de position 3D d'une balle. L'apprentissage est fait sur un jeu de données `synthétiques` en vision monoculaire.
— L'utilisation de la technique de l'apprentissage par transfert pour adapter ce réseau à un jeu de données `avec sportifs`, et reconstruire la séquence de positions 3D grâce à une seule caméra.

Dans la section 2.1 de ce chapitre, nous présentons les différents travaux associés à cette reconstruction. Le processus global est illustré par la figure 2.1. La première étape indispensable est la calibration de la caméra pour obtenir les paramètres internes, liés au dispositif optique (matrice intrinsèque)

et externes, liés à la position de la caméra par rapport à la table de tennis de table (matrice extrinsèque). Après cela, nous faisons un bref état de l'art sur la détection et le suivi de balle. La fin de cette partie est dédiée à la présentation des approches de reconstruction 3D à l'aide de caméra mono- et stéréo-vision.

Nous présentons dans la section 2.2 les deux jeux de données dédiés que nous avons constitués. Dans les deux cas, il s'agit de séquences de couples stéréoscopiques. Le premier jeu contient des vidéos de sportifs acquises en conditions réelles d'entraînement (voir section 2.2.1), et le second des données synthétiques reproduisant des conditions similaires (voir section 2.2.2).

Enfin, la partie 2.3 présente notre approche pour la reconstruction de la séquence de positions 3D. Après la calibration et le suivi de balle au cours de la séquence vidéo, mono-caméra, nous estimons la taille de la balle en pixels grâce à un réseau convolutif et nous exploitons cette information pour en déduire la distance par rapport à la caméra, et reconstruire la séquence de positions 3D dans la scène.

Nous concluons ce chapitre en section 2.5.

2.1 Reconstruction 3D à partir d'une vidéo

2.1.1 Calibration de caméra

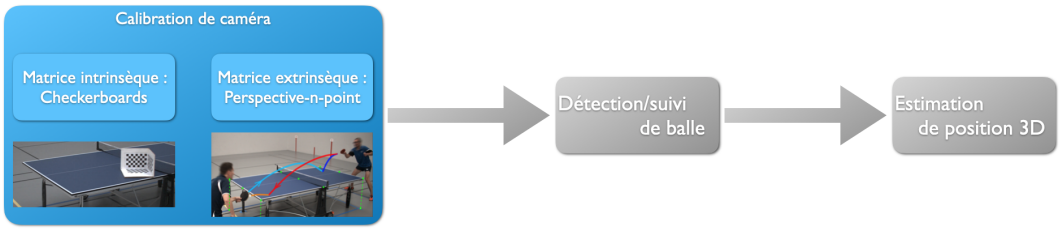


FIGURE 2.2 – Première étape : obtention des matrices intrinsèques et extrinsèques

L'objectif de ce chapitre étant d'estimer des positions 3D à partir d'images 2D, il est indispensable de "modéliser le processus de formation des images, c'est-à-dire trouver la relation entre les coordonnées spatiales d'un point de l'espace avec le point associé dans l'image prise par la caméra" (d'après WIKIPEDIA, 2021). Ce processus est connu sous le nom de calibrage, ou calibration de caméra, il est résumé à la figure 2.2.

Le dispositif optique d'acquisition d'image le plus simple que l'on puisse imager est l'appareil à sténopé. Un appareil à sténopé correspond à une boite fermée dont une face présente un trou minuscule qui laisse entrer la lumière. Les rayons lumineux traversent ce trou et viennent se refléter sur la face opposée à la face contenant ce trou, appelée plan image. L'image observée est donc inversée. Comme l'illustre la figure 2.3, l'image peut être conservée lorsque la face opposée au trou contient un support photosensible, historiquement du papier photographique, aujourd'hui un capteur électronique.

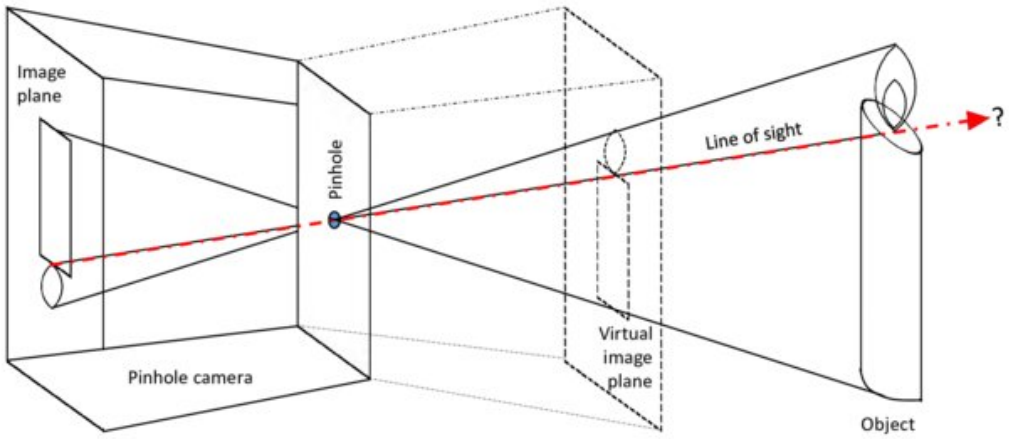


FIGURE 2.3 – Illustration du modèle de sténopé (CHATTERJEE, 2016)

Ce dispositif simplifié, bien qu'il ne soit plus utilisé aujourd'hui, est à l'origine du modèle mathématique qui décrit la relation entre les coordonnées d'un point dans l'espace tridimensionnel et sa projection sur le plan image d'un appareil.

Les paramètres d'une caméra sont séparés en deux catégories, qui sont les paramètres intrinsèques, liés directement aux caractéristiques internes de la caméra, et extrinsèques, liés au positionnement de la caméra dans son environnement. Dans notre cas, il s'agit de la position de la caméra par rapport à la table.

Ces paramètres intrinsèques sont obtenus à l'aide de mires de calibration (HARTLEY et ZISSERMAN, 2003). Différents types de mires existent (HA et al., 2017), et leur usage dépend de la situation et des contraintes d'acquisition. Néanmoins, le principe est similaire pour tous les types de mires. Celles-ci sont constituées de motifs à la géométrie connue, et à fort contraste (en noir et blanc ou colorées), typiquement un damier.

Nous commençons par décrire la matrice $\mathbf{K}$ des paramètres intrinsèques d'une caméra (WENG, COHER et HERNIOU, 1992) (voir équation 2.1). Elle

contient les informations nécessaires à la projection grâce au modèle de sténopé, mais également la conversion des coordonnées images (en unité métrique) en coordonnées images discrètes (pixels) [1]. Ces informations sont :

— la distance focale f,
— les coordonnées en pixels de l'intersection de l'axe optique avec le plan image c_x et c_y (correspondant généralement au centre de l'image). Ce point est appelé point principal.
— les coefficients d'échelle k_x et k_y ($k_x = k_y$ lorsque les pixels sont carrés),
— Un coefficient de déviation entre les axes x et y nommé s_{xy} (zéro dans le cas où les axes sont orthogonaux).

Remarquons, dès à présent, que nous travaillerons toujours en coordonnées homogènes par la suite, et $\mathbf{K}$ est définie de la façon suivante :

$$\mathbf{K} = \begin{bmatrix} k_x & s_{xy} & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

Dans notre cas, comme dans la plupart des cas réels, le modèle de sténopé doit être enrichi pour prendre en compte l'usage de lentilles qui provoquent des distorsions de l'image (voir figure 2.4). Pour cela, un modèle de déformation, est introduit pour rectifier l'image et se ramener au modèle de sténopé. Cela conduit à introduire un certain nombre de paramètres, dits de distorsions, supplémentaires dans le modèle de caméra.

Soit (x_d, y_d) les coordonnées d'un pixel de l'image initiale, (x_u, y_u) les coordonnées du même pixel sur l'image rectifiée, c'est-à-dire tel qu'il aurait été obtenu par une caméra à sténopé idéale. En notant (x_c, y_c) le centre distorsion et K_n et P_n les $n^{\text{ièmes}}$ coefficients de distorsions radiales et tangentielles ainsi que $r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$ la distance d'un pixel de l'image avant rectification et le centre de distorsion, la relation entre les coordonnées des pixels de l'image initiale et rectifiée est la suivante :

$$\begin{cases} x_u = x_d + (x_d - x_c)(K_1 r^2 + K_2 r^4 + \cdots) + (P_1(r^2 + 2(x_d - x_c)^2) \\ \qquad + 2P_2(x_d - x_c)(y_d - y_c))(1 + P_3 r^2 + P_4 r^4 \cdots) \\ y_u = y_d + (y_d - y_c)(K_1 r^2 + K_2 r^4 + \cdots) + (2P_1(x_d - x_c)(y_d - y_c) \\ \qquad + P_2(r^2 + 2(y_d - y_c)^2))(1 + P_3 r^2 + P_4 r^4 \cdots), \end{cases} \tag{2.2}$$

1. La matrice des paramètres intrinsèques est de manière classique déterminée à un facteur d'échelle près.

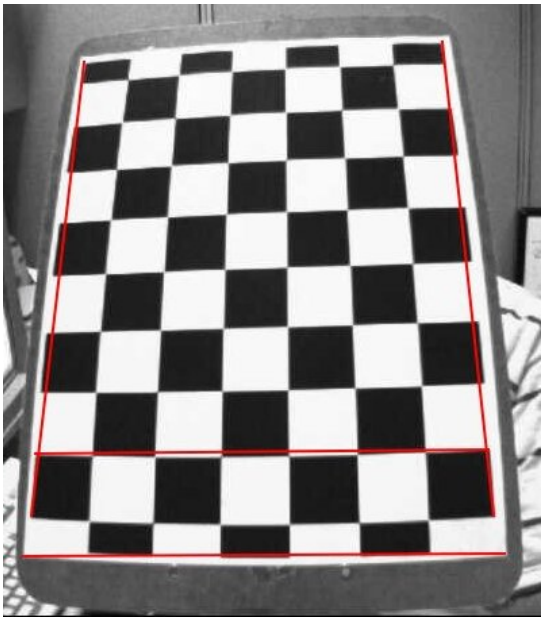


FIGURE 2.4 – Distorsions des lignes d'une mire de calibration (HALLERT., 1960)

Après détection par un algorithme *ad-hoc* des motifs des mires, leurs déformations peuvent être déterminées avec une très grande précision (PLACHT et al., 2014), ce qui permet de déduire les paramètres de distorsion puis, après rectification, les autres paramètres intrinsèques en utilisant le modèle de sténopé.

La mise en œuvre de ces mires imposent des contraintes pratiques. Tout d'abord, elles doivent être placées à hauteur d'utilisation, dans notre cas au niveau de la table. Il est également souhaitable d'acquérir plusieurs images dans lesquelles la mire est placée à différents endroits de la scène (centre et bordure de la table, sol) tout en restant visibles. Une alternative est d'utiliser une seule image contenant plusieurs mires (GEIGER et al., 2012) (voir figure 2.5). Il est important de remarquer que cette étape de la calibration peut être faite une fois pour toutes si la distance focale ne change pas, c'est-à-dire si le zoom optique n'est pas modifié.

Il reste ensuite à estimer les paramètres extrinsèques qui dépendent d'une matrice de rotation $\mathbf{R}$ et d'un vecteur de translation $\mathbf{T}$:

— la matrice de rotation

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix}$$

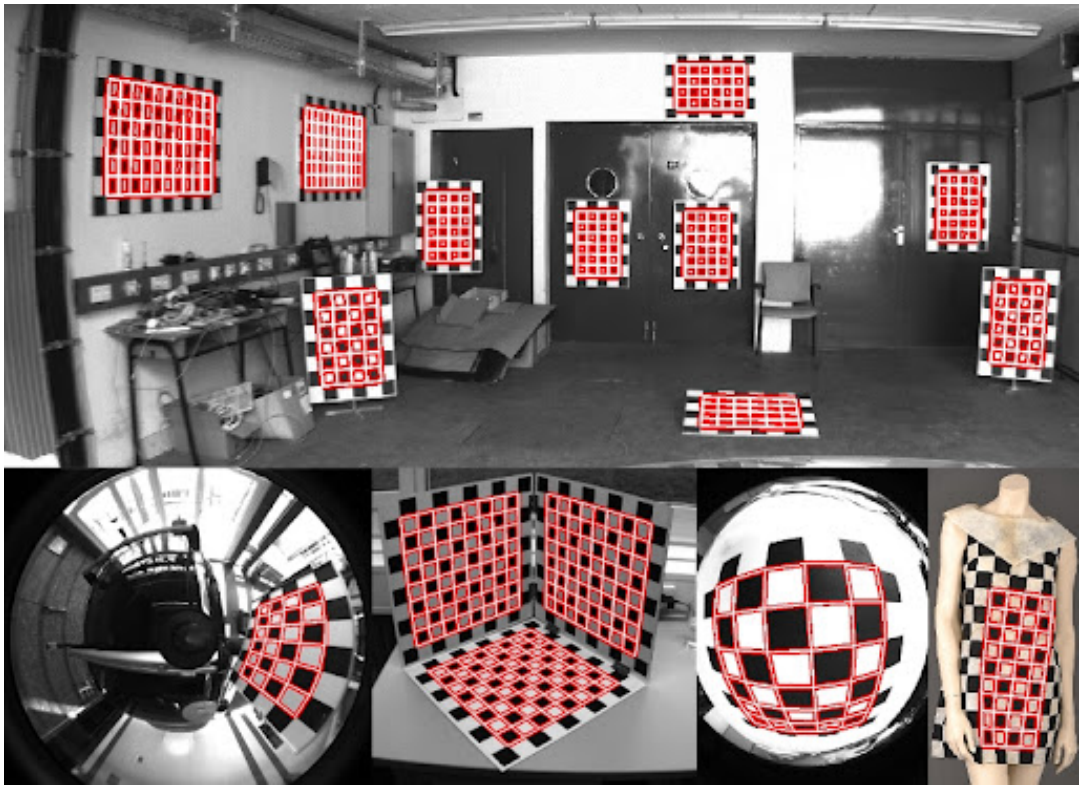


FIGURE 2.5 – Illustration de scène composée de plusieurs mires de calibration (trièdres) (GEIGER et al., 2012)

— le vecteur de translation de la caméra par rapport à la scène

$$\mathbf{T} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Posons $\mathbf{W} = [X, Y, Z, 1]^t$, les coordonnées homogènes 3D d'un point de la scène, et $[u, v, 1]^t$ les coordonnées, exprimées en pixels, de son projeté sur l'image. La matrice extrinsèque est alors définie comme $\mathbf{P} = \left[\mathbf{R}|\mathbf{T}\right]$, et on a les égalités suivantes :

$$\begin{aligned} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &= \begin{bmatrix} k_x & s_{xy} & c_x \\ 0 & k_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & r_{31} & t_1 \\ r_{12} & r_{22} & r_{32} & t_2 \\ r_{13} & r_{23} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \\ &= \mathbf{K} \left[\mathbf{R}|\mathbf{T}\right] \mathbf{W} \\ &= \mathbf{K}.\mathbf{P}.\mathbf{W} \end{aligned} \tag{2.3}$$

On notera par la suite $\mathbf{P^i}$ la $i^{\text{ème}}$ ligne de cette matrice $\mathbf{P}$ dans l'équation 2.4.

Lors d'acquisitions à l'aide de plusieurs caméras, chaque caméra possède sa propre matrice extrinsèque dépendant de sa position dans la scène.

Afin de l'obtenir, et pouvoir passer de l'espace monde à l'espace caméra, ou au repère table (l'origine étant située au centre de la table), il est nécessaire d'avoir des points de correspondance entre les deux repères. Pour ce faire, une annotation manuelle de points est généralement effectuée sur les images dont nous avons une connaissance préalable (SHAPIRO, 1978).

Après obtention de ces deux matrices, **K** et **P**, l'étape suivante dans la chaîne de traitement nécessaire pour la reconstruction 3D est la détection et le suivi de la balle dans le domaine image.

2.1.2 Détection et suivi de balle en mouvement

FIGURE 2.6 – Deuxième étape : détection de la balle sur chaque image

Après avoir déterminé les paramètres de calibration de la caméra, l'étape suivante illustrée figure 2.6 dans la chaîne de traitement est la détection de l'objet que nous souhaitons suivre sur les images. Dans notre cas, il s'agit de la balle de tennis de table. Comme nous le verrons, sa taille estimée en pixels sur chacune des images permettra ensuite sa rétroprojection dans l'espace 3D.

Dans le cas du tennis, la balle se déplace très rapidement et peut atteindre jusqu'à 30 m/s. Les objets qui, entre deux acquisitions, se déplacent sur une distance supérieure à leur taille apparente sont qualifiés de *Fast Moving Object*(FMO) (ROZUMNYI et al., 2017). Ce type de mouvement est actuellement un sujet d'étude très actif et définit un cadre d'étude exploité par les détecteurs que nous utiliserons par la suite.

Dans une séquence vidéo, les FMO sont les objets qui se déplacent sur une grande distance par rapport à leur taille pendant le temps d'acquisition d'une seule image. Ils peuvent tourner autour d'un axe arbitraire avec une vitesse angulaire inconnue. La définition la plus simple donnée par ROZUMNYI et al., 2017 suppose qu'un unique objet F se déplace sur un arrière-plan statique B. L'extension à des objets multiples, qui a également était proposée

par les mêmes auteurs, n'est pas présentée ici car elle sort du cadre de notre étude. La définition d'un FMO est la suivante. Soit une vidéo composée d'une séquence d'images $I_1(x), ...I_n(x)$, où $x \in \mathbb{R}^2$ est une coordonnée image exprimée en pixel. Une image I_t, à l'instant t, s'écrit :

$$I_t(x) = (1 - [\mathcal{H}_t M](x))\, B(x) + [\mathcal{H}_t F](x)$$

où M est la fonction indicatrice de F. En général, l'opérateur $\mathcal{H}_t$ modélise le flou causé par le mouvement et la rotation de l'objet, ainsi que la projection 3D $\rightarrow$ 2D de l'objet F sur le plan image. Cet opérateur dépend principalement de trois paramètres, $\{P_t, a_t, \phi_t\}$, qui sont respectivement la trajectoire du FMO, son axe et son angle de rotation. La fonction $[\mathcal{H}_t M](x)$ correspond à la carte de visibilité, *i.e* le canal alpha, de l'objet. Elle permet de fusionner l'objet flou et le fond partiellement visible.

De par sa petite taille (4 cm), et sa vitesse élevée, la balle de tennis de table peut être difficile à détecter et à suivre surtout lorsque les fréquences d'acquisitions sont faibles. Cela est illustré pour différents sports sur la figure 2.7. Il en résulte que même si une balle de tennis de table est sphérique et semi-rigide, elle est perçue comme ellipsoïdale à cause du flou de mouvement très important.

FIGURE 2.7 – Exemples d'objets en mouvement rapide pouvant être considérés comme des FMO : tennis de table, tir à l'arc, volley-ball, tennis, tempête de grêle et insectes volants (ROZUMNYI, 2017)

Toutefois, ce flou de mouvement aurait pu être limité en utilisant des modalités d'acquisition différentes. En particulier, des caméras autres que les caméras couleur existent, comme par exemple les caméras événementielles, ou `Event camera`. Ces capteurs d'images réagissent aux changements locaux

de luminosité, mais ne capturent pas l'image comme le font les caméras classiques. Chacun des pixels des capteurs des caméras événementielles fonctionne de manière asynchrone et indépendante, entraînant un flou de mouvement bien moins important (voir figure 2.8) et une haute résolution temporelle. FALANGA, KIM et SCARAMUZZA, 2019 utilise ce type de caméra sur des drones afin que ceux-ci puissent détecter et éviter des obstacles de manière sécurisée. Cependant, leur résolution spatiale est plus faible, et les travaux sur les architectures d'apprentissage profond basés sur ce type de caméra sont encore rares et moins performantes que les approches basées sur les caméras classiques (PEROT et al., 2020).

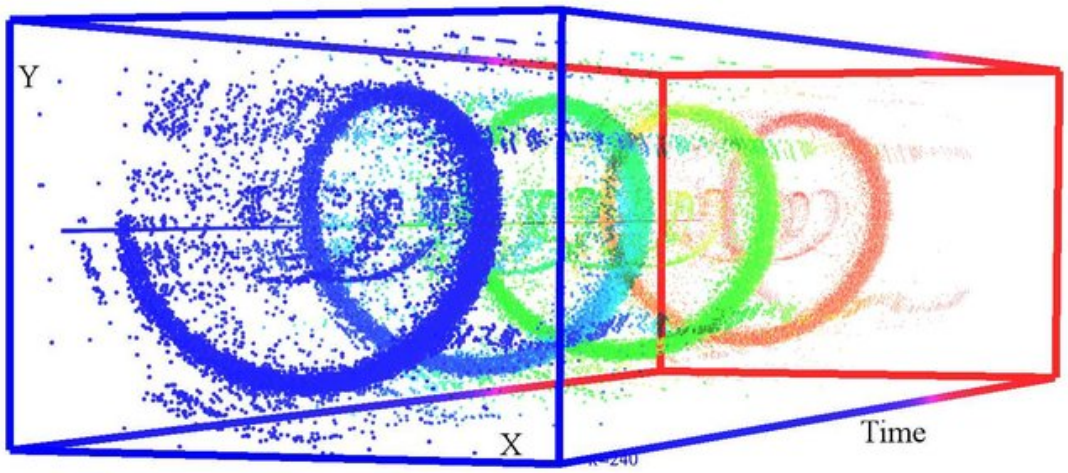


FIGURE 2.8 – Représentation de la sortie d'une caméra événementielle (GALLEGO et al., 2018)

Pour reconstruire les positions 3D successives d'une balle, il est indispensable de la détecter, et de la suivre sur l'ensemble de la séquence d'images. Dans cette thèse, nous commençons par la détection de balle. Comme dans la majorité des travaux dans le domaine, l'objectif de cette étape est d'identifier une région d'intérêt (RoI) englobant l'objet à détecter. Les difficultés pour l'obtention d'une bonne détection sont multiples, et elles sont liées à des perturbations durant l'acquisition qui sont illustrées à la figure 2.9.

La première difficulté est due aux conditions d'éclairage : des variations de luminance **(a)**, la présence de reflets, d'ombres, des effets de halos ou de scintillement peuvent avoir lieu notamment lorsque la fréquence d'acquisition est élevée.

La deuxième difficulté est la présence d'occultations **(b)** : un objet peut être visible uniquement en partie car positionné sur les bords de notre champ de vue ou occulté par un autre objet.

La troisième est la déformation apparente de l'objet observé. Lorsque le placement relatif de la caméra et de l'objet change, l'image de l'objet change **(d)**. L'apparence peut aussi changer si l'objet est déformable (par exemple, le changement de position dans le cas d'un être humain **(e)**), possède une apparence variable (deux personnes ne sont pas d'aspect identique), ou peut être aperçu en mouvement **(c)**. Dans le cas du tennis de table, les balles sont sphériques, donc leur apparence est ellipsoïdale lorsque projetées sur un plan. L'apparence de la balle est peu affectée par le changement de point de vue, mais la taille du grand axe de sa projection ellipsoïdale peut devenir importante par rapport à la taille de son petit axe quand leur vitesse est importante.

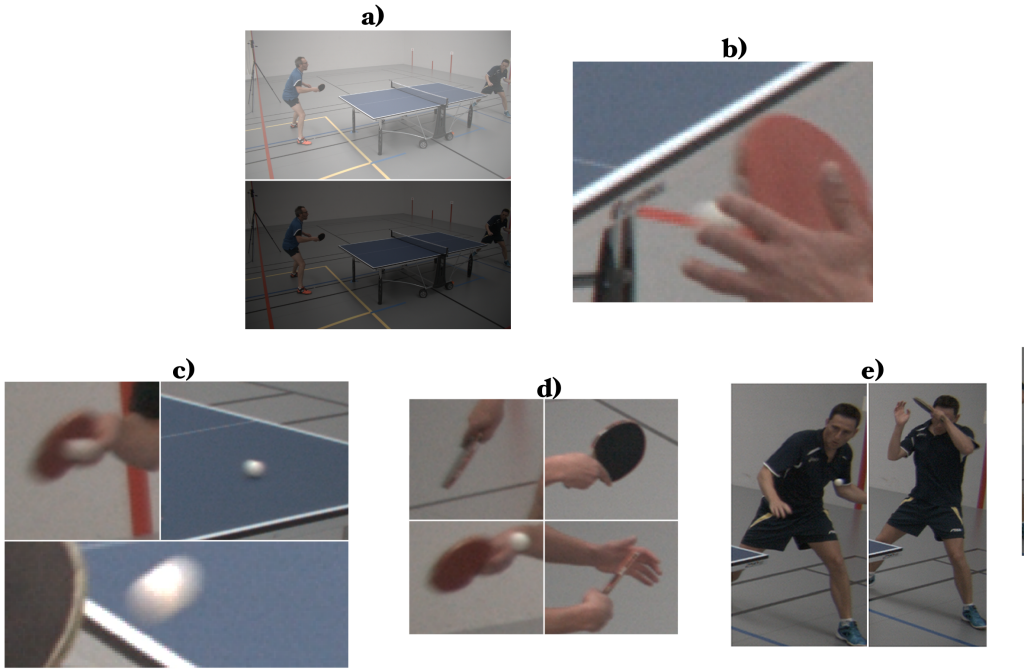


FIGURE 2.9 – Illustrations de difficultés liées à la détection d'objet a) changement de luminance, b) occultation de la balle, c) déformation et flou de mouvement, d) déformations liées au changement d'angle de vue, e) déformation d'objet souple (changement de position)

Un autre facteur déterminant, lié à la vitesse d'acquisition est le temps de traitement de l'algorithme de détection. En effet, un temps de calcul faible, de l'ordre de la durée entre deux d'acquisitions, augmente considérablement la plus-value de la méthode de suivi et les applications possibles.

En réponse à ces différents problèmes, nous présentons à présent les deux principales méthodes d'extraction de régions d'intérêt. La première est l'utilisation de fenêtres glissantes à différentes échelles, et la seconde est liée à la segmentation d'images.

Détection d'objets en mouvement

Détection par fenêtre glissante

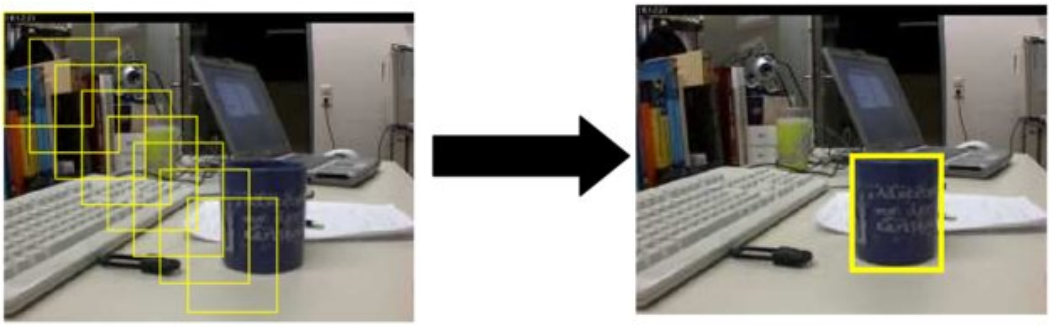


FIGURE 2.10 – Détection d'objet par fenêtre glissante (ROTH, 2008)

L'idée générale sur laquelle repose une détection par fenêtre glissante est la suivante : la région d'intérêt peut être définie comme une zone rectangulaire centrée sur l'objet à détecter. Après la détection initiale, par une méthode que nous discuterons par la suite, l'image est parcourue par une fenêtre de taille fixe, comme illustrée par la figure 2.10, et l'imagette ainsi définie est comparée à l'image de l'objet recherché. Pour pallier la variation de taille des objets dans une image, VO et al., 2021 utilise une pyramide d'images à différentes échelles, ce qui, de plus, permet d'accélérer la recherche.

Le choix du pas, c'est-à-dire le décalage spatial en pixel, vertical et horizontal, entre deux positions successives de la fenêtre glissante, est généralement laissé au choix de l'utilisateur. Un pas petit implique une augmentation de la précision, mais également une augmentation du coup de calcul. Pour chaque imagette, et pour chaque échelle, des caractéristiques sont ensuite extraites pour la comparer à l'image de l'objet suivi. Ces caractéristiques peuvent être issues de méthodes classiques (dites "*handcrafted*") de l'analyse de formes (FERRARI, JURIE et SCHMID, 2010 ; TOSHEV, TASKAR et DANIILIDIS, 2012), l'Histogramme de Gradient Orienté (HOG) (Jianquan LI et al., 2017 ; SEEMANTHINI et MANJUNATH, 2018) étant l'une des plus courantes. Actuellement, les méthodes de référence sont celles s'appuyant sur des méthodes d'apprentissage profond (Q. LI et al., 2018 ; MA et al., 2020).

Dans toutes ces approches, un score est attribué à chaque imagette, et seule celle ayant le meilleur score est conservée, en général après l'application d'un algorithme de suppression des Non Maxima (FORSYTH, 2014).

Les approches par fenêtre glissante restent cependant très lourdes en terme de temps de calcul car elles nécessitent l'analyse d'un grand nombre de fenêtres spatiales, et ce, à plusieurs échelles.

Détection par segmentation

Pour éviter la recherche exhaustive d'objets utilisée par la méthode précédente, la seconde méthode présentée ici est la segmentation. L'objectif, ici, est de prédire des boites englobantes ayant une grande probabilité d'être un objet cible. Une fois les boîtes obtenues elles sont ensuite envoyées à un classifieur (KU et al., 2018).

La classification n'est effectuée que sur les fenêtres ayant les scores de confiance les plus grands, permettant un gain très important en terme de coût de calcul. Cela est particulièrement intéressant pour les modèles ayant pour but le temps réel, ou lorsque le nombre d'images par seconde est élevé.

Cette approche est utilisée dans les réseaux profonds SSD (W. LIU et al., 2016) et YOLO (REDMON et al., 2016), dont l'approche par détection de boîtes englobantes puis sélection des meilleures candidates est illustrée par la figure 2.11 pour une détection d'objets multiples.

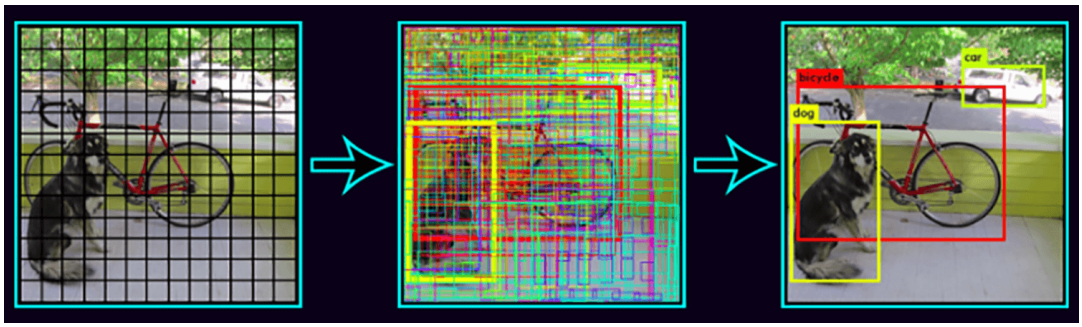


FIGURE 2.11 – Détection d'objets multiples, et sélection des meilleurs candidats. Source : YOLO (REDMON et al., 2016)

Dans notre cas, l'image de la balle étant fortement déformée, sa détection nécessite un classifieur robuste ou entraîné spécifiquement à la détection d'objets en mouvements (voir section 2.3).

Suivi d'objets

Dans le cadre de l'assistance automatisée à partir de vidéos dans le contexte sportif, le suivi d'objets ou de personnes, est fondamental pour l'analyse du geste sportif ou des phases dynamiques de jeu. Pour être exploitable, ce suivi doit se faire en temps réel ou, à défaut, avec une faible latence. En effet, à côté des analyses en temps long, c'est-à-dire durant le "débriefing" lorsque l'activité étudiée est terminée, qui pourraient se satisfaire d'un temps de calcul plus important, nous visons une assistance au fil de l'activité. Il s'agit

alors de fournir un retour rapide des résultats du logiciel sur le lieu même de l'entraînement, typiquement une salle de sport.

Dans le cas qui nous occupe, le suivi est un moyen d'obtenir les trajectoires des balles qui, nous le verrons par la suite, est une source riche d'informations sur le geste du joueur de tennis de table. Nous présentons figure 2.12 un exemple de suivi de balle au cours du temps pouvant servir pour aider un arbitre.

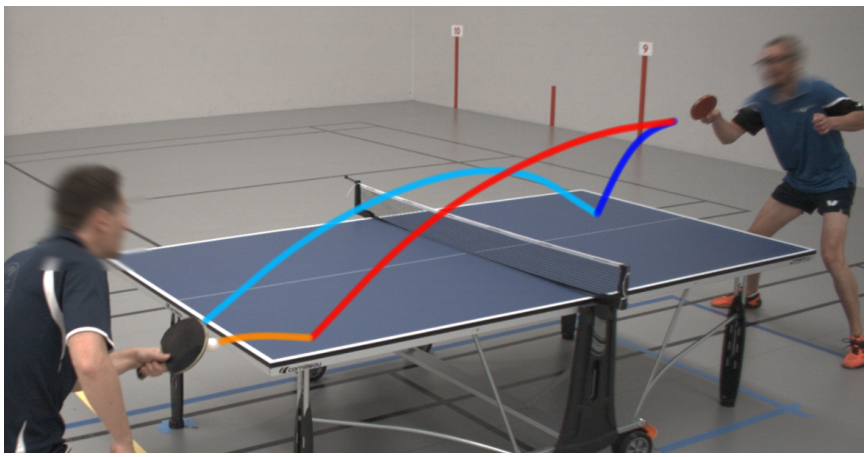


FIGURE 2.12 – Exemple d'application : suivi de balle pour l'analyse ou l'arbitrage

Pour obtenir cette trajectoire, dans un premier temps dans le domaine 2D de l'image, une détection sur chacune des images de la séquence vidéo pourrait être appliquée en utilisant les méthodes vues précédemment. Utilisée de façon systématique, cette méthode s'avérerait coûteuse en temps de calcul, puisque appliquée à toute l'image. De plus, elle n'est pas toujours possible sur chaque image à cause des perturbations (variations d'illumination, occultations, etc) déjà évoquées. Enfin, comme ce sont les trajectoires qui sont le véritable objet d'intérêt, des hypothèses de continuité du mouvement de la balle peuvent être utilisées afin de réduire la zone de recherche mais aussi de limiter les fausses détections.

Avant de préciser l'approche retenue dans notre cas, nous présentons tout d'abord les deux grandes familles de suivi d'objets.

Suivi par détection

Les algorithmes de suivi (les "*trackers*") basés sur l'utilisation de filtres de corrélation ont montré d'excellents résultats en termes de précision et de vitesse de suivi. Une des approches de référence est celle proposée par HENRIQUES et al., 2015, qui utilise des filtres de corrélation linéaires. Cette approche a été améliorée en utilisant des méthodes à noyaux (Kernelized Correlation Filter (KCF)(HENRIQUES et al., 2015)). D'autres améliorations ont été proposées, le

plus souvent pour mieux prendre en compte les changements de tailles des objets au cours du suivi. Ainsi, la méthode DSST (Discriminative Scale Space Tracker) (DANELLJAN et al., 2014) estime la position et l'échelle de la cible de façon itérative. Cette approche augmente la précision du suivi en utilisant deux branches de filtres de corrélation, mais réduit la vitesse de traitement. De plus, elle peut surmonter une occultation de courte durée.

BEWLEY et al., 2016 utilise un filtre de Kalman classique (W. YANG et al., 2010) : à partir d'un modèle de dynamique de l'objet, qui est utilisé pour prédire la position d'un objet sachant ses positions passées, des cibles candidates délimitées par des régions d'intérêt sont détectées et un score, *i.e.* une vraisemblance sur l'objet suivi, leur est associé. Ce score prend en compte le chevauchement entre les ROI prédites par le modèle et celles détectées : plus la zone de chevauchement est importante, plus le score est élevé. Actuellement, ce type d'approche, dont les bases sont relativement anciennes, est rendue plus robuste en prenant en compte les changements d'apparence des objets par l'utilisation de réseaux profonds pour la détection (WOJKE, BEWLEY et PAULUS, 2018).

Remarquons que, pour toutes ces méthodes, l'apprentissage du détecteur est disjoint de celui de la méthode de suivi. Dans certains cas, il semble pertinent d'effectuer les deux apprentissages conjointement.

Détection par suivi

Dans le cas où plusieurs objets, souvent visuellement similaires, doivent être suivis, la gestion des ambiguïtés est un problème majeur. Il s'agit typiquement, lors d'un croisement entre deux ou plusieurs objets, de les identifier individuellement pour correctement prolonger la trajectoire suivie par chacun. Les méthodes de référence reposent sur des modèles probabilistes qui, dès que le nombre de cibles est important, doivent faire face à une explosion combinatoire. Citons les algorithmes Joint Probabilistic Data Association (JPDA) (FORTMANN, BAR-SHALOM et SCHEFFE, 1983) et Multiple Hypothesis Tracking (MHT) (BLACKMAN, 2004).

Une approche développée actuellement est de convertir les détecteurs existants en trackers et de combiner les tâches de suivi et de détection. Dans ce cas, afin de comparer les différences entre deux images consécutives, de nombreux chercheurs s'appuient sur des réseaux siamois (FEICHTENHOFER, PINZ et ZISSERMAN, 2017 ; FANG, JO et LEE, 2020). L'idée principale de ces

réseaux, dans le cas du suivi, est de prédire les déplacements des ROI englobant les objets (FERNANDO et al., 2018)). Pour augmenter la vitesse d'inférence, ZHU et al., 2017 utilise le flot optique pour propager les caractéristiques des images précédentes à l'image courante. Sur des idées proches, les réseaux Transformers (BIAN et al., 2020) ou encore multi-canaux (N. ZHANG et al., 2020) sont également utilisés.

Cette approche de détection par suivi est très efficace lors du suivi d'objets multiples, comme par exemple celui d'une foule, ou lorsque les objets changent de forme. Un très bon cas d'application est le suivi d'êtres humains, dont la forme varie selon l'angle de vue, ou la position de la personne, son habillement, les occultations générées par ses mouvements, etc.

La détection par suivi nécessite une grande quantité de données étiquetées afin que l'apprentissage soit efficace. Obtenir ces annotations est généralement très coûteux en temps, et cela se relève donc peu attractif dans notre cas. De plus, la balle n'est pas soumise à un changement important de forme lorsque l'angle de vue est modifié ou qu'elle est en rotation. Les déformations observées sont surtout liées à la vitesse d'acquisition, comme c'est souvent le cas particulier des FMO (ZITA et ŠROUBEK, 2020b ; ZITA et ŠROUBEK, 2020a ; KOTERA et al., 2019). Un suivi par détection est donc plus approprié, et c'est l'approche que nous avons choisie et qui sera détaillée section 2.3.

2.1.3 Rétroprojection de la balle dans l'espace 3D

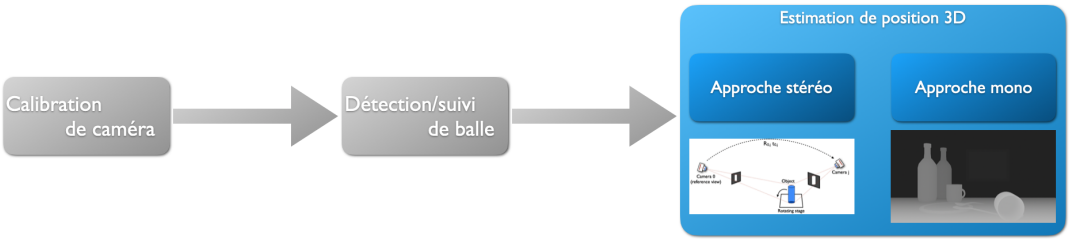


FIGURE 2.13 – Reconstruction 3D par triangulation (caméra stéréo) ou estimation de profondeur (caméra mono)

Après avoir obtenu les positions successives de la balle dans une séquence d'images, la prochaine étape du processus (voir figure 2.13) est l'estimation des coordonnées dans l'espace 3D de ces positions.

Les méthodes de reconstruction 3D sont divisées en deux grandes catégories : celles qui utilisent une seule caméra, dites monoculaires et celles qui utilisent plusieurs caméras, dites multi-vues et dont l'exemple le plus classique

est la stéréovision avec deux caméras. Nous allons principalement nous attarder sur l'approche n'utilisant qu'une seule caméra, puisque c'est celle qui est la plus adaptée à notre cas d'étude, mais nous exploiterons également la stéréovision pour la génération de vérité terrain dans la partie 2.2.1.

Reconstruction 3D multi-vues

Le problème de la reconstruction géométrique du monde 3D à partir d'un ensemble d'images est appelé en vision par ordinateur "reconstruction 3D multi-vues" (SEITZ et al., 2006). La mise en correspondance de points caractéristiques dans des images différentes, en connaissant les positions relatives de chacune des caméras (voir section 2.1.1), permet la reconstruction de la scène 3D en appliquant des techniques de triangulation (HARTLEY et ZISSERMAN, 2003) comme illustré à la figure 2.14. Le principe est similaire au fonctionnement de l'œil chez l'être humain et est très efficace sur des scènes texturées comme des bâtiments, dont nous illustrons un exemple figure 2.15.

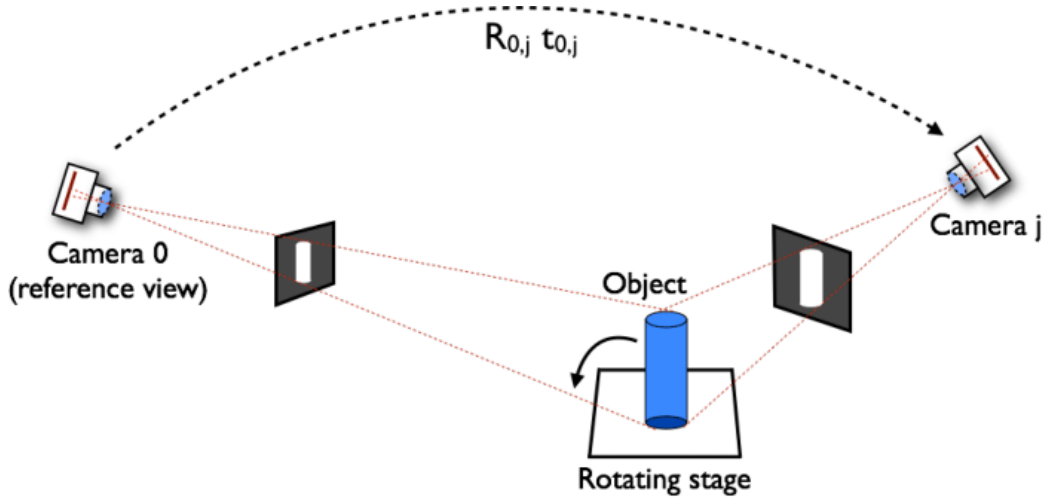


FIGURE 2.14 – Estimation de position 3D à l'aide de techniques de triangulation (LEVINE, MARTINELLO et NEZAMABADI, 2016)

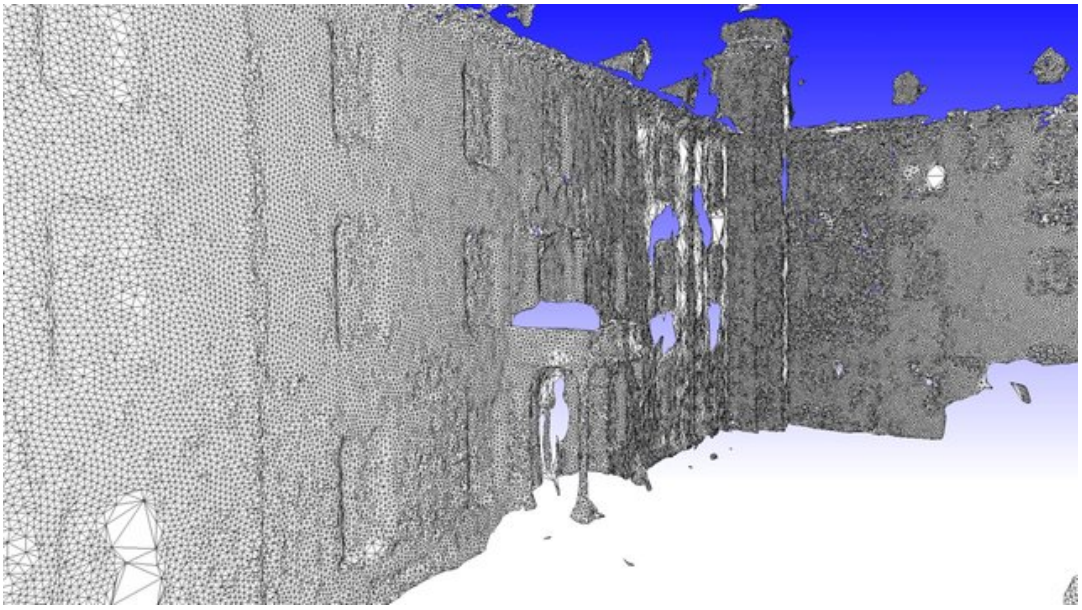


FIGURE 2.15 – Nuage de points 3D obtenu par mise en correspondance de points caractéristiques en stéréovision. (BARTELSEN et al., 2012)

Plusieurs problèmes peuvent compliquer la reconstruction (voir les problèmes liés à la détection d'objets expliqués précédemment, illustrés figure 2.9). Dans le cas où l'angle de vue entre les caméras est très différent, la mise en correspondance entre les observations peut s'avérer impossible. Une technique consiste à augmenter le nombre d'images d'un même objet vu sous des angles différents et les correspondances entre les nuages de points se font le plus souvent deux-à-deux. Cette approche conduit à une meilleure reconstruction de la scène. Par exemple, la présence de caméras tout autour d'une pièce permet d'observer un objet sous tous les angles. Entre deux caméras proches, les nuages de points peuvent être mis en correspondance, ce qui permet la reconstruction 3D partielle d'un objet. Ensuite, la fusion de toutes ces reconstructions partielles permet une reconstruction globale. Cette approche entraîne cependant un coût d'installation et de calcul très important, et nécessite une salle dédiée.

La reconstruction d'une scène 3D peut aussi se faire lorsque les caméras sont mobiles : une application très actuelle est par exemple, la reconstruction pour voitures autonomes. Toutefois, ce type d'approche ne sera pas abordé ici en détails puisque non pertinente dans le cadre de nos travaux.

Reconstruction 3D monoculaire et carte de profondeur

Après la présentation des reconstructions multi-vues, nous nous intéressons ici à la reconstruction 3D à partir d'une seule caméra. Comme précisé auparavant, la difficulté de reconstruire la scène à partir d'une image est liée à la perte de profondeur dûe à la projection de l'espace monde (3D) sur l'image (2D). La reconstruction des objets observés serait donc possible si cette information de profondeur était connue. Cette information est donnée par une *carte de profondeur* (Y. CHEN et VETRO, 2014). Cette carte est une image dans laquelle l'intensité de chaque pixel représente la distance entre la surface 3D et la caméra (voir figure 2.16).

FIGURE 2.16 – Représentation d'un espace 3D à l'aide d'une carte de profondeur. http://www.maurizioturoni.eu/depth-map/

Une carte de profondeur, en vision monoculaire, est généralement obtenue à l'aide de capteur dit "temps de vol", ou "*Time-of-Flight*" (TOF) (L. LI et al., 2014). Le principe est qu'un émetteur illumine la scène par un ensemble de rayons, typiquement lasers, ultrasons ou infrarouges. Ces rayons sont ensuite réfléchis lorsqu'ils entrent en contact avec une surface. Un récepteur les capte et le temps entre l'émission et la réception donne une information de distance entre les surfaces illuminées et la caméra. Comme la carte de profondeur n'est pas obtenue à l'aide de correspondance entre deux images, il y a moins de restrictions liées à la distance entre objets et caméra, que dans l'approche multi-vues.

Ces capteurs ToF sont de plus en plus populaires. Ils sont présents dans la Kinect de Microsoft V2 ou dans les technologies qui lui ont succédé [2], ainsi que dans certains téléphones et tablettes. On retrouve par exemple les capteurs ToF dans le "*Light Detection and Ranging*" (LiDAR) de l'iPad Pro pour des applications en réalité augmentée, et dans la caméra frontale des iPhones. Dans le cas de la reconnaissance faciale, les caméras frontales permettent la reconstruction 3D de la structure osseuse pour une amélioration de la fiabilité par rapport à une reconnaissance uniquement basée image.

Les capteurs de type ToF ont cependant des inconvénients. Par exemple, lorsque l'estimation de profondeur se fait grâce à la réflexion de rayons infrarouges, certains matériaux liquides, transparents ou absorbants peuvent ne pas être détectés, provocant des zones vides lors de la reconstruction 3D. Une seconde restriction est le mouvement. La qualité de l'estimation d'une surface peut être altérée lorsqu'un mouvement rapide est effectué : l'acquisition ne se faisant pas instantanément, un déplacement rapide entre l'émission et la réception des rayons entraîne un flou de mouvement (`Motion Blur`).

L'utilisation d'un unique capteur ToF, tout comme pour l'approche multivue, ne permet pas la génération complète d'un modèle 3D, et seule la zone recevant les rayons peut-être reconstruite. Pour pallier cet inconvénient, NGUYEN, HUYNH et MEUNIER, 2018 utilise des miroirs afin d'avoir une estimation de zones non couvertes. Malgré des résultats satisfaisant et ne nécessitant pas la synchronisation de plusieurs capteurs ToF, l'utilisation de miroirs reste peu adaptée à une utilisation dans un complexe sportif.

L'approche qui correspond le mieux à notre problème est celle de la reconstruction 3D à partir d'une seule caméra, essentiellement parce qu'elle est la moins intrusive et la plus simple à mettre en œuvre. Nous verrons section 2.3 comment nous contournons l'absence de carte de profondeur grâce à des connaissances *a priori* sur le tennis de table.

2.2 Jeux de données utilisés

Les méthodes développées récemment dans le domaine de la reconnaissance d'actions humaines sont de plus en plus performantes, notamment grâce à l'essor de l'apprentissage profond, que ce soit pour des jeux de données contrôlés ou en conditions réelles (dites *in the wild*). La nature et la complexité des jeux de données utilisés ont ainsi évolué au cours des dernières années, reflétant les nouveaux défis qui sont désormais abordés pour

2. La Kinect n'est plus commercialisée depuis 2017

la reconnaissance et l'analyse d'actions humaines. Ainsi, le nombre de classes peut différer, allant jusqu'à plus d'une centaine, comme par exemple la base SPORT1M (SRAVYAPRANATI et al., 2021), qui est un jeu de données sportif comprenant 487 classes, et pouvant atteindre des milliers de classes comme dans la base YouTube8M (ABU-EL-HAIJA et al., 2016), qui contient 4800 classes et plus de 8 millions d'extraits vidéos. En plus du nombre de classes, la complexité des actions présentes et des conditions de prise de vue peut varier. Parmi ces jeux de données, certains sont issus de situations de tous les jours, d'extraits de films, ou bien de vidéos sportives. Plus de détails concernant la reconnaissance d'actions humaines, seront apportés dans le chapitre 5.

La reconnaissance d'action humaine dans les jeux de données sportifs a consisté jusqu'à récemment à reconnaître le type d'action sportive présente sur la séquence vidéo (distinguer un saut en hauteur d'un lancer de poids par exemple). Dans notre cas d'étude, nous nous concentrons sur un seul sport et l'objectif est de pouvoir reconnaître et analyser les différents gestes effectués par un joueur. La particularité du geste sportif pour la reconnaissance d'actions est la faible variabilité interclasse (les coups diffèrent peu dans leur réalisation) et la forte variabilité intraclasse (chaque joueur a son "style" de jeu). On parle alors de reconnaissance et d'analyse d'actions à grain-fin.

Une autre particularité du contexte sportif est le peu de données annotées disponibles pour les algorithmes d'apprentissage, notamment les réseaux profonds. Les bases de données qui se focalisent sur le sport restent rares. UCF-101 (SOOMRO, ZAMIR et SHAH, 2012) contient une dizaine de sports, tandis que la base de données Olympic Sports (NIEBLES, C. W. CHEN et FEI-FEI, 2010) contient seize sports, avec pour chacun une cinquantaine de séquences vidéos. Ces bases de données, rassemblant des activités différentes, ne permettent pas d'analyser les différentes actions d'un même sport. Ainsi, dans le cadre du projet CRISP (voir section 1.1), dont le cas applicatif est le tennis de table, une cohorte importante (données annotées) pour les algorithmes d'apprentissage a été créée. Il s'agit de la première base de données annotée sur le tennis de table, nommée TTStrokes21, qui comprend 20 classes filmées en conditions réelles, et dont la taxonomie a été discutée et conçue avec des professionnels du tennis de table. Ce jeu de données est également utilisé dans le cadre du benchmark MediaEval (P. MARTIN et al., 2019 ; P. E. MARTIN et al., 2020) que nous organisons depuis maintenant trois ans.

Ce chapitre se concentre sur l'estimation dans l'espace 3D de la position de la balle. Nous ferons la distinction tout au long de ce manuscrit entre l'estimation des positions 3D de la balle (issue du domaine image) et sa trajectoire cinématique, qui est issue d'un modèle physique de la dynamique de la balle. Nous présentons dans la section 2.2.1 le processus d'acquisition de séquences sportives faite avec les sportifs du club pongiste de La Rochelle et qui nous serviront d'études.
Le chapitre suivant 3 concerne la reconstruction et l'étude des trajectoires cinématiques des balles, et notamment l'extraction de paramètres cinématiques pertinents que sont la vitesse de translation et la vitesse de rotation de la balle. En raison de l'absence de vérité terrain pour ces paramètres lors de l'utilisation de séquences réelles (section 2.2.1), nécessaire à la validation de ces travaux, nous avons également créé un jeu de données `synthétiques` dont les paramètres peuvent être fixés. La première partie de ce jeu de données synthétique sera présentée dans la section 2.2.2. De même, plus de détails sur la génération de trajectoires intégrant les différentes forces observées seront donnés par la suite dans la section 3.3. Nous nous focalisons ainsi dans ce chapitre uniquement sur l'estimation dans l'espace 3D des positions successives de la balle.

2.2.1 Jeu de données `avec sportifs` en conditions réelles

Le premier jeu de données est celui réalisé avec des sportifs professionnels du club pongiste Rochelais. Les séquences sont enregistrées sans marqueurs ni capteurs sur les joueurs, qui sont les conditions que nous souhaitons dans le cadre de ce travail. Les acquisitions ont été réalisées au gymnase du Service Universitaire des Activités Physiques Sportives et d'Expression (SUAPSE) de l'université de La Rochelle, qui a l'avantage d'avoir un éclairage naturel (verrière) permettant d'éviter les problèmes de scintillement de la lumière artificielle. La fréquence de scintillement est en effet 50 Hz, qui est donc en dessous de notre fréquence d'acquisition (fixée au minimum à 240 images par seconde pour ces images RGB). Malgré cette fréquence d'acquisition, un flou de mouvement est observé sur les images de balle, comme illustré sur la figure 2.17.

Comme indiqué auparavant, nous souhaitons effectuer l'analyse du geste sportif en n'utilisant qu'une seule caméra. Toutefois, deux caméras (voir tableau 2.1) sont utilisées afin d'avoir une vérité pour l'entraînement de notre

FIGURE 2.17 – Flou de mouvement sur une balle observée à 240 fps

modèle d'estimation de la position de la balle dans l'espace 3D (présenté en section 2.3).

Un enregistreur vidéo numérique (DVR) (voir tableau 2.2) connecté à un ordinateur permet de synchroniser temporellement les acquisitions des deux caméras, ainsi que de les configurer (luminosité, temps d'exposition, fréquence d'acquisition ...).

TABLE 2.1 – Spécifications techniques des caméras utilisées (Flare 2M280CCX)

Marque			IO Industries
Type			Camera
Nom			Flare 2M280CCX
Taille du capteur			2048×1088
Fréquence d'images (pleine résolution)		8-bit	337 fps
		10-bit	281 fps
Type de capteur			CMOSIS CMV2000
Taille des pixels			$5,5 \times 5,5 \mu m$
Plage dynamique / Sensibilité			60 dB, 5,56 V/lux.s

TABLE 2.2 – Spécifications techniques du DVR Core2CX

Marque		IO Industries
Type		DVR Express
Nom		Core2CX
Débit de données vidéo supporté		Jusqu'à 1620 MB/s
Supports de stockage vidéo		Jusqu'à 30TB
Format d'enregistrement		Non compressé (système de fichiers IO Industries)
Connexions CoaXPress prises en charge	Single-Link	4 (jusqu'à CXP-3)
	Dual-Link	2 (jusqu'à CXP-3)

Le choix du positionnement des caméras est important. Ainsi, lorsque deux caméras sont placées d'un même coté de la table, beaucoup d'occultations peuvent subvenir, notamment lors de la frappe d'un joueur dans la balle (voir figure 2.18). D'autre part, un droitier occultera la balle si la caméra est placée à sa droite lors d'un coup droit, à gauche pour un revers, et inversement pour les gauchers. Positionner une caméra de chaque côté, et derrière un des joueurs, permet en grande partie de limiter les instants durant lesquels la balle n'est visible sur aucune caméra, et augmente la robustesse de la triangulation. Nous avons donc opté pour cette approche.

FIGURE 2.18 – Occulation de la balle par la raquette ou les bras du joueur

Une autre possibilité est l'utilisation de caméras suspendues au-dessus du terrain, cependant, ce choix est rarement fait pour la captation de séquences sportives pour les raisons suivantes.
Premièrement, de par la projection planaire due à l'acquisition vidéo, un objet en mouvement sera perçu dans la grande majorité des cas comme un déplacement rectiligne. Toujours en raison de la projection planaire, l'information de hauteur de la balle est difficilement observable, or, elle est essentielle

pour les sports de balle comme le golf, le football, le volleyball, le tennis, le tennis de table et bien d'autres.
Deuxièmement, la mise en place d'une caméra suspendue au-dessus d'une scène est plus complexe que celle d'une caméra sur trépieds. Les halls de sport sont souvent équipés de plafonds hauts, rendant difficile l'installation de caméras, et une caméra trop distante entraînerait une baisse de la résolution spatiale.
Enfin, la plupart des vidéos disponibles sur Internet ou lors de diffusion d'événements sportifs sont obtenues par des caméras positionnées sur le côté de la scène. Avoir une prise de vue similaire permet donc d'étendre potentiellement nos travaux sur ces vidéos.

Dans notre cas d'étude du tennis de table, nous plaçons une caméra de chaque côté de la table et à une hauteur suffisante pour observer la retombée de la balle. De plus, les caméras sont orientées de façon à être focalisée sur un joueur afin d'en analyser le geste. La figure 2.19 illustre le positionnement des deux caméras. Ce choix de positionnement est assez classique pour reconstruire des trajectoires de balles (LIN, YU et Y. C. HUANG, 2020), et a l'avantage d'être simple à installer, tout en ne gênant pas les sportifs.

FIGURE 2.19 – Mise en place et positionnement des caméras

Quatre types de séquences vidéos ont été acquises :

- Séquences *Calibration* : ces séquences sont utilisées pour la calibration des caméras et l'estimation des paramètres intrinsèques (voir section 2.1.1). Ce sont des séquences courtes, contenant des mires de calibration en damier (*checkerboards*) positionnées sur la table ou à ses alentours, comme illustré par la figure 2.20.
- Séquences *Gammes* : ces séquences sont des répétitions d'un même type de coup effectué par un joueur (appelé une *gamme*). Les gammes sont fréquentes en début d'entraînement (par exemple le joueur A fait des coups de type `Top Spin` en coup droit, et le joueur B fait des coups de type `Poussettes`).
- Séquences *Gammes alternées* : ces séquences sont similaires aux gammes, à la différence que les joueurs alternent entre deux coups. Ces gammes alternées sont également utilisées pendant les entraînements. On distingue ici deux types d'alternance. Soit un joueur alterne entre un coup droit et son équivalent en revers, soit le joueur alterne entre deux coups différents.
- Séquences *Match* : enfin, le dernier type de séquences contient des matchs, ou des extraits de match entre deux joueurs.

FIGURE 2.20 – Extrait d'un séquence de calibration

Afin d'avoir une grande variété de vidéos, le joueur filmé change son type de gamme entre chaque vidéo. Après avoir fini ses gammes, il change de côté avec le joueur adverse qui est alors filmé à son tour.

La liste du protocole d'acquisition, précisant le matériel nécessaire, la mise en place des caméras, la phase de calibration à faire au préalable, et la liste des séquences à enregistrer sont donnés en annexe A.

Le contenu de chaque vidéo est ensuite annoté manuellement suivant le coup effectué. Les séquences annotées correspondent à un seul échange de balle. C'est-à-dire que la séquence commence lorsque l'adversaire frappe la balle. Le joueur filmé effectue un coup, et la séquence se termine lorsque l'adversaire frappe à nouveau la balle. Une vidéo d'un seul coup peut comporter entre 250 et 350 images, selon la vitesse de translation de la balle.

Dans notre jeu de données `avec sportifs`, nous nous intéressons plus principalement à trois types de coups : le `Top Spin`, la `Contre-Attaque`, et la `Poussette` (voir figure 2.21). Ces coups ont des trajectoires typiques qui diffèrent les unes des autres en termes de vitesse de translation et de rotation, et donc au final, d'effet donné à la balle.

FIGURE 2.21 – Extrait d'un `Top Spin` (gauche), d'une `Contre-Attaque` (centre) et d'une `Poussette` (droite)

Les `Top Spin` sont des coups offensifs, obtenus lorsque le joueur génère un frottement entre la raquette et la balle en utilisant une action de frappe vers le haut. La balle accélère et tourne à une vitesse élevée.

Une `Poussette` est un coup défensif. Par opposition au `Top Spin`, il a un backspin (rotation dans le sens inverse), qui est obtenu en commençant le coup au-dessus de la balle, et en la brossant dans un mouvement vers le bas.

Une `Contre-Attaque` est un coup offensif utilisé lorsque l'adversaire effectue également un coup offensif, parfois appelé `contre-attaque`. La vitesse de rotation et la vitesse de translation sont plus élevées, mais restent inférieures à celles d'un `Top Spin`.

Les coups annotés dans les séquences de type `Gammes` et de type `Gammes alternées` sont ensuite regroupés pour former un seul jeu de données contenant les trois classes de coup présentées ci-dessus. L'ensemble des vidéos acquises est résumé dans le tableau 2.3.

TABLE 2.3 – Tableau récapitulatif des vidéo acquises, et le nombre de segments annotés

Type de séquence	Nombre de vidéos	Segments annotés temporellement
Calibration	12	15
Top Spin	9	12
Contre-Attaques	3	8
Poussette	5	13
Top Spin Revers	5	-
Poussette Revers	3	-
Services divers	5	-
Matches	3	-

Des exemples vidéos de chacun de ses coups sont visualisables en ligne [3].

2.2.2 Jeu de données `synthétiques`

Suite à la pandémie de COVID-19, la fermeture des laboratoires ainsi que la fermeture des halls et clubs sportifs ont retardé la mise en place d'acquisitions. Quelques acquisitions avaient été effectuées au début de cette thèse, mais l'impossibilité de refaire des acquisitions durant les périodes de confinement nous a amené à créer une base de données synthétiques pour continuer le développement de nos travaux. Les jeux de données synthétiques, lorsque ceux-ci sont adaptés au problème abordé, présentent de nombreux points positifs. Un gros avantage est qu'il est possible d'inclure un modèle physique lors de la génération de cet ensemble de données. Dans notre cas d'étude du tennis de table, cela permet d'obtenir une vérité terrain sur les paramètres cinématiques des balles pendant les échanges (vitesses de translation, de rotation). Cela n'était pas possible lors d'acquisitions sans marqueurs de séquences réelles. Un autre avantage est la possibilité de rapidement générer des grandes quantités de données sans faire d'acquisitions sur le terrain. Notre contrainte initiale était la fermeture des clubs sportifs à cause de la pandémie de COVID-19, mais d'autres travaux exploitent cette génération de données. Par exemple en utilisant le jeu GTAV, HURL, CZARNECKI et WASLANDER, 2019 créé une base de données contenant à la fois des images et des informations LiDAR pour des applications aux voitures autonomes.

3. https://vimeo.com/jordancalandre

Dans les cas où la construction de jeux de données est coûteuse et que l'obtention d'une grande quantité de données est longue et fastidieuse, les données synthétiques peuvent également s'avérer utiles. Pour pallier ces problématiques de quantité de données, CUNHA et al., 2020 entraîne un réseau de neurones convolutifs (CNN) sur des données séismiques synthétiques, puis transfère l'apprentissage (*Transfer Learning*) sur un jeu de données `avec sportifs` de plus petite taille. Dans le cas d'applications sportives, la création de données nécessite des locaux spécifiques, plusieurs joueurs afin d'éviter un sur-apprentissage, et des connaissances expertes pour annoter manuellement les jeux de données. Le *Transfer Learning* à partir de données synthétiques est donc tout à fait adapté.

Afin de générer cet ensemble de données, nous avons utilisé le logiciel Blender (BLENDER ONLINE COMMUNITY, 2013). Pour reproduire au mieux le rendu de la scène réelle avec des sportifs, les caméras sont placées de façon identique. Pour cela, un modèle 3D pour la table est tout d'abord positionné au centre de notre repère. Cette position de la table par rapport au repère caméra est connu sur les séquences avec sportifs grâce aux matrices extrinsèques (voir section 2.2.1). CE qui permet d'obtenir la position des caméras à partir de l'équation 2.3.

Un aperçu de la scène est présenté sur la figure 2.22. Afin d'avoir la même distance focale, nous affectons aux caméras dans Blender les paramètres intrinsèques des caméras physiques de l'équation 2.1. Les rendus entre les deux jeux de données sont donc très similaires, comme illustré par la figure 2.23. Bien que les deux caméras sont positionnées pour comparaison avec la scène réelle, dans la pratique une seule sera utilisée. La deuxième caméra servait uniquement pour la triangulation de la balle dans le cas réel, elle est inutile dans ce jeu de données `synthétiques` où toutes les positions 3D sont connues.

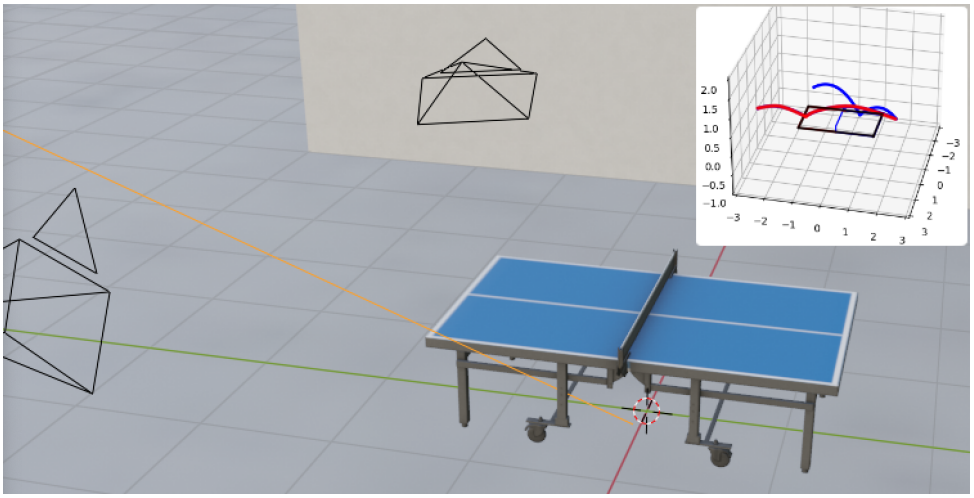


FIGURE 2.22 – Génération de scène synthétique et positionnement des caméras

FIGURE 2.23 – Aperçu d'une séquence réelle à gauche, et d'une scène synthétique à droite

Un modèle 3D de balle (sphère) est ensuite utilisé, puis nous simulons des trajectoires 3D réalistes de la balle, prenant en compte les forces s'appliquant sur une balle lors d'un échange (voir chapitre 3), et les différents rebonds (voir chapitre 4). Une fois la scène recréée, et les trajectoires de balle générées, un flou de mouvement est simulé avec l'outil de rendu *Cycles* [4] (voir Fig. 2.22) afin de reproduire au mieux les conditions réelles. Ce flou de mouvement généré est illustré sur la figure 2.24.

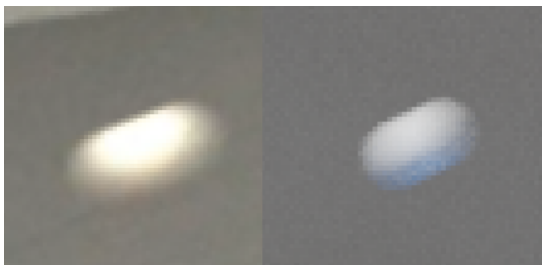


FIGURE 2.24 – Flou de mouvement réel (à gauche) et image générée (à droite)

4. https://docs.blender.org/manual/en/latest/render/cycles/index.html

Le jeu de données synthétiques obtenu est composé de 200 séquences vidéos (représentant un total de 60 676 images). Tout comme les séquences réelles, chaque vidéo est composée d'un échange entre deux joueurs. Elle démarre lorsque le joueur opposé à celui observé frappe, et se termine lorsque la balle est à nouveau frappée. Chaque séquence synthétique comporte donc un seul coup effectué par le joueur observé. La position 3D initiale ainsi que les paramètres cinématiques (angle de frappe, rotation ...) sont choisis de façon aléatoire, et seront expliqués en détails dans la section 3.3. En plus des images générées, sont sauvegardées les positions 3D des balles dans la scène synthétique, mais aussi les positions du centre de la balle dans l'image obtenues à partir des matrices de calibration et de l'équation 2.3.

2.3 Estimation de la position 3D à l'aide de CNN

Après présentation des différents jeux de données utilisés ainsi que les différentes étapes nécessaires à la reconstruction 3D, nous présentons ici l'approche choisie dans ces travaux pour extraire les positions 3D de la balle lors d'un échange. Nous commençons par présenter la préparation des données dans la section 2.3.1, ce qui inclue la calibration, la détection et le suivi de balle. Pour la principale contribution de ce chapitre, c'est-à-dire la reconstruction 3D des positions successives de la balle à partir d'une séquence 2D, nous avons fait le choix d'utiliser un réseau neuronal convolutif pour obtenir la taille de la balle en pixels. Dans cette section, nous définissons la taille de la balle en pixels comme la longueur du petit axe de l'ellipsoïde apparente. La balle ayant un diamètre fixe dans l'espace monde (4 *cm* depuis octobre 2000), sa taille en pixels dans le domaine image permet d'obtenir sa distance à la caméra. Nous présentons ce réseau dans la section 2.3.2, qui permet d'extraire cette taille de balle malgré la présence de flou de mouvement.

2.3.1 Préparation des données

Avant d'introduire l'architecture du réseau convolutif utilisé, nous allons présenter dans un premier temps les étapes préliminaires que sont la calibration, la détection et le suivi de balle (voir figure 2.1).

Calibration

La première étape est la calibration des caméras. Cette étape n'est réalisée que sur les séquences avec sportifs. Elle n'est pas nécessaire pour la base

de données synthétiques qui utilise à l'identique la configuration des caméras : la matrice intrinsèque estimée sur les séquences réelles est directement fournie au logiciel Blender pour générer les séquences synthétiques.

Matrice Intrinsèque

La première étape du processus consiste à obtenir les paramètres intrinsèques de la caméra. La matrice intrinsèque contient la distance focale, le point principal, et permet la correction des distorsions géométriques. Cette opération n'est effectuée qu'une seule fois avant de commencer les acquisitions.

Les acquisitions ayant été faites avec des mires de calibration (voir section 2.2.1), cette matrice intrinsèque est obtenue en suivant la méthodologie décrite par GEIGER et al., 2012. Les principales étapes sont données ci-dessous.

Tout d'abord, les mires sont détectées sur les séquences de la base *Calibration*. Cette détection des damiers est réalisée en utilisant la bibliothèque *Libcbdetect* (FTDLYC, 2020). Il s'agit d'une réimplémentation en C++ de (GEIGER et al., 2012 ; SCHONBEIN, STRAUS et GEIGER, 2014), originellement en Matlab.

La géométrie des mires étant connue (un damier), les paramètres du modèle de distorsion de l'image (voir équation 2.2) permettant de la rectifier sont estimés. Connaissant la largeur en centimètres des cases, la distance focale est calculée à partir de leur largeur en pixels dans le domaine image. Cette calibration est effectuée en utilisant la bibliothèque *libcalib* (FTDLYC, 2019).

Matrice extrinsèque

La deuxième étape essentielle au calibrage est l'obtention de la matrice extrinsèque $\mathbf{P}$, qui contient les informations de positionnement de caméra (rotation et translation) dans la scène.

Dans le cas général, pour trouver les coefficients de cette matrice, il faut considérer un ensemble de points de référence dans les images 2D dont les coordonnées Monde (3D) sont connues. Cela n'est pas toujours possible sans annotation manuelle préalable. Cependant, dans notre cas, toutes les dimensions de la table sont standardisées (voir figure 2.25).

Il y a au total six coefficients à obtenir. Trois correspondent au positionnement de la caméra (trois axes 3D), et trois autres définissent l'orientation de la caméra (trois angles polaires). Le nombre minimum de points permettant de résoudre le système 2.3 est donc de six.

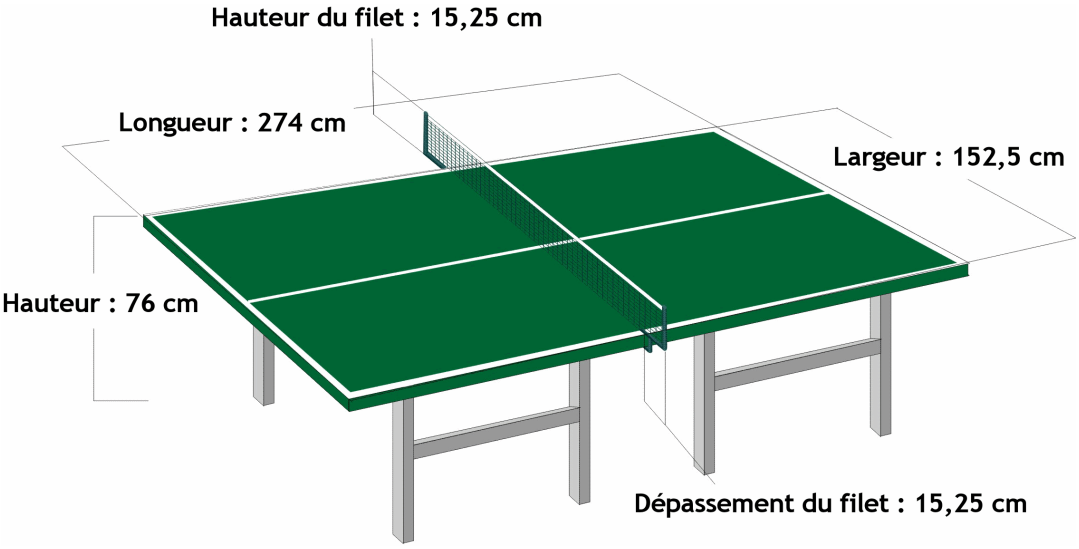


FIGURE 2.25 – Dimensions de la table au tennis de table https://fr.wikipedia.org/wiki/Tennis_de_table/media/Fichier:Table_de_tennis_de_table_fr.png.

Toutefois, pour augmenter la précision de l'estimation de ces coefficients, il est souhaitable d'utiliser davantage de points. Nous disposons de 12 points de référence facilement identifiables : les 4 coins de la table, les 3 points sur la ligne centrale, les 2 extrémités du filet, ainsi que 3 points sur le sol, à l'aplomb des coins de la table représentés en verts sur la figure 2.26. Cette annotation est faite manuellement sur les images rectifiées, mais elle pourrait être remplacée par une détection automatique de lignes (SZENBERG, CARVALHO et GATTASS, 2001). La caméra étant fixe, l'intérêt de cette automatisation est cependant limité, mais pourrait être mise en place si le jeu de données comportait différents placements de caméras. Les points de référence sont représentés sur figure 2.26.

Pour chaque point annoté, deux équations sont obtenues (HARTLEY et ZISSERMAN, 2003), correspondant à la relation entre la matrice extrinsèque, les coordonnés Monde, et les coordonnés Image. En reprenant les notations introduites dans 2.3, le calcul de cette matrice $\mathbf{P}$ est effectué en résolvant l'ensemble des équations données pour chaque point, comme ci-dessous :

$$\begin{bmatrix} \mathbf{0^T} & -\mathbf{W^T} & v\mathbf{W^T} \\ \mathbf{W^T} & \mathbf{0^T} & -u\mathbf{W^T} \end{bmatrix} \begin{bmatrix} \mathbf{P^1} \\ \mathbf{P^2} \\ \mathbf{P^3} \end{bmatrix} = 0 \tag{2.4}$$

La résolution du système engendré par les 12 points de référence est faite en utilisant l'algorithme "Perspective-n-point" (HARTLEY et ZISSERMAN, 2003). Plus précisément, nous utilisons la méthode *SolvePnP*, avec optimisation de

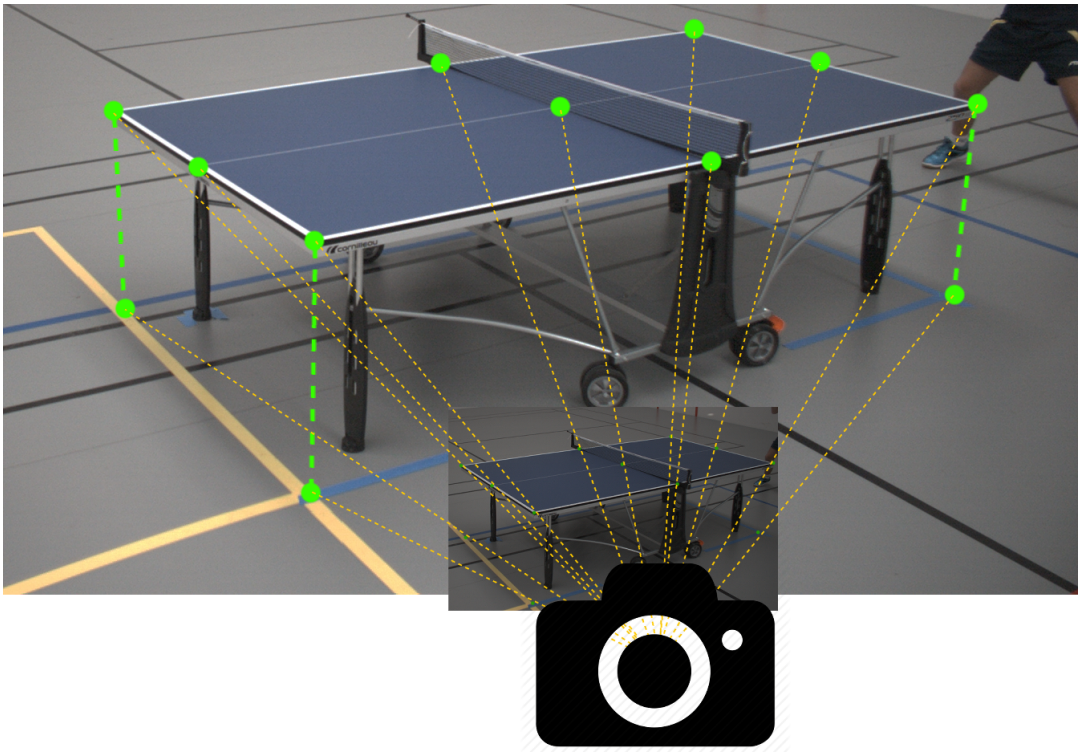


FIGURE 2.26 – Points de référence (en vert), et mise en correspondance entre les coordonnées Monde et Image.

Levenberg-Marquardt en choisissant le centre de la table comme origine du repère Monde. Nous obtenons ainsi les paramètres de la matrice extrinsèque, qui permet de passer du repère caméra au repère table et inversement.

Détection de la balle

L'étape suivante consiste à détecter puis à suivre la balle dans une séquence d'images. Même avec une fréquence d'acquisition élevée (dans notre cas, 240 images par seconde), la balle est souvent perçue comme une forme floue et ellipsoïdale (voir figure 2.24). Avec la fréquence d'acquisition choisie (240fps), nous ne sommes cependant pas dans le cadre des FMO, puisque nous avons un recouvrement de la surface perçue de la balle entre deux images consécutives. Cela simplifie la détection et le suivi de balle au cours du temps.

Nous avons opté pour une méthode de détection pour initialiser un algorithme de suivi, permettant d'avoir à la fois une bonne détection de balle, et une méthode globalement rapide : le suivi permet de ne pas avoir à détecter la balle sur chaque image.

L'étape de détection de la balle repose sur le framework `Detectron2` (WU et al., 2019) développé en PyTorch (PASZKE et al., 2019) pour la segmentation et la détection d'objets. Contrairement au détecteur de référence YOLO (REDMON et al., 2016), `Detectron2` détecte la balle de tennis de table, malgré le flou de mouvement. Cette détection échoue parfois pour quelques images lorsque le flou de mouvement est trop important (pour quelques coups `Top Spin` par exemple), surtout lorsque la balle est proche de la caméra, retardant la première détection sur la séquence d'image.

Suivi temporel de la balle

La première instance de balle détectée permet d'initialiser un algorithme de suivi et d'obtenir une séquence de points 2D représentant le centre de la balle sur l'ensemble de la séquence d'images considérée. Nous avons retenu le tracker CSRT (LUKEZIC et al., 2018), car bien que légèrement plus lent que le tracker KCF, il est plus précis et s'adapte bien aux changements d'échelle (MI et M.-T. YANG, 2019), de déformation et de rotation de la cible.

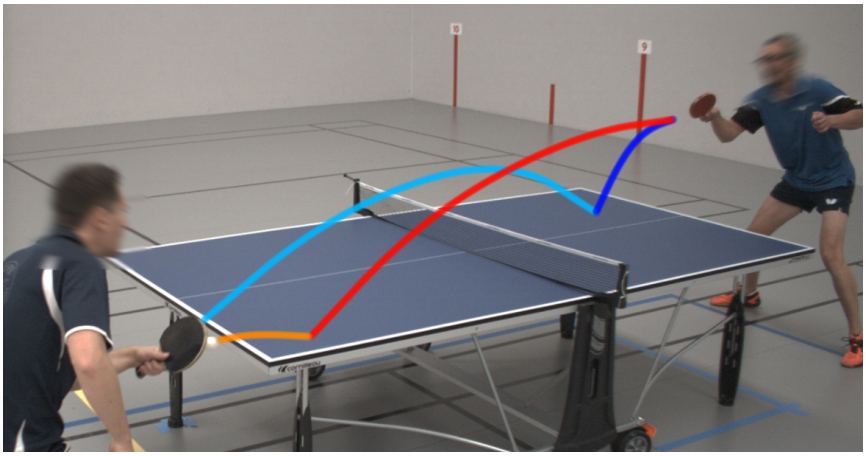


FIGURE 2.27 – Segmentation de la trajectoire de balle à partir de ses impacts.

Comme indiqué en préambule de ce chapitre, nous distinguons la séquence de positions 3D d'une balle - obtenue grâce au suivi, de sa trajectoire, qui intègre un modèle physique de la dynamique du mouvement, et qui sera abordée dans le prochain chapitre. De manière générale, la trajectoire d'une balle lors d'échanges peut être représentée par un ensemble de courbes définies par morceaux, chaque morceau de courbe étant connecté au suivant par un impact sur la table ou sur une raquette. La balle, lorsqu'elle rebondit sur la table ou qu'elle est frappée par une raquette, change rapidement de direction. Nous utilisons ces changements pour segmenter temporellement les

trajectoires de la balle. La figure 2.27 illustre une situation typique d'échange entre deux joueurs composé de divers segments pseudo-paraboliques.

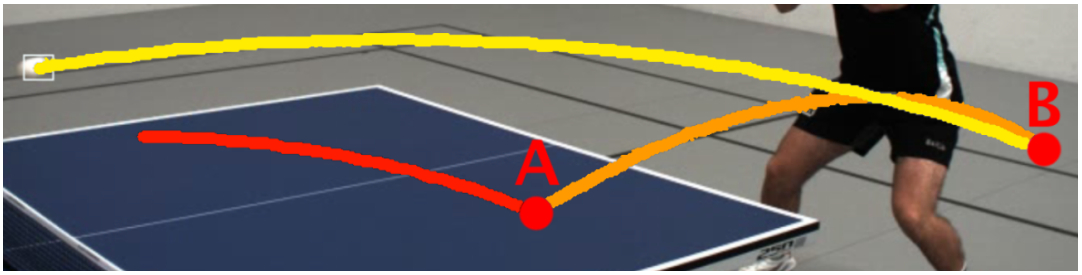


FIGURE 2.28 – Représentation des deux types de points d'impact sur une image. A : rebond sur la table, B : impact avec une raquette

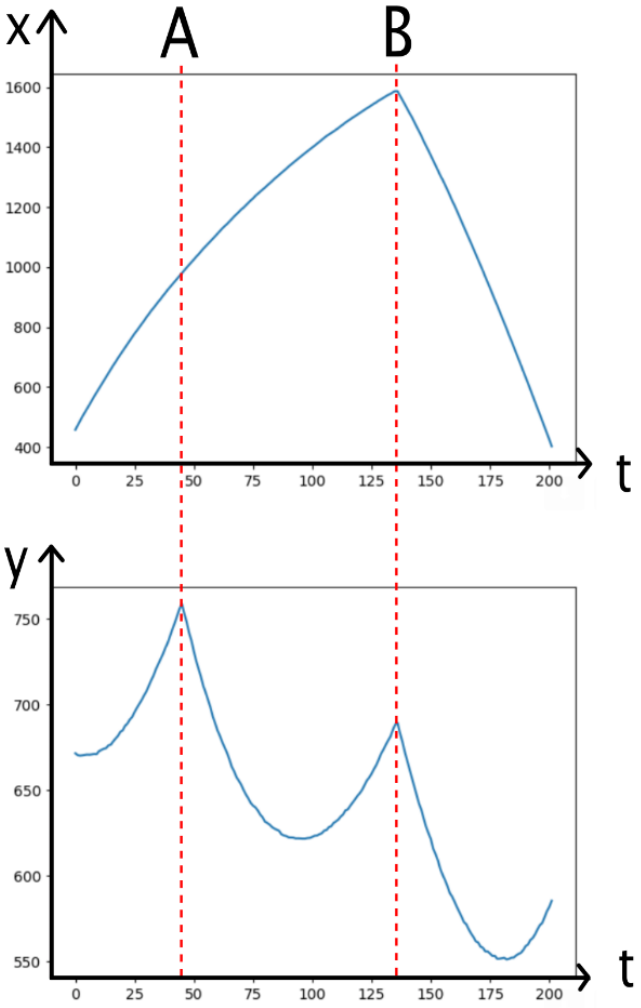


FIGURE 2.29 – Variation de la position de balle sur les axes x et y, permettant une segmentation temporelle de la trajectoire. A : rebond sur la table, B : impact avec une raquette

Les impacts correspondent à un changement brusque de direction de la balle. Par exemple dans la figure 2.28 représentant la trajectoire sur une demie-table, la trajectoire peut être divisée en trois intervalles : une première courbe allant de la frappe initiale, hors champ de vision, jusqu'au rebond sur la table au point A, puis une deuxième courbe de ce point jusqu'à l'impact sur la raquette du joueur au point B et enfin une troisième courbe partant du point B jusqu'à un point hors caméra. Le point A est caractérisé par un changement de direction selon l'axe y dans le repère image et le point B par un changement de direction selon les axes x et y dans le même repère. La figure 2.29 montre l'évolution de la position de la balle au cours du temps dans l'espace 2D sur chacun des deux axes. Nous utilisons une détection de minima locaux pour détecter ces changements et les classer comme *rebond table* ou *rebond raquette*.

2.3.2 Obtention de la taille de la balle dans le domaine image

Pour obtenir la succession des positions 3D de la balle au cours du temps, nous devons obtenir la distance caméra-balle en vision monoculaire. Il faut pour cela estimer la taille de la balle (son diamètre) dans le domaine image, malgré un flou de mouvement important.

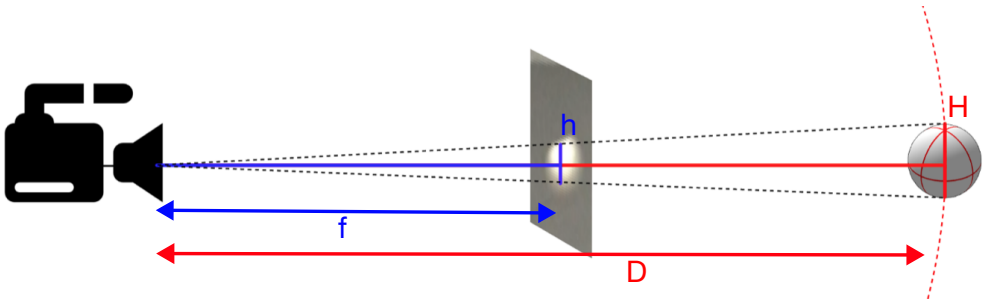


FIGURE 2.30 – Représentation de l'homothétie permettant la rétroprojection à partir de la distance focale f et la taille de la balle en pixels h

Dans la plupart des problèmes de reconstruction 3D en caméra monoculaire, l'obtention de cartes de profondeur précises est faite à l'aide de plusieurs caméras en stéréovision ou par les techniques de type ToF (Time of Flight), comme évoqué dans la section 2.1.3. Dans notre cas, nous pouvons exploiter la géométrie de la balle qui est une sphère de taille connue (4 cm), et qui, combinée avec sa taille observée en pixels dans l'espace image, permet de déduire sa distance à la caméra à l'aide d'une homothétie comme illustrée par la figure 2.30.

Étant donné la distance caméra-balle considérée, la projection de la balle peut être approximée raisonnablement comme un disque. Elle est très peu soumise à des variations de formes selon le point de vue. Les seules modifications de forme sont liées à son déplacement, transformant ce disque en ellipse (voir Fig. 2.9.c)

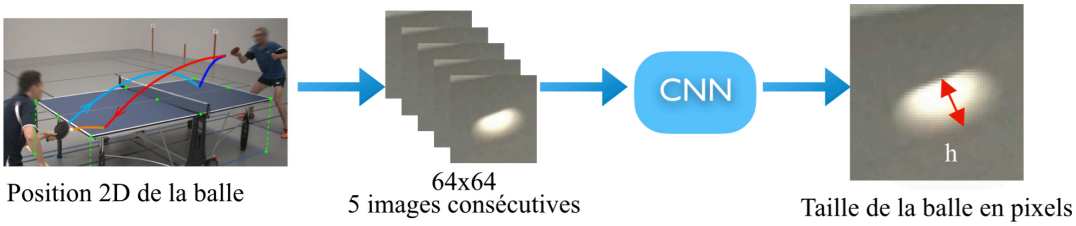


FIGURE 2.31 – Schéma représentant les différentes étapes de l'estimation de la taille de la balle en pixels

L'estimation de la taille de la balle, c'est-à-dire son diamètre, est donc compliquée par le flou de mouvement qui empêche une segmentation précise. Si l'on dispose d'une fréquence d'acquisition très grande, par exemple supérieure 1000 fps (images par seconde), et que la résolution spatiale de capture est très élevée, la balle peut être perçue comme un disque net. Toutefois, ces conditions d'acquisition demandent un matériel adapté onéreux, ainsi qu'une grande capacité de stockage. Cela n'est pas envisageable dans l'objectif d'une utilisation grand public et utilisable en conditions réelles. Ainsi, dans nos acquisitions (fréquences d'environ 240 fps), les contours de la balle sont perçus comme flous et de forme ellipsoïdale. De plus, son diamètre, en pixels, est relativement petit par rapport à la dimension de la scène. Il est donc difficile d'exploiter de telles séquences en utilisant des méthodes classiques de segmentation d'objet. Pour pallier ce problème, nous avons mis en œuvre une méthode de régression pour estimer le diamètre de la balle à l'aide d'un réseau neuronal convolutif (CNN) (voir figure 2.31).

Architecture du CNN proposé pour l'estimation de la taille de la balle dans le domaine image

La connaissance de la position de la balle à chaque instant sur les images est insuffisante pour estimer sa position 3D, car la distance entre la balle et la caméra est inconnue. Intuitivement, lorsque la balle se rapproche de la caméra dans la scène 3D, son diamètre dans les images augmente. Il diminue lorsqu'elle s'en éloigne. L'estimation de son diamètre en pixels dans le domaine image de la balle est alors intéressante, car elle fournit des informations sur la distance qui la sépare de la caméra. Étant donnée H la taille réelle

(physique) de la balle (soit 4.00 cm), h la taille apparente de la balle dans le domaine images (en pixels), f la distance focale, et $f_{dev} = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ la distance en pixels dans l'image entre le point principal et le centre de la balle, la distance caméra-balle D est calculée suivant l'homothétie :

$$D = (H \times \sqrt{f^2 + f_{dev}^2})/h \tag{2.5}$$

Connaissant la distance D, l'intersection entre la droite passant par le centre de la balle et le centre optique de la caméra permet de positionner la balle dans l'espace Monde 3D en utilisant la matrice de calibration extrinsèque comme présentée par la figure 2.30.

En raison de la vitesse élevée des balles de tennis de table et donc du flou de mouvement généré, la taille exacte de la balle en pixels est difficile à obtenir. Une petite erreur sur l'estimation de sa taille dans le domaine image entraîne une erreur conséquente sur la distance balle-caméra. L'estimation de la position de la balle dans l'espace 3D sera donc erronée.

Pour résoudre ce problème, nous avons conçu un CNN entraîné sur un ensemble de données synthétiques créées avec le logiciel Blender (présenté section 2.2.2). Un jeu de données généré présente de nombreux avantages : pour entraîner le réseau, nous pouvons générer autant de séquences que nécessaires, et la position 3D exacte de la balle est connue. De plus, la quantité de flou de déplacement peut être modifiée pour simuler différents temps d'exposition au moment des acquisitions.

VOEIKOV, FALALEEV et BAIKULOV, 2020 utilisent un CNN pour estimer de façon précise la position une balle de tennis de table dans une image, malgré le flou de mouvement observé. Ils obtiennent une détection de balle précise à 98,3% sur leurs images. Nous nous sommes inspirés de ce réseau `TTNet`, plus précisément de la détection locale de la balle pour estimer sa taille en pixels dans nos images. Contrairement à leur réseau, nous avons préalablement sélectionné une zone centrée sur la balle lors du processus de détection et suivi de balle. La taille de nos images d'entrées est également inférieure puisque nous avons sélectionné une zone carrée autour de la balle. L'auteur définit un `ConvBlock` comme la succession d'une couche convolutive de taille 3×3, une couche ReLU de régularisation, et une couche de `Pooling` de taille 2×2. Après ces différents `ConvBlock`, une couche de `Drop Out` est effectuée, puis le résultat est transformé en vecteur. L'auteur utilise ensuite plusieurs couches `Fully Connected` pour terminer par une sigmoïde

pour déterminer la position de la balle sur les axes x et y. Leurs images étant de taille 320×128, un total de sept `ConvBlock` est utilisé. Nos images étant plus petites, nous avons fait le choix de n'en utiliser que trois. Pour accélérer l'entraînement, et éviter que certains gradients ne soient nuls, nous avons également remplacé les couches de `ReLU` par des `Leaky ReLU` (Xu et al., 2015). Nous avons également limité notre réseau à deux couches `Fully Connected` puisqu'en utiliser trois n'améliorait pas nos résultats sur la base de validation.

La sigmoïde a également été supprimée puisque nous souhaitons effectuer une régression, et non une classification. Nous présentons l'architecture de notre réseau dans le Tableau 2.4. Celle-ci est représentée également par le schéma 2.32. Les couches jaunes représentent les couches de convolution, les couches rouges représentent les couches de `Pooling`, et les couches violettes représentent les couches `Fully Connected`.

Ce réseau utilise T images consécutives de taille 64×64 pixels, centrées sur la balle, et prédit en sortie la taille de la balle sur l'image centrale de cette fenêtre temporelle. La taille de la balle varie peu entre des images consécutives, et utiliser une fenêtre temporelle permet de régulariser l'estimation. Il faut noter qu'en raison de la précision requise sur le diamètre de la balle, nous utilisons les images en pleine résolution sans sous-échantillonnage spatial.

Comme précisé dans les sections 2.2.1 et 2.2.2, la vérité terrain correspondant à la taille de la balle en pixels a été obtenue à partir de la position 3D réelle et à l'aide de l'équation (2.5). Dans le cas idéal, il n'y a pas de flou de mouvement et le contour de la balle est donc net. La taille apparente correspond à la longueur du petit axe de l'ellipse lorsque l'image est générée avec le flou de mouvement, comme illustré sur la figure 2.24.

Augmentation de données

L'augmentation de données est fréquemment utilisée dans le domaine de l'apprentissage profond pour palier au manque de données d'entraînement. Cette technique consiste à générer de nouvelles données à partir des données existantes via différentes transformations. Dans notre cas, pour chaque vidéo et à chaque instant t, nous utilisons une fenêtre spatio-temporelle glissante de taille $64 \times 64 \times T$ en entrée du réseau. L'image de la balle, et donc le contenu du cube spatio-temporel, dépend à la fois de sa vitesse et de sa position dans la scène. Deux types d'augmentation de données peuvent être

TABLE 2.4 – Tableau des différentes couches du réseau pour estimer la taille de la balle en pixels avec $T = 5$

Taille d'entrée	Opérateur	Canaux d'entrée/Canaux de sortie
64×64	Conv 3×3	5/32
64×64	Leaky Relu	32/32
64×64	MaxPool 2×2	32/32
32×32	Conv 3×3	32/64
32×32	Leaky Relu	64/64
32×32	MaxPool 2×2	64/64
16×16	Conv 3×3	64/128
16×16	Leaky Relu	128/128
16×16	MaxPool 2×2	128/128
8×8	Flatten	128/-
8192	FullyConnected	-
2048	Relu	-
2048	Dropout	-
2048	FullyConnected	-

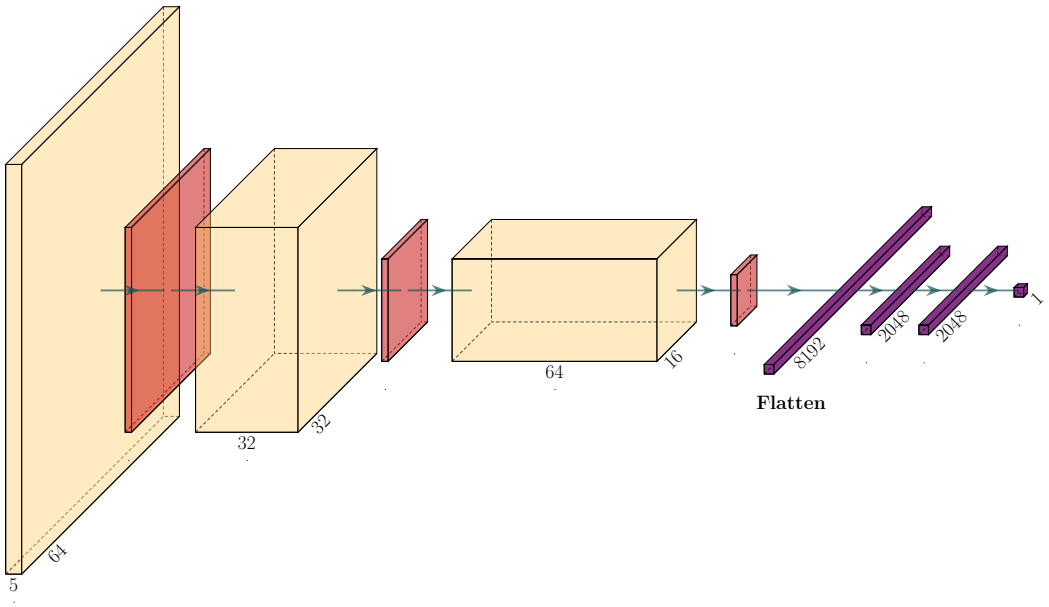


FIGURE 2.32 – Architecture du réseau pour estimer la taille de la balle en pixels pour $T = 5$

envisagés selon la dimension à laquelle elle est appliquée, soit temporelle ou bien spatiale.

Aucune augmentation n'est réalisée temporellement, puisque la variation de la vitesse de translation de la balle suffit à modifier la distance séparant le centre de la balle entre deux images consécutives. Modifier l'échantillonnage

temporel revient donc à modifier la distance caméra-balle que nous cherchons précisément à estimer.

Les seules possibilités restent donc les transformations agissant dans le domaine spatial de l'image. Une augmentation spatiale fréquemment utilisée est l'homothétie : agrandir ou réduire la taille d'une image est parfois utilisé lorsque l'analyse doit être indépendante de la taille de l'objet d'intérêt. Ceci n'est pas valable dans notre cas d'application, puisque la taille réelle de la balle est connue et qu'une modification de sa taille dans l'image entraînerait une erreur d'estimation sur la profondeur. Deux types d'augmentations spatiales sont cependant réalisées. La première est l'utilisation de filtres Gaussiens de taille 5×5 pour flouter les images et agir comme un effet de défocalisation. La seconde modification effectuée est la rotation des images. Cette rotation est appliquée de façon aléatoire dans un intervalle de [-15°,+15°]. En effet, l'estimation de la taille de balle doit être indépendante du niveau de flou de déplacement, du flou de focalisation, et de la direction de son vecteur de translation.

Méthodes d'entraînements

Afin d'évaluer notre approche sur le jeu de données `avec sportifs` (composé d'extraits de séquences *Gammes* et *Gammes Alternées*), contenant un nombre limité de séquences, nous commençons par entraîner notre réseau sur le jeu de données `synthétiques`. Tous les modèles ont été entraînés à l'aide du framework PyTorch (PASZKE et al., 2019), sur un GPU NVIDIA 1070 GTX et 48 Giga de RAM.

Comme le but est d'estimer la taille de la balle en pixels par une méthode de régression utilisant un CNN, l'erreur choisie est l'erreur quadratique. Il s'agit de la fonction de perte la plus couramment utilisée pour la régression.

Pour le jeu de données `synthétiques`, contenant 200 vidéos, celui-ci est séparé en trois sous-ensembles : *Entraînement*, contenant 140 vidéos, *Validation* contenant 30 vidéos, utilisé ajuster les hyperparamètres (nombre de couches, coefficient d'apprentissages, taille des batches) et *Test* contenant 30 vidéos. Le jeu de données étant conséquent, des batches de 250 images sont utilisés, et ce sur 50 époques. Le jeu de données contenant 60 676 images, des batchs de grande tailles permettent d'accélérer la durée d'entraînement (RADIUK, 2017) sans pour autant réduire la précision de notre modèle. Pour faire face à la taille importante des batches, nous avons choisi un coefficient d'apprentissage de 0,0001. Un coefficient d'apprentissage plus petit,

par exemple 0,00001, nécessiterait un nombre d'époques plus important et convergerait plus lentement.

Le jeu de données `avec sportifs` est séparé en deux sous-ensembles : *Entraînement* contenant 24 vidéos, et *Test* contenant 9 vidéos. Le choix du modèle et la taille de la fenêtre glissante ont été validés préalablement sur le jeu de données `synthétiques`. La taille de ce jeu de données étant restreint, nous souhaitons donc utiliser le plus de séquences possibles pour l'entraînement et le test, et avons fait le choix de ne pas utiliser de jeu de validation sur cet ensemble de données.

La répartition est faite de manière à ce que la base *Test* contiennent 3 séquences de `Top Spin`, 3 séquences de `Contre-Attaque` et 3 `Poussettes`. Le coefficient d'apprentissage est laissé à 0,001, mais le nombre d'images dans le batch est réduit à 100 pour accélérer la vitesse de convergence sur ce jeu de donnée plus restreint. Nous utilisons 50 époques. Pour compenser la faible quantité de données, nous utilisons la méthode d'apprentissage par transfert (*transfer learning*) : les poids sont initialisés à partir du meilleur score obtenu sur les données synthétiques, puis l'entraînement est poursuivi sur le jeu de données `avec sportifs`.

Méthodes d'évaluation

Pour le jeu de données `synthétiques`, nous disposons des positions 3D générées des balles, ainsi que de la taille théorique de la balle permettant la reconstruction 3D obtenue à partir de l'équation 2.5. Pour ce jeu de données, la métrique d'évaluation sera donc la distance au sens des moindres carrés entre la taille de la balle estimée en pixels par le CNN et celle de la vérité terrain. Nous présentons l'erreur globale ainsi que les erreurs sur chacun des trois classes de coups.

Différentes tailles de fenêtres temporelles T ont été testées, leur but étant de régulariser l'estimation de la taille de balle.

Pour le jeu de données `avec sportifs`, nous n'avons pas la vérité terrain de la position en utilisant une seule caméra. Cette position peut être estimée à l'aide de deux caméras (voir reconstruction stéréo figure 2.18). L'utilisation de deux caméras est nécessaire pour obtenir la vérité terrain des positions 3D de balle, mais ne servira que pour la phase d'entraînement. Pour les trois classes de coups choisies, nous avons supposé que la balle se déplace dans un plan entre deux impacts de raquette. Les positions 3D obtenues à partir de la stéréo-vision sont régularisés préalablement en utilisant une régression planaire minimiser les erreurs lors de l'entraînement du réseau.

La métrique d'évaluation que nous utilisons pour l'entraînement sera également la même que pour le jeu de données synthétiques, c'est-à-dire l'erreur au sens des moindres carrés entre l'estimation de la taille de balle en pixels et sa taille réelle. En utilisant cette taille de balle, nous présentons également l'erreur sur les positions 3D résultant de sa rétroprojection. La distance euclidienne sera utilisée pour calculer cette erreur.

2.4 Expérimentations et résultats

2.4.1 Estimation de la taille de balle sur le jeu de données `synthétiques`

À notre fréquence d'acquisition (240 fps), la taille de la balle et la distance caméra-balle varient peu entre deux images consécutives. Différentes tailles de fenêtre glissante sont considérées pour l'entraînement du réseau ($T = 1$, $T = 3$, $T = 5$ et $T = 7$). L'entrée du réseau est un ensemble de T images consécutives de taille 64×64, centré temporellement sur la frame $(T+1)/2$, et centré spatialement sur le centre de la balle.

La figure 2.33 représente la courbe d'apprentissage correspondant à l'erreur en pixels au cours des itérations pour la base d'entraînement et la base de validation (pour $T = 5$). Nous observons une convergence à la fois sur l'ensemble d'apprentissage, l'ensemble de validation et l'ensemble de test. L'erreur obtenue sur ces trois sous-ensembles décroît vers zéro, indiquant une bonne précision de notre modèle. Nous obtenons une erreur sur le jeu de validation et de test similaire à celui de la base d'entraînement. Nous n'avons donc pas de surapprentissage, traduisant une bonne généralisation du modèle pour estimer la taille de la balle. Nous avons limité le nombre d'époques à 50 puisque la différence entre les résultats sur la base d'entraînement et la base de test sont minimes au bout de 50 itérations.

Nous observons une convergence rapide du modèle, qui peut s'expliquer en partie par les bornes sur la taille de balle observée. La taille de balle minimale observée est aux alentours de 9 pixels, et maximale aux alentours de 17 pixels, ce qui facilite l'estimation sur la taille. La difficulté vient du fait de prédire les séquences où la balle se déplace très rapidement, c'est-à-dire les coups de type `Top Spin`, où le flou de mouvement est le plus important. Les frontières de la balle sont floues, rendant plus difficile une estimation précise de la taille de la balle, et nécessitant plus d'itérations.

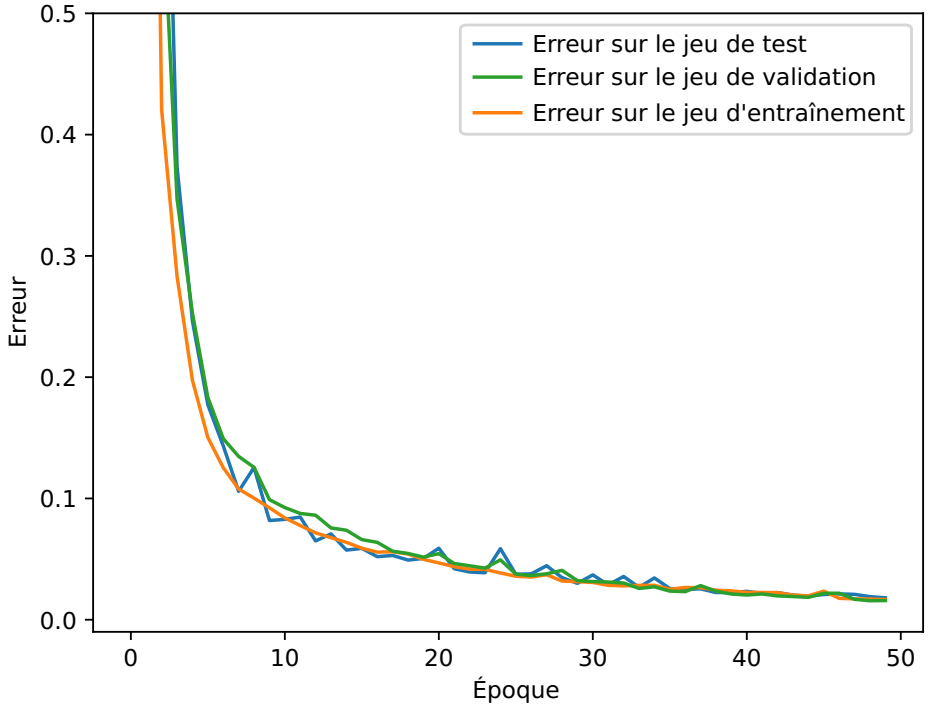


FIGURE 2.33 – Courbes d'apprentissage montrant l'erreur d'estimation de la taille de la balle pour les ensembles d'apprentissage et de validation du jeu de données `synthétiques` ($T = 5$)

Le tableau 2.5 récapitule les différentes erreurs obtenues lors de l'apprentissage pour chacune des tailles de fenêtre glissante choisies. Le meilleur résultat est une erreur de $0,041$ pixels obtenu lorsque $T = 5$.

Au moment de l'impact entre la balle et la raquette, la taille moyenne de la balle est d'environ 11 pixels. L'erreur sur la taille de balle estimée pour $T = 5$ la précision de la distance entre la caméra et la balle est de l'ordre de $99,6\%$.

Les vitesses de translation ayant un effet direct sur le flou de mouvement observé, nous présentons dans le tableau 2.6 le récapitulatif des erreurs moyennes obtenues pour chaque type de coup ($T = 5$). Nous présentons également la répartition des erreurs d'estimation de taille sous forme de graphique en violon figure 2.34.

Dans les graphiques en violon, la largeur horizontale représente la fréquence des observations, le point blanc représente la médiane, le rectangle noir représente les frontières des premiers et troisièmes quartiles.

Bien que le flou de mouvement soit plus important sur les `Top Spin` que les autres coups, nous obtenons une erreur moyenne de 0,049 pixel sur les

TABLE 2.5 – Comparaison d'estimations de taille de balle pour différentes fenêtre temporelles

Taille de la fenêtre temporelle	Erreur sur la taille (en pixels)
1	0,086
3	0,051
5	0,041
7	0,088

TABLE 2.6 – Comparaison d'estimation de taille de balle pour chacun des types de coup avec $T = 5$

Coup	Erreur sur la taille (en pixels)
Poussette	0,044
Contre-Attaque	0,031
Top Spin	0,049

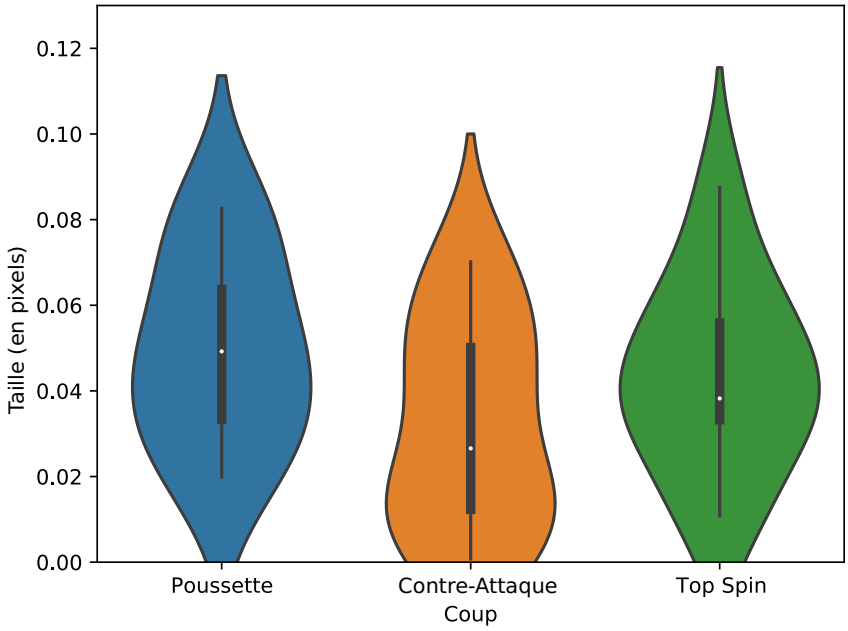


FIGURE 2.34 – Graphique en violon des erreurs d'estimation de taille de balle pour chaque type de coup, avec $T = 5$

`Top Spin`, proche de l'erreur obtenue sur les `Poussettes` qui est de 0,044 pixel. Le coup dont l'estimation de la taille de balle est la plus précise est

la Contre-Attaque, avec une erreur de 0,031 pixel. Sur le graphique en violon, nous observons que la médiane des erreurs (point représenté en blanc) est de 0,05 pixel pour les Poussette, contre 0,04 pour les Top Spin.

La répartition des erreurs entre les Poussettes et Top Spin, représentée par la forme globale et la largeur horizontale des graphiques en violons, est assez similaire entre les deux coups. Peu de séquences ont une erreurs moyenne inférieure à 0,02 pixel (largeur horizontale faible). Au contraire, pour les Contre-Attaque, nous observons une médiane proche de 0,25, et de nombreux coups prédits avec une erreur moyenne en dessous de 0,03 pixels.

La raison de cette meilleure prédiction pour les Contre-Attaque est la variété du jeu de données. Celui-ci contient un mélange équilibré de Poussettes, avec peu de flou de mouvement, de Contre-Attaques avec un flou de mouvement moyen, et de Top Spin avec un flou de déplacement élevé. Nous pensons qu'un réseau estimant avec précision la taille des balles lors d'un Top Spin aurait du mal à prédire une balle sans flou de mouvement, et inversement. La Contre-Attaque, ayant un flou de mouvement modéré est donc plus correctement prédite que les autres coups.

Pour chacun de ces coups, nous présentons figure 2.35 un exemple représentatif de l'estimation de taille de balle, ainsi que la rétroprojection associée. Pour chacun de ces coups, nous observons une oscillation dans l'estimation de la taille de la balle par rapport à la vérité terrain. Cette oscillation entraîne une légère erreur de l'estimation de la distance caméra-balle.

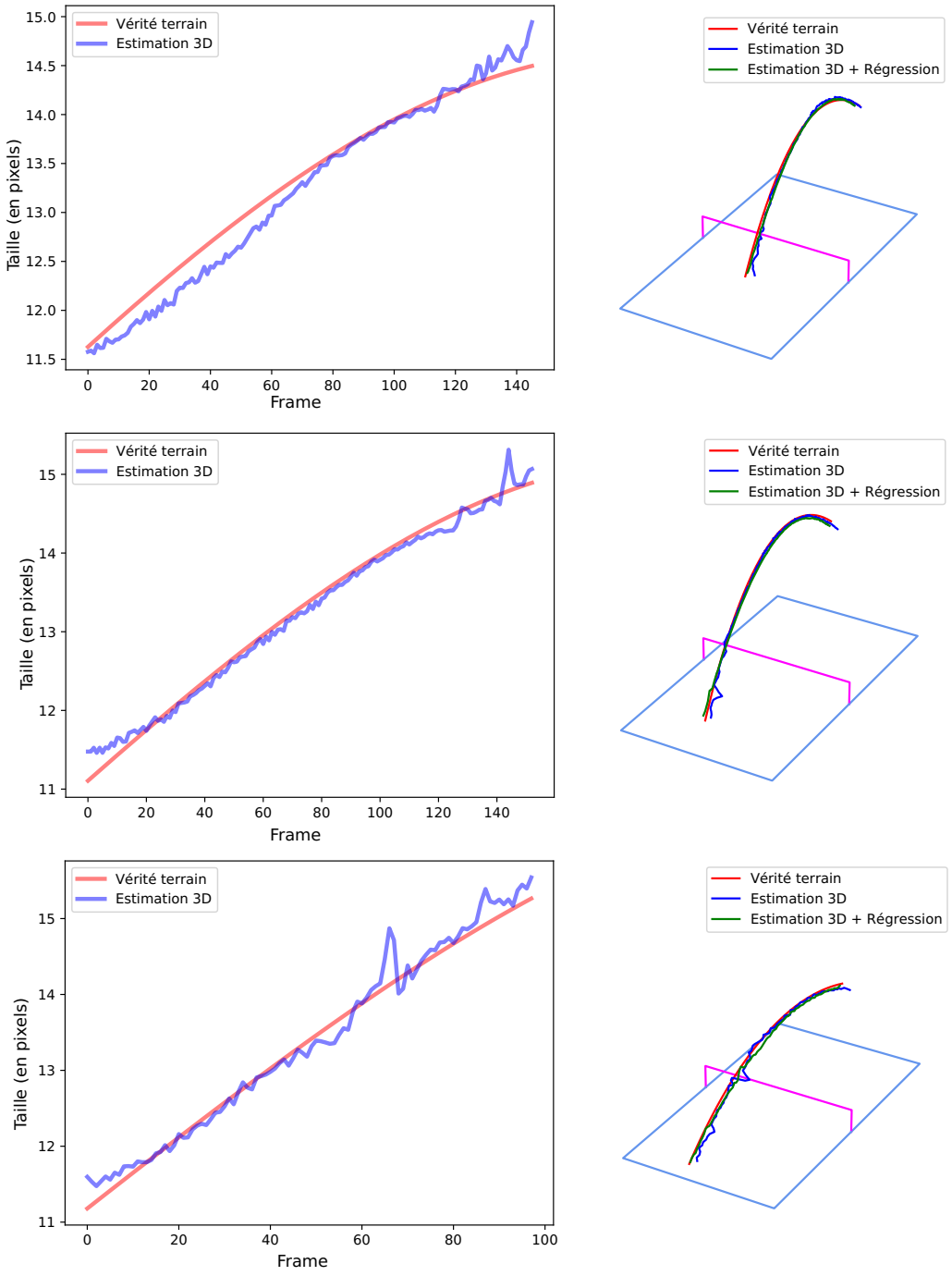


FIGURE 2.35 – Représentation des résultats de reconstruction 3D sur trois coups : `Poussette` en haut, `Contre-Attaque` au centre, `Top Spin` en bas. À gauche : Évolution de la taille de la balle au cours du temps, et taille estimée. À droite : Rétroprojection de la balle associée

2.4.2 Apprentissage par transfert sur le jeu de données avec sportifs

Après avoir entraîné le réseau sur le jeu de données synthétiques, avec une fenêtre glissante de taille $T = 5$, nous effectuons un apprentissage par transfert sur le jeu de données avec sportifs. L'objectif est de faire de l'adaptation de domaine pour obtenir de bonnes performances sur le jeu de données avec sportifs de petite taille. Parmi les différentes formes d'apprentissage par transfert, nous utilisons une approche dite de Fine-tuning. Comme son nom l'indique, les poids du modèle entraîné sur un autre jeu de données (dans notre cas synthétique) sont conservés. Ces poids servent de point de départ pour le réseau, et entraînent une convergence rapide du modèle sur le nouveau jeu de données (dans notre cas avec sportifs), notamment comparé à une initialisation des paramètres aléatoire. Aucune couche n'est gelée, c'est-à-dire que tous les poids du réseau sont mis à jour lors de l'entraînement sur le nouveau jeu de données. Le coefficient d'apprentissage est fixé à 0,0001, et les batches sont de taille 250 comme celui pour le jeu de données synthétiques.

La figure 2.36 représente la courbe d'apprentissage pour le jeu de données avec sportifs, sur l'erreur en pixels au cours des itérations pour la base d'entraînement (24 séquences) et la base de test (9 séquences).

Le réseau étant pré-entraîné, celui-ci converge rapidement vers une erreur de 0,32 sur la base d'entraînement, malgré un jeu de données de petite taille comparé au jeu de données synthétiques. Nous n'avions pas utilisé de jeu de validation pour utiliser un maximum de séquences pour l'entraînement, mais nous observons cependant une erreur différente entre le jeu de test et le jeu d'entraînement. L'erreur sur la base de test continue de décroître au cours des itérations, et converger progressivement vers une erreur de 0,49. À partir de 30 époques, nous n'observons pas de différence notoire et pourrions donc arrêter l'entraînement plus tôt. De plus, nous observons une augmentation de l'erreur entre les données d'entraînement et d'apprentissage. Notre principale explication de ce sur-apprentissage est la faible quantité d'images présentes dans la base de données d'entraînement. Augmenter le nombre de séquences pourrait être une première approche pour diminuer l'erreur sur le jeu de test. En raison des différences de flou de mouvement induites par la vitesse qui donc diffère selon les coups, les résultats pour chaque type de coups sont présentés séparément dans le tableau 2.7. L'erreur moyenne de distance

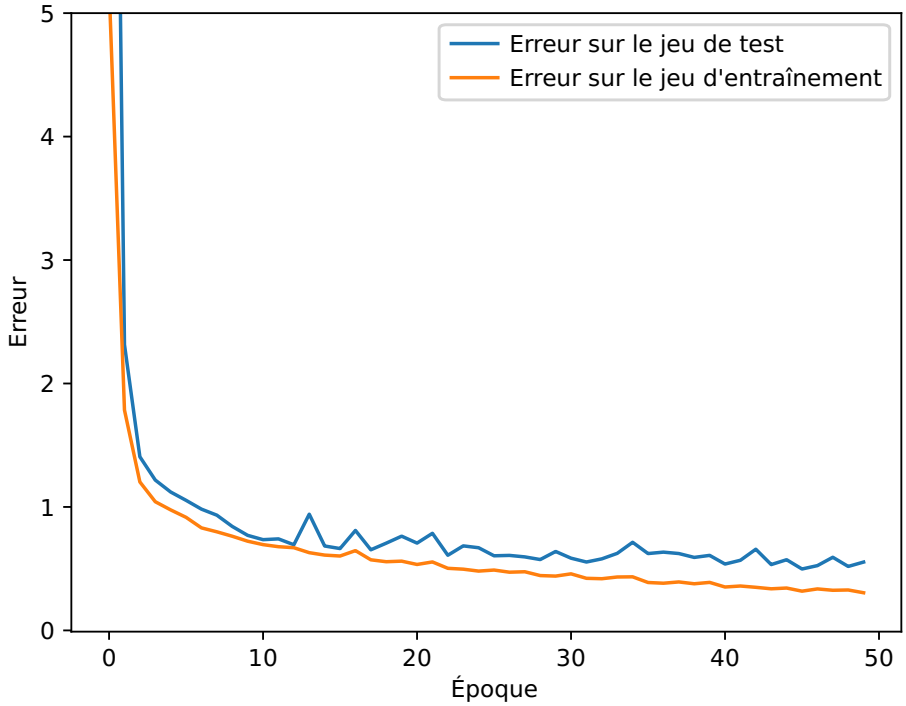


FIGURE 2.36 – Courbes d'apprentissage montrant l'erreur d'estimation de la taille de la balle pour les ensembles d'apprentissage et de validation du jeu de données `avec sportifs` ($T = 5$)

entre la caméra et la balle entre la position 3D estimée et la vérité terrain augmente lorsque la vitesse de la balle est plus élevée (erreur moyenne de 1, 33 % pour le `Top Spin` correspondant à une erreur sur la distance euclidienne de rétroprojection de 6, 43 *cm* entre l'estimation et la vérité terrain, contre 1, 10 % pour une `Poussette` correspondant à une erreur sur la distance euclidienne de retroprojection de 5, 27 *cm*).

TABLE 2.7 – Précision relative moyenne sur l'estimation de la distance entre la caméra et la balle

Type de coup	Précision en sortie du CNN (en %)	Précision après régression (en %)
Top Spin	98,67	99,10
Contre-Attaques	98,78	99,19
Poussette	98,90	99,26

Comme pour le jeu de données `synthétiques`, la valeur de la taille estimée de la balle oscille autour de sa taille correcte, ce qui entraîne des erreurs

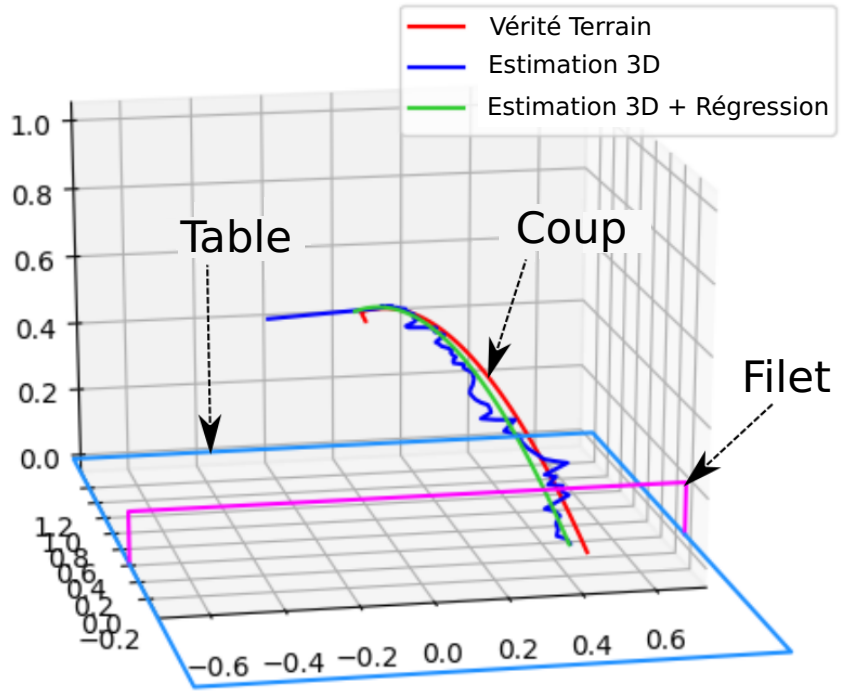


FIGURE 2.37 – Comparaison entre la position 3D d'une balle obtenue par triangulation, la position estimée avec le CNN, et la position estimée après régression planaire.

dans la position 3D estimée (voir figure 2.37). Comme on suppose que la balle a une trajectoire qui se situe dans un plan, une régression planaire est utilisée pour projeter ces points 3D sur un plan 2D. Les positions lissées de la balle qui en résultent sont situées à l'intersection entre le plan de régression et le rayon allant de la caméra aux points 3D. Cela conduit à une trajectoire sur laquelle l'erreur moyenne de la distance entre la caméra et la balle pour un `Top Spin` passe de $1,33\%$ à $0,90\%$. Cette erreur représente une erreur de rétroprojection de $4,32\ cm$.

2.5 Conclusion

Dans ce chapitre, nous avons présenté dans la section 2.1 les différentes étapes essentielles à la reconstruction de séquences de positions 3D à partir d'une vidéo : calibration, détection et suivi de balle dans la séquence d'images.

Après une présentation des deux jeux de données que nous avons créés (voir section 2.2), un jeu de données `avec sportifs` ainsi qu'un jeu de données `synthétiques`, nous avons présenté section 2.3 une technique de reconstruction de trajectoires 3D par apprentissage. Avec un flou de mouvement important, nous avons utilisé un premier jeu de données `synthétique` pour entraîner un réseau convolutif et estimer la taille de la balle en pixels, puis effectué un apprentissage par transfert sur le jeu de données `avec sportifs`.

Dans la section 2.4, nous obtenons une erreur d'estimation de taille de balle de 0,041 pixels, ce qui correspond à une précision relative de 99,6 % sur la distance caméra-balle sur un jeu de données `synthétiques`. En effectuant un apprentissage par transfert, nous avons montré la capacité de notre approche à s'adapter à un jeu de donnée réel `avec sportifs` de petite taille. Nous obtenons une précision de rétroprojection 3D de balle moyenne de 98,89 % après avoir projeté les positions successives de la balle dans un plan, malgré un flou de mouvement important.

Bien que précise, l'estimation de la distance entre la caméra et la balle est effectuée pour chaque image, et nécessite une détection et un suivi robuste.

Parmi les 33 séquences de notre jeu de données `avec sportifs`, 7 d'entre-elles n'étaient pas suivies en intégralité, et ont nécessité une annotation manuelle (ré-initialisation du tracker). La principale cause de la perte de suivi de la balle est le changement brusque de direction, et le flou de mouvement lors d'un impact avec la table ou avec la raquette. Le fort flou de mouvement, combiné à une obstruction partielle ou complète de la balle par la raquette rend difficile le suivi de la balle. La seconde raison de la perte du suivi de la balle est l'apparence du joueur. Après impact, la balle est perçue avec un flou de mouvement important, et lorsque celle-ci passe par-dessus une zone de couleur chair, le tracker choisi continue parfois de suivre la main du joueur. Une amélioration du suivi de la balle fait donc partie des perspectives futures.

Nous avons fait l'hypothèse que la balle se déplaçait dans un plan pour régulariser nos résultats. Cependant, pour certains types de coups (notamment des services), la balle peut avoir une rotation latérale, et donc un effet

Magnus non planaire. Si nous souhaitons par la suite étendre le nombre de classes étudiées, nous devons ainsi prendre en compte les possibles effets latéraux. La régression planaire pourrait être remplacée par une projection sur une surface plus générale pour prendre en compte ces variations, ou des approches globales pour rétroprojecter l'ensemble des positions 3D sans utiliser une approche itérative pourraient être considérées.

Le chapitre suivant introduit la deuxième partie de ces travaux, c'est-à-dire l'extraction des paramètres cinématiques d'une balle en mouvement en vision monoculaire. Nous allons passer d'une suite de points dans le domaine réel à une trajectoire physique. Nous nous concentrons sur l'étude de la vitesse de rotation de la balle sur elle-même. Ces trajectoires sont aussi affectées par d'autres forces comme la gravité, et les frottements de l'air (effet de traînée). Le passage des points 3D à une trajectoire permettra d'extraire des informations pour précisément caractériser un coup.

Chapitre 3

Modélisation de trajectoires et extraction de paramètres cinématiques

Sommaire

3.1 Introduction

L'objectif de ce chapitre est d'extraire d'une séquence de points 3D précédemment estimée (Chapitre 2) une trajectoire "*Physique*", obtenue à partir d'équations du mouvement de la balle. Il est ainsi par la suite possible d'étudier les différents paramètres cinématiques qui régissent la trajectoire d'une balle. Ces paramètres cinématiques, tels que les vitesses de translation ou de rotation de la balle permettent de caractériser et de quantifier la performance d'un coup effectué par un joueur.

La trajectoire d'une balle au tennis de table est différente de celle d'autres sports de balle. Il y a plusieurs raisons à cela. Dans des sports comme le volleyball, ou basketball, la balle est uniquement propulsée avec les mains des joueurs. La trajectoire est donc influencée par la vitesse initiale donnée par le joueur, ainsi que la gravité et le frottement de l'air. Dans ce cas, et à l'issue du bilan des forces, la trajectoire estimée de la balle est parabolique (H. T. CHEN et al., 2011). Pour des sports comme le tennis (MEHTA et PALLIS, 2001), le baseball (NATHAN et al., 2006), le golf (Jing LI, TSUBOKURA et TSUNODA, 2017), la balle est frappée avec une interface (dans le cas du tennis de table, l'interface est la raquette). Un angle et une vitesse sont généralement donnés à l'interface lors de la frappe. Le frottement au moment de l'impact entraîne un effet de rotation de la balle sur elle-même. Ainsi, une balle de baseball peut atteindre plus de 2 600 tours par minute. Au tennis, Rafael Nadal a la rotation moyenne la plus élevée en coup droit avec 3 200 tours par minute [1]. À titre de comparaison, au tennis de table, le record de rotation est de 8 000 tours par minute pour une balle qui fait 40 mm de diamètre [2].

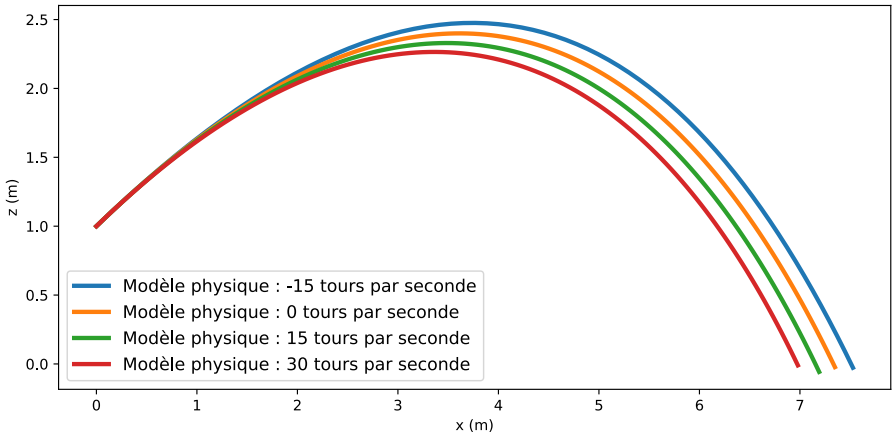


FIGURE 3.1 – Effets de la rotation d'une balle de tennis de table sur sa trajectoire

1. https://olympic.ca/2017/10/11/getting-the-spins-highest-revolutions-in-sport/
2. La modification progressive des matériaux de la raquette (colle, éponge ...) a entraîné une augmentation régulière de la vitesse de la balle, rendant de plus en plus difficile pour le spectateur la visualisation des trajectoires de balle lors de diffusions de match. En octobre 2000, la taille du diamètre de la balle a alors été augmentée, passant de 38 mm à 40 mm. Cet léger accroissement de taille a entraîné une augmentation des frottements avec l'air, diminuant sa vitesse de translation. Cette baisse de vitesse de translation a également eu comme effet de diminuer la vitesse de rotation de la balle sur elle-même.

Cette rotation va générer une force dite *Magnus* (décrite en section 3.2.2), qui impactera la trajectoire de la balle (MIYAZAKI et al., 2017 ; SCHNEIDER et al., 2018). Nous illustrons l'effet de cette force sur la figure 3.1 avec différentes trajectoires de balles au tennis de table ayant le même vecteur mouvement initial, mais avec une vitesse de rotation différente. Une forte rotation entraîne un effet dit `Lifté` (la balle plonge plus vite et accélère) alors qu'une rotation de sens opposé entraîne un effet dit `Coupé`, comme la courbe bleue.

Les principales contributions de ce chapitre sont :

— La génération de séquences synthétiques de trajectoires de balles en utilisant un modèle physique.

— L'extraction de paramètres cinématiques (vitesses de rotation et de translation de la balle) à partir de positions 3D et d'un modèle physique. Ces paramètres permettront la caractérisation et l'analyse des coups d'un joueur.

— La classification des coups de Tennis de Table à l'aide de ces paramètres cinématiques extraits.

Ces travaux ont donné lieu à des publications dans :

— une conférence nationale (CALANDRE, PÉTERI, MASCARILLA et TREMBLAIS, 2020b) et une conférence internationale (CALANDRE, PÉTERI, MASCARILLA et TREMBLAIS, 2020a) avec comité de lecture pour l'extraction de paramètres cinématiques en utilisant un modèle physique discret et une recherche par grille,

— une conférence internationale avec comité de lecture après l'acquisition de nouvelles séquences avec les sportifs du club Pongiste Rochelais (CALANDRE, PÉTERI, MASCARILLA et TREMBLAIS, 2021). Nous utilisons dans ces travaux une équation différentielle ordinaire, permettant d'améliorer le temps de traitement.

Dans la suite de ce chapitre, nous nous concentrons uniquement sur les balles à surfaces lisses, les balles avec une surface rugueuse ne seront pas traitées. Des travaux spécifiques existent pour l'étude de trajectoires de balles ayant des coutures, comme au baseball (NATHAN et al., 2006) ou une surface sculptée, comme au golf (SMITS et SMITH, 2021). Dans ce chapitre, nous analysons dans un premier temps les sections de trajectoire entre l'impact avec la raquette et l'impact sur la table (Fig. 3.6). Nous intégrons le rebond sur la table dans le chapitre 4.

Suite à l'extraction des positions 3D successives des balles présentée au chapitre 2, nous présentons en section 3.2, l'estimation de trajectoires de balles

à partir de cet ensemble séquentiel de points. Les méthodes que nous présentons sont l'interpolation polynomiale et l'utilisation d'un modèle physique.

Dans un second temps, nous présentons en section 3.3 le jeu de données `synthétiques` utilisé pour nos expérimentations, ainsi que la validation du modèle physique. Il s'agit d'une extension de ce qui a été présenté en section 2.2.2, dans laquelle nous nous étions focalisés sur la position de la balle et sur la création de la scène. Nous détaillerons ici la génération de trajectoires à l'aide du modèle physique introduit dans la section 3.2.2.

Enfin, nous présentons dans la section 3.4 notre approche pour extraire les paramètres cinématiques que sont la vitesse de translation, et la vitesse de rotation de la balle. Nous minimisons la distance entre les positions 3D obtenues au chapitre 2, et les positions obtenues à l'aide d'une trajectoire simulée utilisant ce modèle physique. Nous évaluons notre approche sur deux jeux de données. Sur le jeu de données `synthétiques`, nous évaluons la précision des paramètres cinématiques obtenus. Sur des séquences `avec sportifs`, sans vérité terrain sur les paramètres physiques de balle, les paramètres physiques seront extraits et utilisés pour classifier trois types de coups.

Nous concluons le chapitre dans la section 3.5

3.2 Estimation de trajectoires de balles à partir de leurs positions 3D

Dans cette section, nous présentons deux approches utilisées pour la reconstruction de trajectoires de balles au tennis de table à partir de leur séquence de positions 3D.

La première méthode consiste à faire une interpolation polynomiale sur les positions 3D de balle. Bien que rapide, cette méthode ne permet pas d'obtenir un modèle physique, et donc d'en déduire des indicateurs de performance sur une frappe.

La seconde est l'utilisation d'un modèle physique de trajectoire de balle. Ce modèle intègre la vitesse de translation de la balle, sa vitesse de rotation ainsi que les forces de gravité et les frottements de l'air.

3.2.1 Approximation des positions 3D par régression polynomiale

Afin d'obtenir un polynôme s'approchant au mieux des données observées, il est possible de faire une régression polynomiale sur un ensemble de points 3D. Les régressions polynomiales ont l'avantage d'être très rapides à estimer, ce qui est intéressant dans le cas où le temps réel est un facteur important. Un autre avantage est que les trajectoires obtenues peuvent interpoler le nuage de points avec une grande précision si le degré du polynôme est suffisamment élevé. Sur la figure 3.2, la trajectoire de balle avec un modèle physique et un polynôme d'ordre 4 est comparée, avec dans ce cas particulier, une très bonne précision sur l'estimation obtenue. Dans LIN, YU et Y. C. HUANG, 2020, les auteurs utilisent un polynôme d'ordre 1 pour les axes `X` et `Y`, et un polynôme d'ordre 4 pour l'axe vertical `Z`. Les auteurs obtiennent une précision de l'ordre de 99,03% pour l'axe `X` et `Y`, et 99,7% pour l'axe `Z`. Les estimations 3D étant souvent sensibles aux bruits, cette approche peut également être utilisée pour lisser les positions successives estimées.

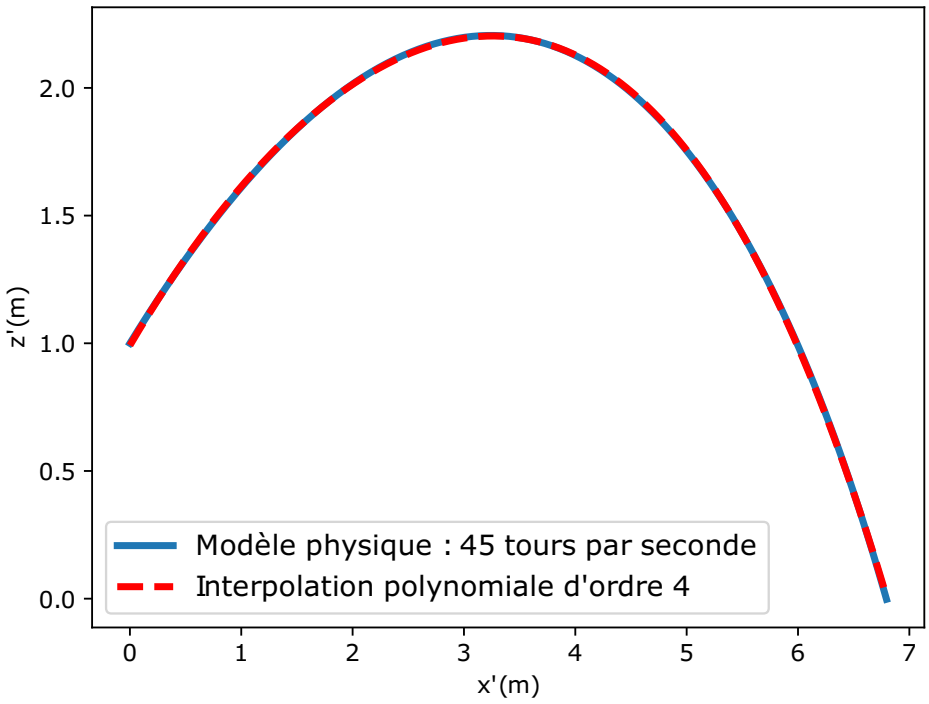


FIGURE 3.2 – Comparaison en 2D entre une trajectoire avec un modèle physique, et la régression polynomiale associée

Cependant, cette approche n'est pas liée à un modèle physique, et n'a que peu d'utilité pour notre objectif d'extraction d'indicateurs de performance. Nous avons donc privilégié l'utilisation d'un modèle dynamique pour estimer les trajectoires de balle.

3.2.2 Estimation de la trajectoire via un modèle dynamique

Dans cette section, une méthode utilisant un modèle physique pour inférer la trajectoire de la balle à partir de position 3D est présentée. La trajectoire de la balle est supposée s'effectuer dans un plan 2D, et les forces exercées sur une balle en mouvement sont la gravité, l'effet de traînée, et l'effet Magnus (ou *lift*) lié à la rotation de la balle sur elle-même.

Bilan des forces exercées

D'après le Principe Fondamental de la Dynamique de Newton appliqué à la balle pendant son mouvement, la dérivée temporelle de la quantité de mouvement $\mathbf{P}$ est égale à la somme des forces extérieures qui s'exercent sur le solide :

$$\frac{d\mathbf{P}}{dt} = m\frac{d\mathbf{V}}{dt} = \mathbf{F}_G + \mathbf{F}_A(\mathbf{V}) \tag{3.1}$$

où m est ici la masse (constante) de la balle ($27g$), $\mathbf{V}$ son vecteur vitesse, $\mathbf{F}_G = -m\mathbf{g}$ avec $\mathbf{g}$ l'accélération de la pesanteur, et $\mathbf{F}_A(\mathbf{V})$ la force aérodynamique. Sans la force aérodynamique, la trajectoire serait un arc de parabole. En la prenant en compte, la forme de la parabole est modifiée par l'effet de traînée $\mathbf{F}_D$ et par l'effet Magnus $\mathbf{F}_L$ (BRIGGS, 1959).

$$\mathbf{F}_A(\mathbf{V}) = \mathbf{F}_D(\mathbf{V}) + \mathbf{F}_L(\mathbf{V}) \tag{3.2}$$

La première force aérodynamique est la force de traînée $\mathbf{F}_D$. Elle agit comme une force de friction de direction opposée à celle de la trajectoire. Si on note C_D le coefficient de frottement, ρ la densité de l'air ($1,2kg/m^3$), $A = \pi.r^2$ la surface de frottement avec l'air pour une balle de rayon r ($2cm$), et $\mathbf{V}$ le vecteur vitesse de norme V et d'angle θ (dans le repère lié au centre de masse de la balle), la force de traînée $\mathbf{F}_D$ s'écrit :

$$\mathbf{F}_D(\mathbf{V}) = -\frac{1}{2}C_D\rho AV\mathbf{V} \tag{3.3}$$

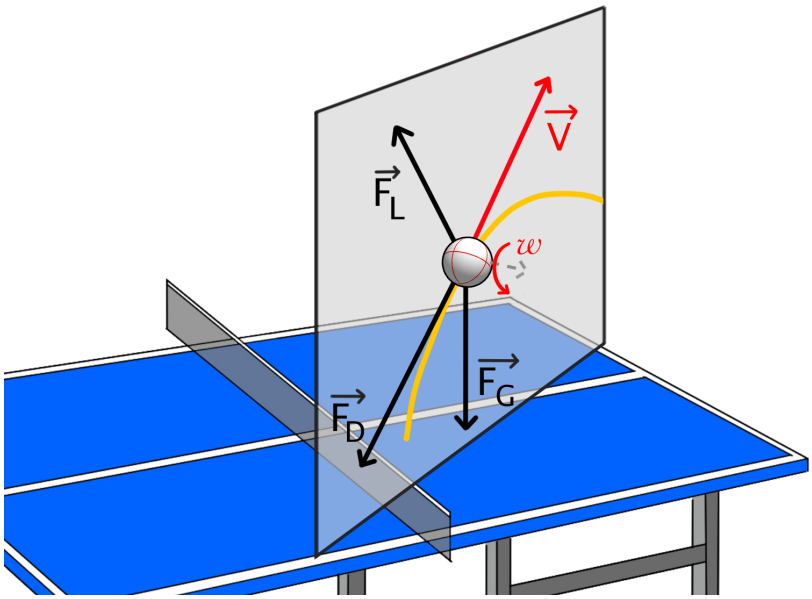


FIGURE 3.3 – Forces exercées sur la balle et sa vitesse de translation. Le déplacement est considéré dans un plan

La seconde force aérodynamique est l'effet Magnus $\mathbf{F}_L$. Lorsque la balle est en mouvement et tourne sur elle-même, la variation de pression de l'air entre la partie supérieure et la partie inférieure de la balle entraîne une altération de sa trajectoire, illustrée par la figure 3.4.

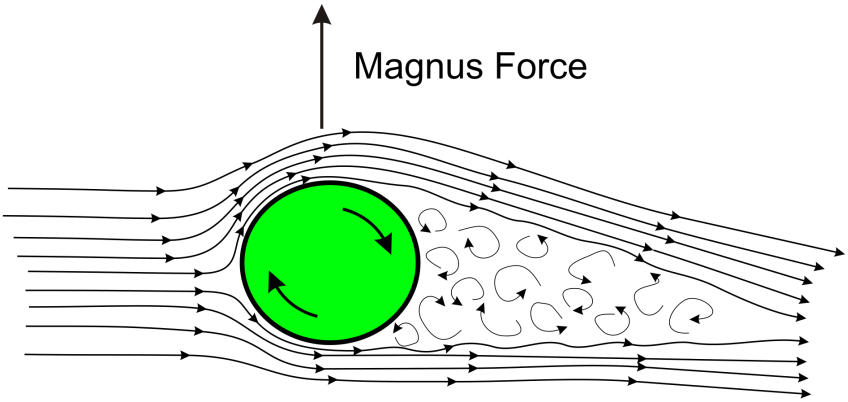


FIGURE 3.4 – Représentation du flux d'air lié à la rotation d'une balle https://en.wikipedia.org/wiki/Magnus_effect

En notant $\boldsymbol{\omega}$ le vecteur de vitesse de rotation de la balle, de norme ω, et S_0 le paramètre de lift, l'effet Magnus peut s'écrire comme :

$$\mathbf{F}_L(\mathbf{V}) = S_0 \boldsymbol{\omega} \wedge \mathbf{V} \tag{3.4}$$

Cette force, orthogonale au vecteur vitesse, est à l'origine des effets, coupés ou liftés, qui sont essentiels en tennis de table. Comme KUSUBORI, YOSHIDA et SEKIYA, 2012; SHEN et al., 2016, nous ferons l'hypothèse que pour tous types de coups, la balle se déplace dans un plan entre deux frappes. Le vecteur de vitesse angulaire $\boldsymbol{\omega}$ est orthogonal au plan $x - z$. En notant le coefficient de lift C_L :

$$C_L = \frac{2S_0}{\rho A}\frac{w}{V} \tag{3.5}$$

sa norme s'écrit :

$$F_L(V) = \frac{1}{2}C_L\rho A V^2 \tag{3.6}$$

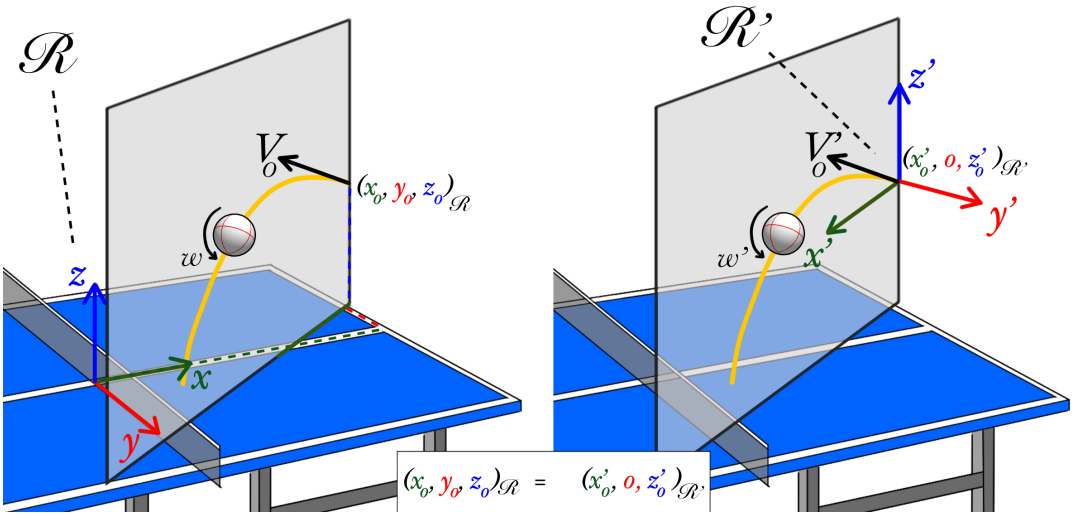


FIGURE 3.5 – Repère monde $\mathcal{R}$ au centre de la table (gauche); repère $\mathcal{R}'$ lié à la trajectoire de balle après la frappe (droite).

À partir du bilan des forces (équations 3.1 et 3.2), les composantes horizontales et verticales de $\mathbf{F_D}$ et $\mathbf{F_L}$, respectivement (F_{D_x}, F_{D_z}) et (F_{L_x}, F_{L_z}), s'écrivent :

$$\begin{cases} F_{D_x} &= -\frac{1}{2}C_D\rho A V^2\cos(\theta) \\ F_{D_z} &= -\frac{1}{2}C_D\rho A V^2\sin(\theta) \\ F_{L_x} &= -\frac{1}{2}C_L\rho A V^2\sin(\theta) \\ F_{L_z} &= +\frac{1}{2}C_L\rho A V^2\cos(\theta) \end{cases} \tag{3.7}$$

On considère maintenant le repère $\mathcal{R}'$, repère local qui a pour origine le point d'impact de la balle au moment de la frappe (soit $t = 0$), et dont ses axes x' et z' sont contenus dans le plan de sa trajectoire (voir figure 3.5, droite). À l'instant t, on note $V'(t)$ la norme du vecteur vitesse, $\theta'(t)$ l'angle du vecteur vitesse avec l'axe horizontal x', et $C_L(t) = \frac{2S_0}{\rho A}\frac{w}{V(t)}$ le coefficient de lift. En utilisant l'équation 3.1 avec $\frac{d\mathbf{V}'(t)}{dt} = \left(\frac{d^2x'(t)}{dt^2}, \frac{d^2z'(t)}{dt^2}\right)$, les équations de mouvements dans le repère $\mathcal{R}'$ s'écrivent :

$$\begin{aligned}\frac{d^2x'(t)}{dt^2} &= -\frac{1}{2m}\rho A V'(t)^2(C_D\cos(\theta'(t)) + C_L(t)\sin(\theta'(t)))\\ \frac{d^2z'(t)}{dt^2} &= -g - \frac{1}{2m}\rho A V'(t)^2(C_D\sin(\theta'(t)) + C_L(t)\cos(\theta'(t)))\end{aligned} \tag{3.8}$$

En connaissant, pour $t = 0$, la position initiale $(x'_0, 0, z'_0)_{\mathcal{R}'}$ de la balle, la norme V'_0 du vecteur vitesse, l'angle θ_0 et la vitesse de rotation angulaire $\boldsymbol{\omega_0}$ de la balle, nécessaires au calcul du coefficient de lift $C_{L_0} = C_L(t = 0)$ (voir équation 3.5), il est possible d'estimer de manière itérative les positions x'(t) et z'(t) du centre de masse de la balle le long des trajectoires.

En effet, si Δt est l'écart temporel, supposé petit, la méthode d'Euler explicite à l'ordre 2 WANNER et HAIRER, 1996 donne pour l'estimation des positions :

$$\begin{aligned}x'(t+\Delta t) &= x'(t) + V'(t)\cos(\theta(t))\Delta t\\ &\quad - \frac{1}{4m}\rho A V'(t)^2(C_D\cos(\theta(t)) + C_L(t)\sin(\theta(t)))\Delta t^2\\ z'(t+\Delta t) &= z'(t) + V'(t)\sin(\theta(t))\Delta t\\ &\quad - \frac{1}{2}(g + \frac{1}{2pm}\rho A V(t)^2(C_D\sin(\theta(t)) + C_L(t)\cos(\theta(t))))\Delta t^2\end{aligned} \tag{3.9}$$

En discrétisant le domaine temporel, avec f_s est la fréquence d'acquisition de la caméra, on obtient alors l'estimation itérative des positions de la balle :

$$
\begin{aligned}
x'[i+1] &= x'[i] + V'[i]\cos(\theta[i]) \\
&\quad - \frac{1}{4m}\rho A V'[i]^2 (C_D \cos(\theta[i]) + C_L[i]\sin(\theta[i])) \\
z'[i+1] &= z'[i] + V'[i]\sin(\theta[i]) \\
&\quad - \frac{1}{2}(g + \frac{1}{2m}\rho A V'[i]^2 (C_D \sin(\theta[i]) + C_L[i]\cos(\theta[i])))
\end{aligned}
\tag{3.10}
$$

avec $i \in \mathbb{N}$ (indice de la frame), $x'[i] = x'(i \times fs)$, $z'[i] = z'(i \times fs)$, $V'[i] = V'(i \times fs)$ et $\theta[i] = \theta(i \times fs)$. Le début de la frappe correspond à $i = 0$.

Nous faisons l'hypothèse que la vitesse de rotation ω est constante entre la frappe du joueur et un rebond sur la table (NONOMURA, NAKASHIMA et HAYAKAWA, 2010). Nous la noterons ω_0 , qui est sa valeur initiale lors de la frappe de balle.

Cette équation d'évolution de la trajectoire d'une balle après frappe va être utilisée dans deux contextes :

1. Simuler des frappes de balles dans une scène synthétique (section 3.3). Il sera ainsi possible de générer autant de trajectoires nécessaires qui seront utilisées par la suite dans notre algorithme d'apprentissage pour inférer l'information 3D à partir d'une séquence mono-caméra. De plus, dans ces conditions contrôlées, il sera ainsi possible d'avoir la vérité terrain sur les paramètres cinématiques (vitesses de translation et vitesse de rotation) que l'on cherchera à extraire à partir d'une séquence mono-caméra virtuelle.
2. Estimer ces paramètres cinématiques sur des séquences synthétiques et des séquences réelles (section 3.4)

3.3 Simulation de trajectoires de balles dans des scènes synthétiques

Dans cette section, nous décrivons la manière dont les sections de trajectoires de balles utilisant le modèle dynamique présenté en section 3.2.2 sont estimées à partir du jeu de données `synthétiques` généré avec le logiciel Blender (BLENDER ONLINE COMMUNITY, 2013). Le processus de génération de la scène synthétique 3D et du placement des 2 caméras virtuelles a été présenté en section 2.2.2. Ce jeu de données `synthétiques` contient

200 séquences vidéos, contenant chacune un échange de balle aller/retour (une frappe par joueur). Trois types de coups différents sont considérés : des `Top Spin`, coups offensifs rapides et avec une forte rotation, des `Poussettes`, coups défensifs plus lents et avec une rotation faible de direction opposée au mouvement, et des `Contre-Attaques`, coups offensifs moins rapides que les `Top Spin` mais avec une rotation restant soutenue. Nous présentons ici plus en détails la génération de trajectoires 3D des balles de la scène synthétique. Les différents rebonds (impact sur la table et avec une raquette) seront présentés dans le chapitre 4.

L'avantage de ce jeu de données `synthétiques`, est qu'il est possible d'avoir la vérité terrain sur les vitesses de translation et de rotation de la balle. Ceci n'est pas possible sur le jeu de données `avec sportifs`, pour lequel nous n'avons que les observations *Image* obtenues par les caméras rapides.

Un coup peut être déterminé par un ensemble de variables intervenant dans l'équation 3.10, et qui sont exprimées par rapport au système de coordonnées $\mathcal{R}$ de la table :

— La position initiale de la balle après la frappe $(x_0, y_0, z_0)_{\mathcal{R}}$
— Une vitesse de rotation $\boldsymbol{\omega_0}$, supposée constante sur l'ensemble de la trajectoire
— Un vecteur initial de vitesse de translation $\mathbf{V_0}$

Les trois coups considérés (`Poussettes`, `Contre-Attaques` et `Top Spins`) ont une vitesse de translation et une vitesse de rotation de balle qui diffèrent. Il est ainsi possible de faire des *a priori* pour restreindre des estimations aberrantes (une vitesse de rotation nulle pour un `Top Spin` par exemple).
Ainsi, d'après IINO et KOJIMA, 2009, la vitesse moyenne d'une balle après un `Top Spins` pour joueurs expérimentés est de $18,7\ m/s$. Nos séquences consistent principalement en des gammes de coups (répétition d'un même coup ou alternance entre deux coups), et les joueurs favorisent donc les échanges longs, d'après nos discussions avec les sportifs impliqués dans nos séquences tests. Nous avons donc limité notre vitesse maximale à 60 km/h pour ce jeu de données synthétiques, ce qui représente une vitesse de $16,66\ m/s$. Les `Contre-Attaques` sont des coups offensifs mais moins rapides que les `Top Spins`. Nous avons décidé de générer dans ce jeu de données synthétiques des vitesses supérieures à 15 km/h, soit $4,17\ m/s$, et maximales à 10 m/s. Enfin, les `Poussettes` sont des coups plutôt lents par rapport aux `Contre-Attaques` et `Top Spins`, avec une rotation de balle moins rapide et dans le sens inverse des deux autres coups. La vitesse de rotation maximale

de 15 rps, et une vitesse de translation entre 5 km/h et 20 km/h ont ainsi été fixés, ce qui représente une vitesse de translation entre $1,38$ m/s et $5,55$ m/s.

Ainsi, les plages de données utilisées pour la génération du jeu de données sont présentées dans le tableau 3.1.

TABLE 3.1 – Plages de données des vitesses de translation et de rotation initiales choisies pour chaque type de coup

Coup	**Translation (m/s)**		**Rotation (rotations/s)**	
	min	max	min	max
Top Spin	10,00	16,66	30,00	70,00
Contre-Attaque	4,17	10,00	10,00	20,00
Poussette	1,38	5,55	-15,00	0,00

Afin de générer une séquence de positions 3D successives, la position initiale de la balle (x_0, y_0, z_0) dans le référentiel de la table $\mathcal{R}$ (figure 3.5) est choisie aléatoirement selon une distribution uniforme en prenant :

— $1,37 < |x_0| < 2,00$ (x_0 en dehors de la table et jusqu'à 2 m du filet)
— $0,00 < z_0 < 0,50$ (z_0 jusqu'à 50 cm au-dessus de la table)
— $|y_0| < 1$ (jusqu'à 25 cm des bords latéraux de la table)

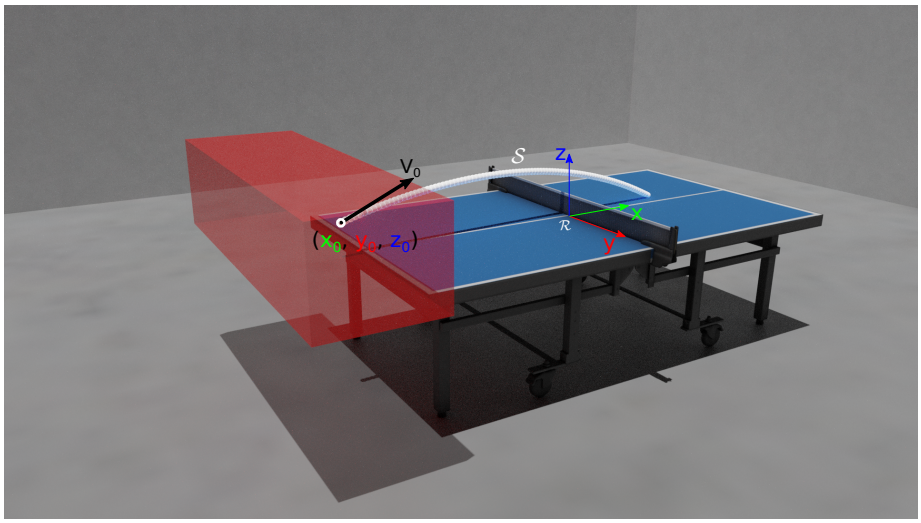


FIGURE 3.6 – Représentation de la zone d'initialisation de la trajectoire. La première position de balle (x_0, y_0, z_0) est choisie de manière aléatoire dans la zone représentée en rouge.

En plus de la position initiale de la balle (x_0, y_0, z_0), le type de coup, les normes du vecteur vitesse de translation et de rotation sont choisie aléatoirement parmi les valeurs possibles du tableau 3.1.
La direction initiale du vecteur vitesse est ensuite choisie aléatoirement.

Pour résoudre l'équation différentielle ordinaire avec ces conditions initiales, nous utilisons la méthode explicite de Runge-Kutta à l'ordre cinq (DORMAND et PRINCE, 1980). La méthode de Runge-Kutta estime itérativement les valeurs de la dérivée en plusieurs points de l'intervalle, ce qui permet d'obtenir une précision supérieure à la méthode d'Euler. L'impact sur la table est simulé à l'aide d'un modèle de rebond qui sera présentée dans le chapitre 4 (équation 4.1).

Des contrôles sont effectués pour garantir que le coup simulé est valide :

— La balle ne doit pas toucher le filet
— La balle doit toucher la table du côté de l'adversaire
— La balle ne doit rebondir qu'une seule fois du côté de l'adversaire

Dans le cas où les trois critères ne seraient pas réunis, une nouvelle direction du vecteur vitesse initial est tirée aléatoirement jusqu'à ce que la trajectoire soit valide.

Pour chaque trajectoire synthétisée, les paramètres cinématiques (vitesse de translation et de rotation) sont enregistrés, ainsi que les positions 3D de la balle dans $\mathcal{R}$, les positions de la balle projetées dans le domaine image, et enfin son diamètre estimé en pixels permettant la reconstruction 3D à partir de l'équation 2.5.

3.4 Estimation des paramètres cinématiques

Il s'agit ici d'estimer les paramètres cinématiques importants de la trajectoire de la balle (vitesses de translation et de rotation) dans le repère "physique", c'est-à-dire le repère $\mathcal{R}$. Ils sont en effet directement liés à l'effet donné à la balle par les joueurs. L'approche directe, basée sur l'image est d'abord rapidement introduite, et ses limites identifiées. Nous présentons ensuite les approches utilisant l'équation de trajectoire de mouvement pour extraire les paramètres cinématiques, à la fois sur des séquences synthétiques et des séquences réelles.

3.4.1 Approche basée image

Une fois la scène calibrée (chapitre 2), il est théoriquement possible d'estimer les vitesses de translation et de rotation de la balle à partir d'une séquence d'images mono-caméra. Si la fréquence d'acquisition est suffisante (au dessus de 250 fps par exemple), l'estimation de la vitesse de translation peut être effectuée par des techniques de suivi.

L'estimation de la vitesse de rotation est quant à elle plus difficile pour deux raisons :

— suivant l'effet mis par le joueur, la vitesse de rotation peut être très grande (jusqu'à 100 tours/s) et la fréquence d'acquisition ne pas respecter le critère de Shannon

— si la balle a une radiométrie uniforme (pas de textures, marquage, ou logo), il y aura une ambiguïté lors de l'estimation de la rotation

Sous réserve d'une fréquence d'acquisition suffisante et dans le cas favorable où la balle possède un marquage quelconque, celui-ci peut être suivi pour estimer sa vitesse de rotation, comme illustré par la figure 3.7.

FIGURE 3.7 – Rotation d'une balle bicolore à 600 fps

Comme abordé au chapitre 2, le suivi temporel du marquage sur une balle s'effectue principalement en deux étapes : détection du marquage présent sur la balle, puis un suivi sur la séquence d'images. Plusieurs types de marquages sur les balles existent, et peuvent nécessiter un détecteur particulier adapté à chaque cas. BLANK, GROH et ESKOFIER, 2017 utilisent deux caméras à 1000 Hz dans le cas d'un marquage avec des lignes noires sur la balle (voir figure 3.8).

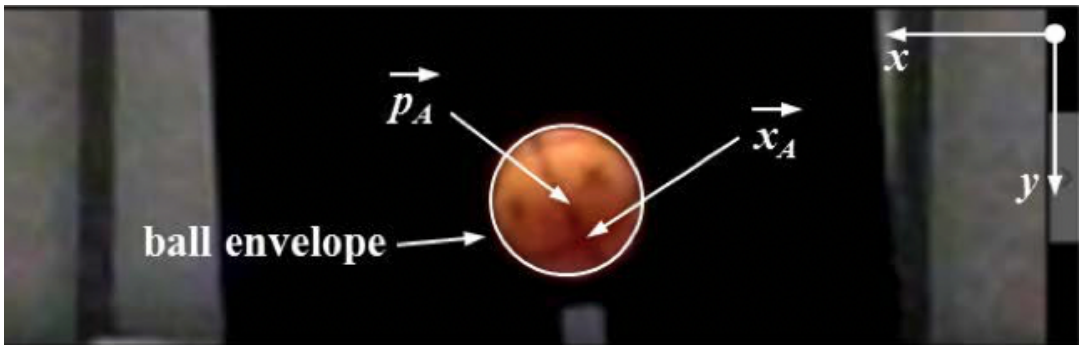


FIGURE 3.8 – Suivi de lignes sur la balle pour estimer la rotation d'une balle (BLANK, GROH et ESKOFIER, 2017)

Dans la majorité des cas, les balles ne possèdent pas de marquage par lignes mais d'un logo. Ce logo est souvent circulaire, et contient des informations comme la marque de la balle, sa provenance, parfois sa taille, ainsi que sa qualité. Dans un contexte d'applications robotiques, TEBBE et al., 2020

comparent différentes méthodes pour estimer la rotation de la balle, et notamment l'utilisation de la position du logo. Les auteurs utilisent des filtres colorimétriques pour isoler le logo sur la balle. La balle est ensuite reconstruite en 3D afin de suivre la position du logo dans la scène 3D (voir figure 3.9) . Un CNN est également entraîné pour détecter la position du logo, en remplacement des filtres colorimétriques, ce qui apporte une amélioration des résultats.

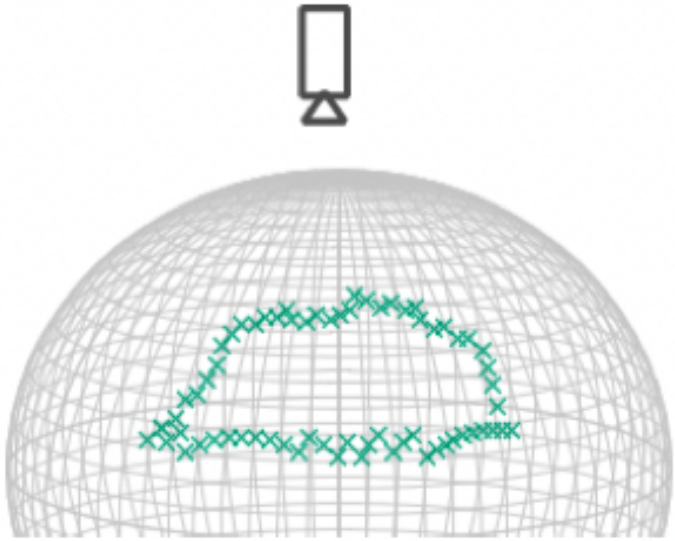


FIGURE 3.9 – Projection du logo sur un modèle 3D, (TEBBE et al., 2020)

Les limites principales des méthodes basées images pour estimer les paramètres cinématiques (vitesses de translation et de rotation) sont :

— La couleur de la balle, ou l'aspect du logo ne sont pas standardisés. Les méthodes basées images peuvent donc être impactés par cette variabilité.
— Le logo n'est pas toujours visible, car il peut se situer du côté opposé à la caméra.
— La balle pouvant aller jusqu'à plus de 133 rotations par seconde, l'emploi de caméra à haute fréquence peut être nécessaire pour éviter l'aliasing temporel et détecter la rotation. De plus, la résolution spatiale doit être assez élevée pour pouvoir observer les éventuels marquages observés sur la balle. L'utilisation d'une caméra grand public pour donc s'avérer problématique tant au niveau de la résolution fréquentielle que spatiale. De plus, il faut aussi gérer un important flux de données vidéos, ce qui implique un matériel adapté.

Dans notre contexte qui est de développer des méthodes pouvant être utilisées en situations naturelles (dites écologiques) pour le grand public ou les clubs de sport, nous souhaitons nous affranchir de matériel d'acquisition trop spécifique. Nous nous sommes ainsi tournés vers une approche basée sur un modèle physique du mouvement, et qui permet d'inférer par la trajectoire estimée les paramètres cinématiques de la balle.

3.4.2 Approche basée modèle physique

L'idée de l'approche basée sur un modèle physique (section 3.2.2) est de trouver la trajectoire issue de ce modèle qui corresponde le mieux aux observations extraites de la séquence mono-caméra. Cela revient à minimiser, en fonction des paramètres cinématiques initiaux, une fonction de coût prenant en compte les écarts entre les séquences de positions 3D obtenues à partir de la séquence d'images (section 2.3) et les positions 3D obtenues à partir de la section de trajectoire $\mathcal{S}$ dérivant du modèle physique (section 3.2.2).
Si on se place dans le repère monde $\mathcal{R}$ et avec i est l'indice de la frame, on note :

- $t[i] = i \times f_s$: variable temporelle discrète (f_s étant la fréquence de la caméra)
- $\boldsymbol{P^I}[i]$: position de la balle obtenue à partir de la séquence d'images
- $\boldsymbol{P}^{\mathcal{S}_{\tilde{V}_0,\tilde{\omega}_0}}(t[i])$: position estimée à $t[i]$ sur $\mathcal{S}$ en utilisant le modèle physique avec paramètres initiaux $(\tilde{\boldsymbol{V}}_0, \tilde{\boldsymbol{\omega}}_0)$

Les paramètres cinématiques initiaux permettant d'obtenir la trajectoire optimale sont :

$$\tilde{\mathbf{V}}_0, \tilde{\boldsymbol{\omega}}_0 = \arg\min_{\mathbf{V}_0,\boldsymbol{\omega}_0} \sum_i \|\boldsymbol{P}^{\mathcal{S}_{V_0,\omega_0}}(t[i]) - \boldsymbol{P}^I[i]\|_2^2 \tag{3.11}$$

Cette approche a été testée sur les deux jeux de données présentés au chapitre 2, avec pour chacun un objectif différent : pour le jeu de données `synthétiques`, l'existence d'une vérité terrain sur les vitesses de rotation et de translation permet de quantifier la qualité de notre méthode d'estimation.
Sur le jeu de données `avec sportifs`, pour lequel il n'est pas possible d'avoir cette vérité terrain, nous validerons la méthode par une classification sur le type d'effet donnée à la balle (qui lui est connu).

Nous détaillons ci-après ces deux expériences effectuées pour valider l'approche proposée.

Extraction des paramètres cinématiques sur le jeu de données `synthétiques`

Comme indiqué, la vérité-terrain sur les vitesses de rotation et translation étant connues sur le jeu de données `synthétiques`, l'objectif est de retrouver leurs valeurs à partir de la séquence de positions 3D extraites au chapitre 2. Cela revient à trouver la section de trajectoire $\mathcal{S}_{V_0,\omega_0}$ issue du modèle physique qui corresponde le mieux aux observations 3D extraites de la séquence, ou de manière équivalente, aux paramètres cinématiques initiaux $\boldsymbol{V_0}$ et $\boldsymbol{\omega_0}$ (équation 3.11).

Nous comparons deux méthodes d'estimation de $\boldsymbol{V_0}$ et $\boldsymbol{\omega_0}$:

— La première méthode consiste en une recherche exhaustive multi-grille sur un intervalle de valeurs de ces paramètres. Cette minimisation par force brute utilise deux tailles de grille successives pour ajuster la trajectoire cible. Une première recherche est effectuée entre des valeurs minimales et maximales fixées de chaque paramètre, soit pour la vitesse de translation de $1,4\ m/s$ à $16,6\ m/s$, et la vitesse de rotation de $-15\ rps$ à $70\ rps$ (rotations par seconde). À cette première étape, un pas d'une unité est utilisé pour chaque paramètre. Après cette approximation grossière, une grille plus fine est utilisée autour des valeurs obtenues précédemment, avec un intervalle de $-5/+5$ et un pas de $0,1$, ce qui permet de mieux ajuster la trajectoire cible.

— La seconde méthode est une minimisation non linéaire au sens des moindres carrés basée sur la méthode de Levenberg-Marquardt (NEWVILLE et STENSITZKI, 2018).

Le tableau 3.2 présente l'erreur obtenue avec la première approche, c'est-à-dire les erreurs obtenues entre la vérité-terrain et les paramètres estimés en utilisant une minimisation de recherche par grille.

TABLE 3.2 – Erreur moyenne estimée des paramètres extraits pour chacun des trois types de coups avec une méthode par grille

Type de Coup	Erreur sur $\boldsymbol{V_0}$ (m/s)	Erreur sur $\boldsymbol{\omega_0}$ (rps)
Top Spin	0,75	4,48
Contre-Attaque	0,16	2,70
Poussette	0,09	0,99
Moyenne	0,41	3,04

Des exemples de trajectoires obtenues pour une `Contre-Attaque` et une `Poussette` sont présentés sur les figures 3.10 et 3.11, ainsi que la trajectoire réelle de la balle (en bleu), obtenue par stéréo-vision (Chapitre 2). Les paramètres cinématiques initiaux V_0 et ω_0 estimés sont aussi indiqués. La trajectoire estimée en prenant en compte l'effet Magnus est très proche de la trajectoire réelle. La trajectoire obtenue avec le modèle physique mais sans prendre en compte l'effet Magnus a aussi été tracée à titre de comparaison. Elle possède le même vecteur de translation initial V_0 mais n'a par contre pas de rotation initiale ω_0. On peut donc observer que dans le cas d'une `Contre-Attaque` (voir figure 3.10), la vitesse de rotation fait retomber plus vite la balle lorsque l'effet Magnus est pris en compte. Cela se traduit lorsque l'on compare les points d'impact sur la table par rapport au même coup mais sans effet de rotation par un $\Delta x' < 0$.
En revanche, lors d'une `poussette` (voir figure 3.11), la balle effectue une rotation dans le sens inverse, ce qui allonge sa trajectoire. Dans ce cas, la comparaison des points d'impact sur la table par rapport à un coup sans effet de rotation donne comme attendu $\Delta x' > 0$.

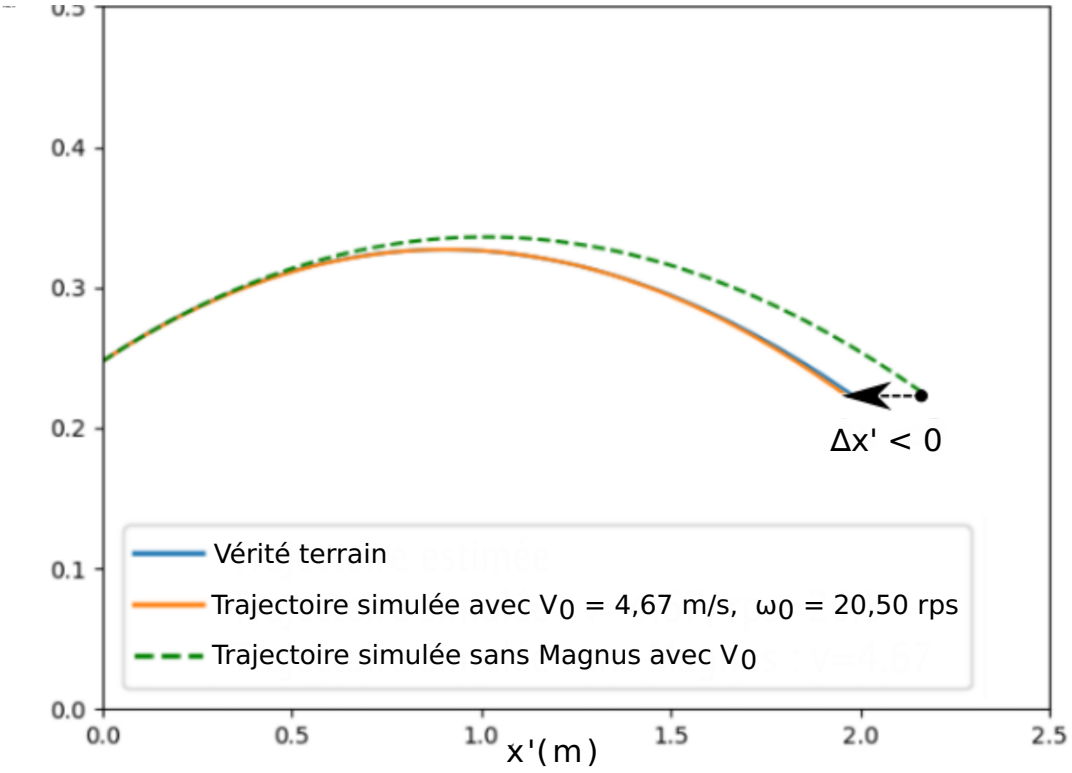


FIGURE 3.10 – Comparaison pour une `Contre-Attaque` entre la trajectoire vérité terrain, la trajectoire estimée avec effet Magnus, et la trajectoire sans effet Magnus. Avec prise en compte de l'effet Magnus, le point d'impact est plus proche de l'origine dans $\mathcal{R}'$, la balle retombe plus vite.

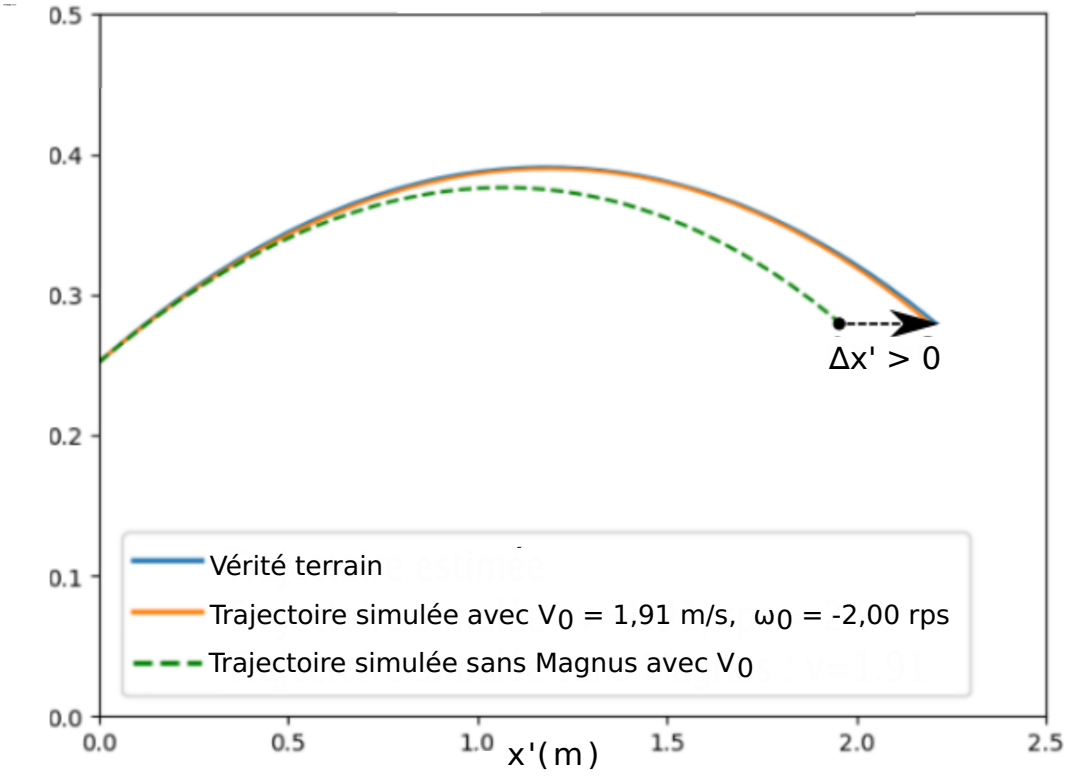


FIGURE 3.11 – Comparaison pour une `Poussette` entre la trajectoire vérité terrain, la trajectoire estimée avec effet Magnus, et la trajectoire sans effet Magnus. Avec prise en compte de l'effet Magnus, le point d'impact est plus éloigné de l'origine dans $\mathcal{R}'$, la balle retombe plus lentement.

Les erreurs obtenues sur $\boldsymbol{V_0}$ et $\boldsymbol{\omega_0}$ sont très faibles. Toutefois, l'approche de recherche par grille est coûteuse en temps de calcul car c'est une méthode par force brute. Afin d'améliorer la vitesse de traitement, la deuxième approche de minimisation utilise l'algorithme de Levenberg-Marquardt. Afin d'éviter des valeurs non-réalistes lors de la minimisation, les valeurs de $\boldsymbol{V_0}$ sont restreintes à l'intervalle $[1,39\ m/s, 16,67\ m/s]$ (correspondant à $[5\ km/h, 60\ km/h]$) et de $\boldsymbol{\omega_0}$ à l'intervalle $[-15\ rps, 70\ rps]$.

Le tableau 3.3 présente pour l'algorithme de Levenberg-Marquardt, les erreurs obtenues entre la vérité-terrain et les paramètres $\boldsymbol{V_0}$ et $\boldsymbol{\omega_0}$ estimés pour chaque type de coup.

Par rapport à une approche de recherche d'optimum par grille, le temps de calcul est réduit d'un facteur proche de 3, passant en moyenne de 89 à 27 secondes par coup. Cette méthode d'optimisation nous permet également d'améliorer l'estimation du vecteur vitesse $\boldsymbol{V_0}$ initial. L'erreur sur $\boldsymbol{\omega_0}$ est cependant plus importante par rapport à la méthode de recherche par grille. Toutefois, comme nous le verrons dans le chapitre 4, l'introduction d'un modèle de rebond permettra d'améliorer l'estimation de $\boldsymbol{\omega_0}$, tout en gardant un

TABLE 3.3 – Erreur moyenne estimée des paramètres extraits pour chaque type de coup en utilisant l'algorithme de Levenberg-Marquardt

Type de Coup	Erreur sur V_0 (m/s)	Erreur sur ω_0 (rps)
Top Spin	0,03	10,66
Contre-Attaque	0,01	6,47
Poussette	0,02	1,72
Moyenne	0,02	6,61

gain important sur le temps de calcul.

Nous présentons, pour chaque type de coup, la répartition des erreurs pour l'estimation de V_0 (Fig. 3.12). Dans ces graphiques en violon, la largeur représente la fréquence des observations, le point blanc représente la médiane, le rectangle noir représente les frontières des premier et troisième quartiles.

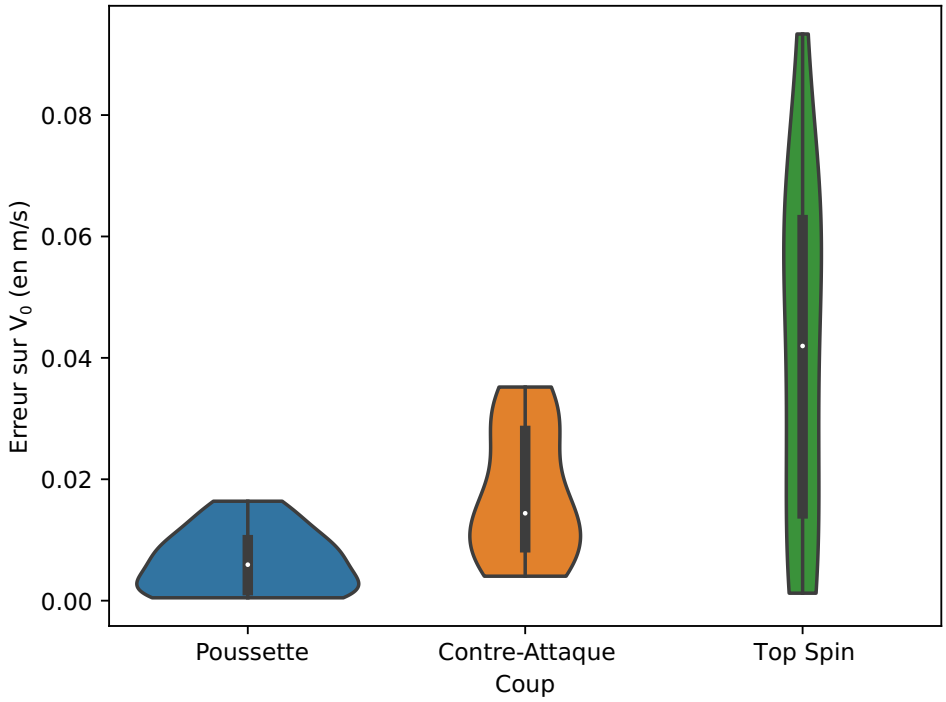


FIGURE 3.12 – Graphique en violon des erreurs de l'estimation sur V_0 .

Pour les `Contre-Attaques`, nous observons une erreur médiane pour V_0 proche de $0,01\ m/s$. Plus de la moitié des observations (rectangle noir) ont une erreur autour de 0 à 10^-2 près. Pour les `Top Spins` et les `Poussettes`, l'erreur sur V_0 est un peu plus importante, avec une erreur moyenne de $0,03\ m/s$ et $0,02\ m/s$ et une erreur médiane de $0,02\ m/s$ et $0,03\ m/s$ respectivement. L'erreur sur V_0 est toutefois très faible, le maximum obtenu étant de $0,06\ m/s$, pour un `Top Spin` alors que dans les séquences une vitesse V_0 maximale de $16,67\ m/s$ est atteinte. L'erreur moyenne pour l'estimation de V_0 sur toutes les séquences est de $0,02\ m/s$.

Nous avons jusqu'à présent estimé la vitesse initiale de translation V_0 , qui paramétrise avec ω_0 la section de trajectoire $\mathcal{S}_{V_0,\omega_0}$ de la balle. Le long de $\mathcal{S}_{V_0,\omega_0}$, la vitesse de translation V_t n'est pas constante au cours du temps, la balle étant en effet soumis à la force de gravité, celle de frottement ainsi qu'à l'effet Magnus. Sur la figure 3.13 sont présentés l'estimation de la position de la balle dans le plan de régression lors d'une `Contre-Attaque` (ligne continue), et l'estimation de l'évolution temporelle de la norme de vecteur vitesse V_t (ligne pointillée).

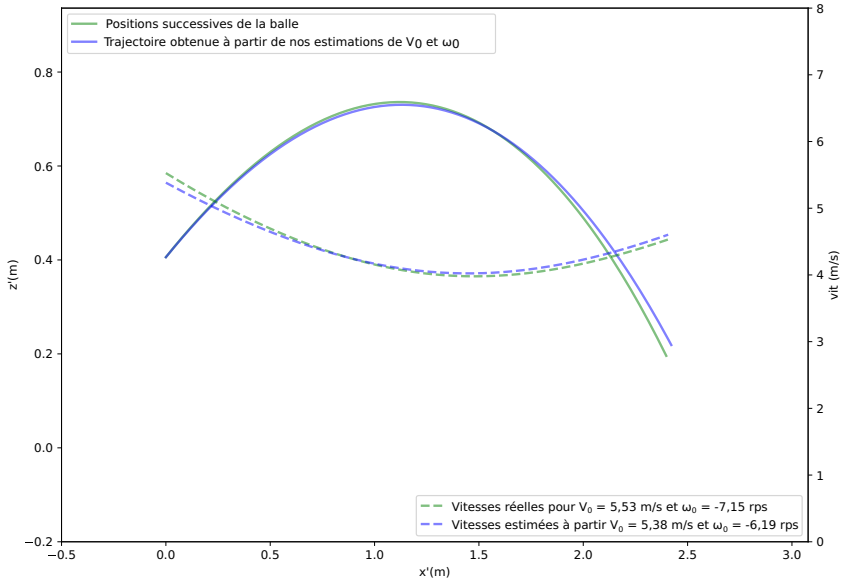


FIGURE 3.13 – Évolution au cours du temps de la position d'une balle (ligne continue) sur une `Contre-Attaque` et de sa vitesse (en pointillés)

Les erreurs sur l'estimation de la vitesse de rotation ω_0 sont illustrées par la figure 3.14. Nous rappelons que la vitesse de rotation est supposée constante sur toute la section de trajectoire $\mathcal{S}_{V_0,\omega_0}$. Comme lors de l'approche par recherche par grille, nous observons toujours une erreur d'estimation

plus faible pour les `Poussettes`, en majorité moins de 2 *rps*. L'erreur moyenne pour les `Poussettes` est de $1,72$ *rps* et l'erreur médiane est de $1,45$ *rps*.

L'erreur d'estimation sur ω_0 augmente pour les `Contre-Attaques`, qui, tout comme les `Top Spin`, ont des vitesses de rotation plus élevées que les `Poussettes`. Pour les `Contre-Attaques`, nous obtenons une erreur médiane de $6,47$ *rps*. La majorité des séquences a une erreur sur l'estimation de la rotation initiale entre 4 *rps* et $7,5$ *rps*. Pour les `Top Spin`, l'erreur moyenne est de $10,65$ *rps*, et l'erreur médiane de $9,92$ *rps*. Nous observons sur la figure 3.14 également une distribution moins étalée des erreurs pour ce type de coup.

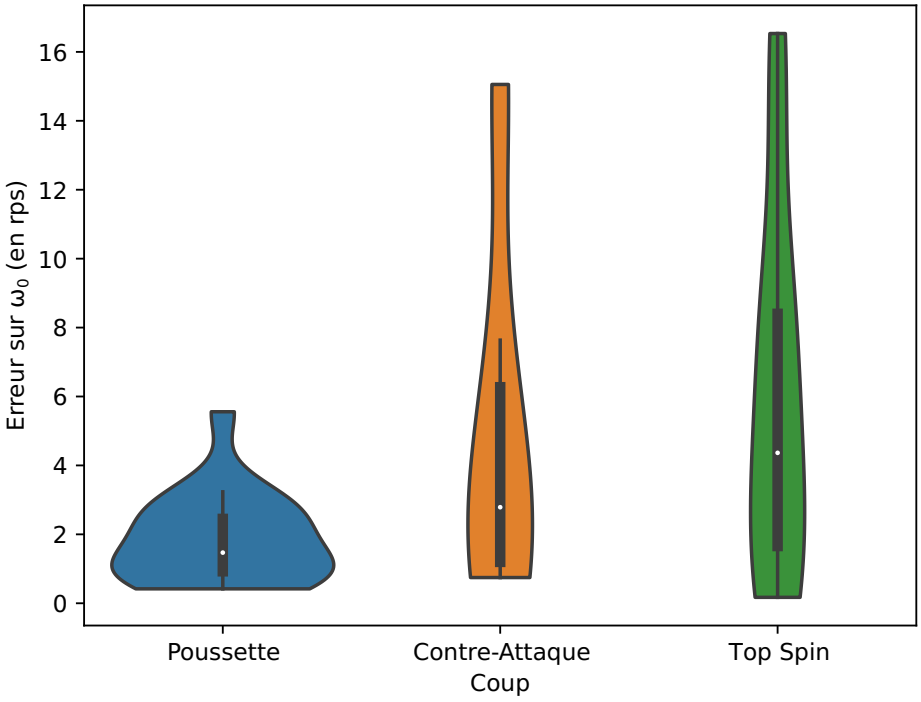


FIGURE 3.14 – Graphique en violon des erreurs de l'estimation de ω_0 .

Comme évoqué précédemment, l'erreur sur ω_0 est cependant plus importante par rapport à la méthode de recherche par grille. Toutefois, comme nous le verrons dans le chapitre 4, l'introduction d'un modèle de rebond permettra d'améliorer l'estimation de ω_0 , tout en gardant un gain important sur le temps de calcul.

Pour analyser si une corrélation existe entre l'erreur sur V_0 et celle sur ω_0 , un diagramme de dispersion a été tracé figure 3.15 pour chacun des trois types de coups.

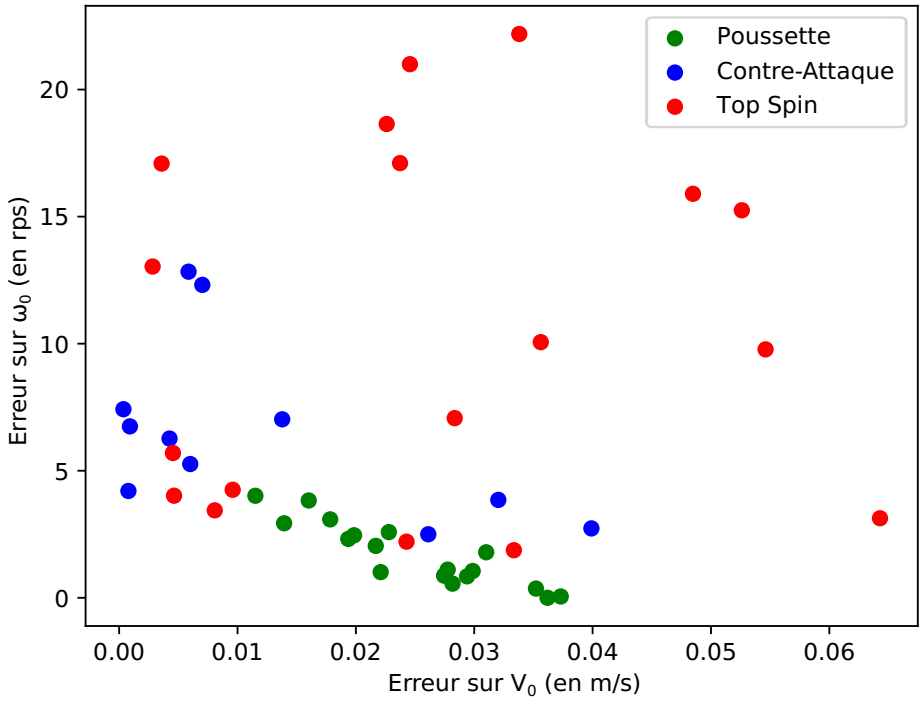


FIGURE 3.15 – Diagramme de dispersion entre les erreurs sur V_0 et ω_0

Il n'est pas observable sur cette figure de corrélation claire entre ces deux erreurs pour les `Top Spins`. En effet, le coefficient de corrélation est de -0,15, ce qui confirme l'absence de corrélation entre l'erreur sur V_0 et celle sur ω_0 .

Pour les `Contre-Attaques`, le coefficient de corrélation est de -0,55. Il y a donc une corrélation moyenne entre l'erreur sur V_0 et celle sur ω_0 .

Cependant, pour les `Poussettes`, nous observons une corrélation linéaire -0,6973. Il y a donc une corrélation entre les erreurs sur V_0 et celles sur ω_0 .

Pour les `Contre-Attaques` et `Poussettes`, le coefficient de corrélation négatif signifie que les séquences ayant une erreur de translation forte ont tendance à avoir une erreur de rotation plus faible.

La raison pourrait être que ω_0 est faible pour ces types de coups, et a donc moins d'impact sur la trajectoire que V_0 . Ayant plus d'impact, l'algorithme de minimisation choisi doit avoir tendance à privilégier une bonne estimation de vitesse que de rotation.

Reconnaissance du type de coup à partir des paramètres cinématiques sur le jeu de données `avec sportifs`

Le jeu de données `avec sportifs`, présenté dans la section 2.2.1, n'a pas de vérité terrain sur les vitesses de translation ou de rotation. La seule information à notre disposition est le type de coup effectué, qui était une consigne lors de l'enregistrement des séquences avec les sportifs. Toutefois, la pertinence des paramètres cinématiques extraits peut être testée pour la reconnaissance des trois types de coup : `Top Spin`, `Contre-Attaque` et `Poussette`. Ces trois types coups ont chacun des effets sur la rotation de la balle qui leur sont propres.
Lors du chapitre 2, 24 séquences parmi les 33 à notre disposition ont servi à entraîner le CNN pour estimer la position 3D de la balle. Nous utiliserons ici 9 séquences pour nos tests, soit trois séquences pour chaque type de coup. Après estimation des positions 3D de la balle dans chaque séquence avec sportifs, les paramètres cinématiques $\boldsymbol{V_0}$ et $\boldsymbol{\omega_0}$ sont extraits en utilisant la méthode de Levenberg-Marquardt. À partir de ces paramètres, une classification naïve bayésienne est effectuée (H. ZHANG, 2004). Les paramètres $\boldsymbol{V_0}$ et $\boldsymbol{\omega_0}$ obtenus pour chaque séquence et les scores de classification sont présentés dans le tableau 3.4.

TABLE 3.4 – Paramètres cinématiques extraits et taux de classification sur le jeu de données `avec sportifs`

Séquence	$\boldsymbol{V_0}$ (m/s)	$\boldsymbol{\omega_0}$ (rps)	Label	Taux de classification (%)		
				TS	Contre-Attaque	Poussette
Seq. 1	13,00	**38,50**	Top Spin	**100**	0	0
Seq. 2	14,00	32,50	Top Spin	**100**	0	0
Seq. 3	13,00	32,00	Top Spin	**100**	0	0
Seq. 4	10,00	10,00	Contre-Attaque	3	**97**	0
Seq. 5	9,80	9,50	Contre-Attaque	0	**100**	0
Seq. 6	9,00	9,00	Contre-Attaque	0	**100**	0
Seq. 7	5,00	-15,00	Poussette	0	0	**100**
Seq. 8	4,80	-13,50	Poussette	0	0	**100**
Seq. 9	5,00	-14,00	Poussette	0	0	**100**

Un classificateur simple tel que le classificateur naïf bayésien permet d'obtenir des taux de classification quasi-parfait pour le jeu de données utilisé. Cela illustre l'aspect discriminant des paramètres cinématiques extraits pour les trois types de coups choisis. De plus, ces valeurs peuvent servir aux entraîneurs et joueurs en donnant une indication sur la qualité de réalisation

d'un coup, par exemple pour quantifier si la rotation a été suffisante dans le cas d'un `Top Spin`.

3.5 Conclusion

Dans ce chapitre, nos avons proposé une méthode pour extraire à partir d'une séquence de points 3D précédemment estimée (Chapitre 2) une trajectoire "*Physique*", obtenue à partir de l'équation du mouvement de la balle. Cette trajectoire dépend de paramètres cinématiques, que sont la vitesse de translation et la vitesse de rotation de la balle. Ces derniers sont intéressants pour caractériser et quantifier la performance d'un coup effectué par un joueur, et nous nous sommes focalisés sur leur estimation dans ce chapitre.
Section 3.3, nous avons présenté plus en détails la génération de séquences `synthétiques` à partir du modèle physique introduit. Nous avons présenté dans la section 3.4 l'évaluation des paramètres cinématiques extraits sur un jeu de données `synthétiques`, en comparant avec la vérité terrain sur les vitesses de rotation et de translation initiales.

La méthode proposée utilise une minimisation avec l'algorithme de Levenberg-Marquardt. Elle permet de réduire le temps de calcul d'un facteur proche de 3 par rapport à une approche de recherche d'optimum par grille, tout en améliorant l'estimation du vecteur vitesse $\boldsymbol{V_0}$ initial. L'erreur sur $\boldsymbol{\omega_0}$ est cependant plus importante par rapport à la méthode de recherche par grille.

Enfin, sur le jeu de données `avec sportifs`, dont la vérité terrain est inconnue, nous avons extrait les paramètres cinématiques pour effectuer une classification sur 3 classes de coups, et démontrer leur pertinence.

Nous introduisons dans le chapitre 4 suivant un modèle de rebond de la balle. L'angle d'incidence est impacté par la rotation de la balle, et nous utiliserons cet angle pour améliorer l'estimation sa vitesse de rotation $\boldsymbol{\omega_0}$, tout en gardant un gain important sur le temps de calcul. Ce modèle de rebond permet de simuler un rebond sur une surface statique, comme la table, mais également sur une surface en mouvement, comme la raquette. L'apport du modèle de rebond va permettre d'extraire d'autres informations d'intérêt pour le joueur et l'entraîneur, telles que l'angle de frappe de la balle ou la vitesse de la raquette au moment de l'impact.

Chapitre 4

Modèles de rebonds et estimations de la cinématique de la balle et de la raquette

Sommaire

4.1 Introduction

À l'issue du chapitre précédent, nous avons obtenu une estimation des paramètres cinématiques de la balle entre le début de sa trajectoire, caractérisée par des vitesses $(\mathbf{V}_0, \boldsymbol{\omega}_0)$ à l'instant $t = 0$, et sa fin à l'instant du premier impact sur la table. Nos expérimentations ont montré que l'estimation de la vitesse de translation est robuste mais que l'estimation de la vitesse de rotation est perfectible.

L'objectif de ce chapitre est d'améliorer cette estimation en intégrant dans notre modèle les rebonds sur la table et sur la raquette. Afin de clarifier le discours, nous utiliserons les conventions suivantes :

Un échange entre deux joueurs sera représenté comme un ensemble de quatre sections $\mathcal{S}_.$ séparées par des impacts $\mathcal{I}_.$. La section $\mathcal{S}_0$ débute à l'instant $t = 0$, et se termine au premier impact $\mathcal{I}_1$ sur la table, $\mathcal{S}_1$ la prolonge jusqu'à l'impact $\mathcal{I}_2$ sur la raquette d'un joueur. $\mathcal{S}_2$ et $\mathcal{S}_3$ sont séparées par l'impact $\mathcal{I}_3$ sur la table. $\mathcal{I}_4$ correspond à l'impact avec la raquette adverse et termine l'échange. La Figure 4.1 illustre ce découpage de la trajectoire.

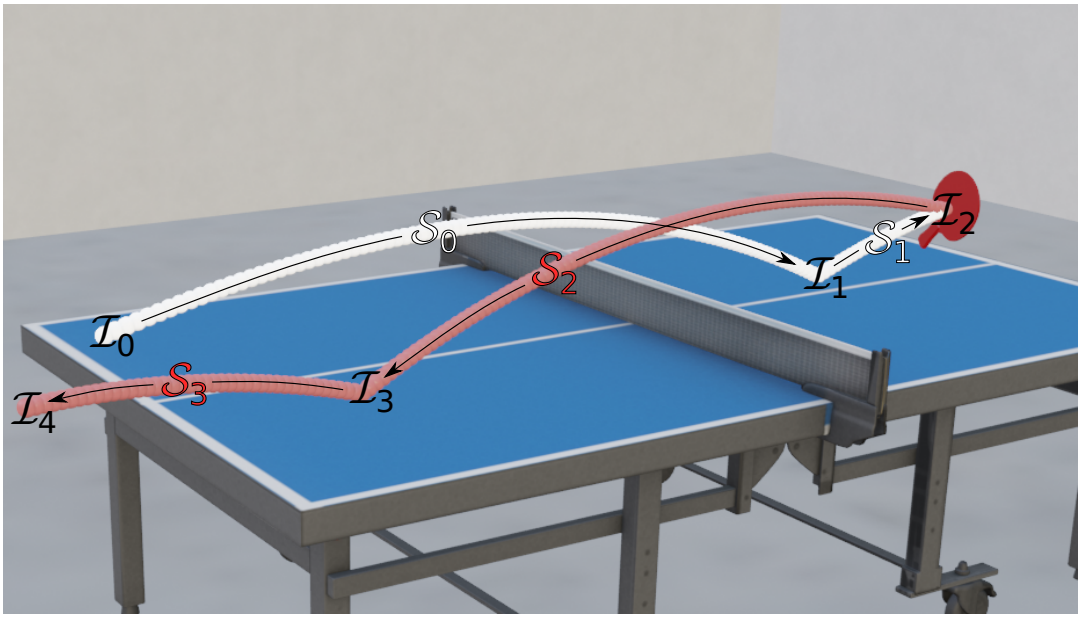


FIGURE 4.1 – Sections $\mathcal{S}_.$ et impacts $\mathcal{I}_.$ d'une trajectoire correspondant à un échange.

La modélisation des rebonds est complexe car dépendante de nombreux paramètres : matériaux entrant en contact avec la balle (matériel constituant le plateau de la table, composition de la raquette), vitesse de translation et de rotation avant l'impact.

En particulier, la vitesse de rotation avant impact de la balle conditionne l'angle avec la verticale de sa trajectoire après rebond. La Figure 4.2 illustre les trois cas typiques à considérer. La géométrie des trajectoires avant et après impact est donc conditionnée par la vitesse de rotation, et nous exploiterons par la suite ce lien pour améliorer l'estimation des paramètres cinématiques. Pour cela, nous utiliserons un modèle physique de rebond de la balle adapté aux différents cas de figure.

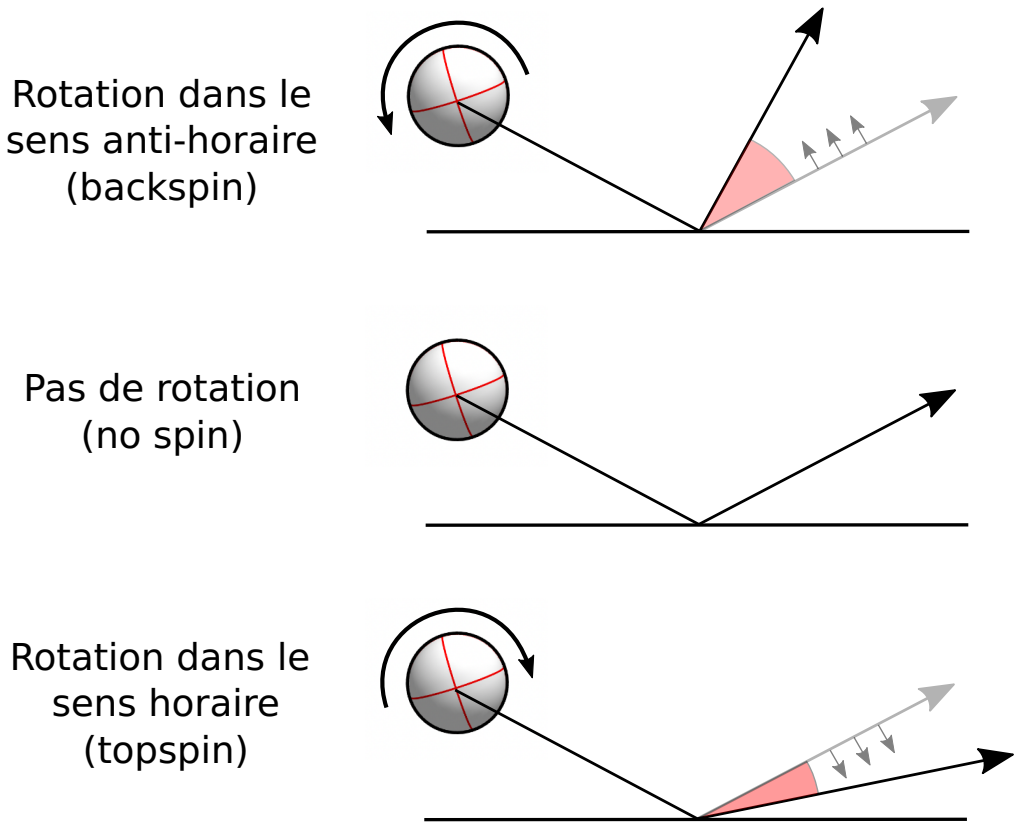


FIGURE 4.2 – Impact de la rotation sur le rebond

Les principales contributions de ce chapitre sont :

— L'amélioration de l'estimation de la vitesse de rotation de la balle en prenant en compte le rebond sur la table,

— L'utilisation des paramètres cinématiques avant et après impact avec la raquette pour déduire l'angle de frappe et la vitesse de la raquette et donc de lier la trajectoire de la balle et geste sportif.

Ce chapitre est divisé en deux sections, répondant chacune à une des deux problématiques citées.

En section 4.2, nous commençons par introduire un modèle de rebond sur la table, et améliorons l'estimation des paramètres cinématiques obtenus au chapitre 3.

En section 4.3, nous analysons l'impact entre la balle et la raquette, ce qui permettra de connaître les paramètres cinématiques de la raquette lors de la frappe.

Nous concluons enfin ce chapitre en section 4.4 en montrant les points forts de notre approche, les pistes d'amélioration et les développements futurs en termes d'analyse du geste sportif.

4.2 Prise en compte du rebond balle/table dans l'estimation des paramètres cinématiques de la balle

4.2.1 Modèle de rebond balle/table

Lorsque qu'une balle en mouvement, avec une vitesse de rotation nulle, entre en contact avec une surface horizontale, et que les frottements sont négligeables, l'angle d'incidence de sa trajectoire est égal à l'angle après rebond (voir Figure 4.3). De plus, la balle entre en contact avec la table en un seul point qui est à la fois, le point d'impact de la balle sur la table et également le point de rebond.

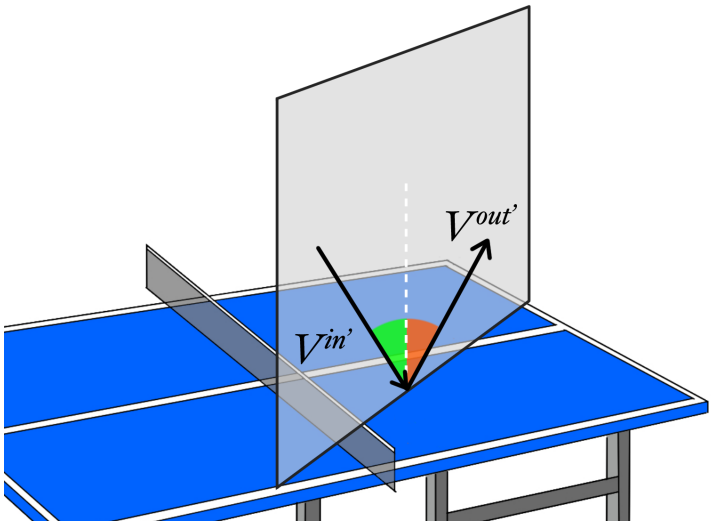


FIGURE 4.3 – Cas simplifié : Angle d'incidence = Angle après rebond

En pratique, ce modèle simplifié n'est pas exploitable dans notre cas, car trop éloigné des conditions réelles de jeu du tennis de table, et différents paramètres doivent être pris en compte pour obtenir une modélisation physiquement réaliste du rebond. En plus de la vitesse de rotation de la balle, les

matériaux constituant la balle et la table doivent être considérés. Chaque matériau possède une élasticité au rebondissement qui est modélisée par le coefficient de restitution (COR) de Newton et qui est défini comme le rapport entre la vitesse après impact et la vitesse avant impact. Sur cette base, de nombreux travaux sur l'interaction entre une balle et une surface ont été menés. GARWIN, 1969 étudie les interactions entre une balle rebondissant dans un plan vertical entre deux surfaces horizontales et parallèles qui figurent le sol et le plafond. Cet auteur propose un modèle de collision vérifiant la conservation de l'énergie cinétique, et analyse différentes problématiques liées au modèle de rebond. Il montre que celui-ci est affecté par l'angle d'incidence, la vitesse de la balle et sa vitesse de rotation mais également par la déformation de la balle lors d'un impact. Cette déformation peut provoquer un roulement ou un glissement sur la table, qui sont caractérisés par une quantité V_s qui sera définie par la suite. Ce comportement de la balle décale la position du point d'impact de celle du point de rebond et angle d'incidence et angle après rebond sont distincts. Dans le premier cas, le vecteur vitesse après rebond se rapproche de la verticale, dans le second il s'en écarte. Ces deux cas sont visualisés Figure 4.4.

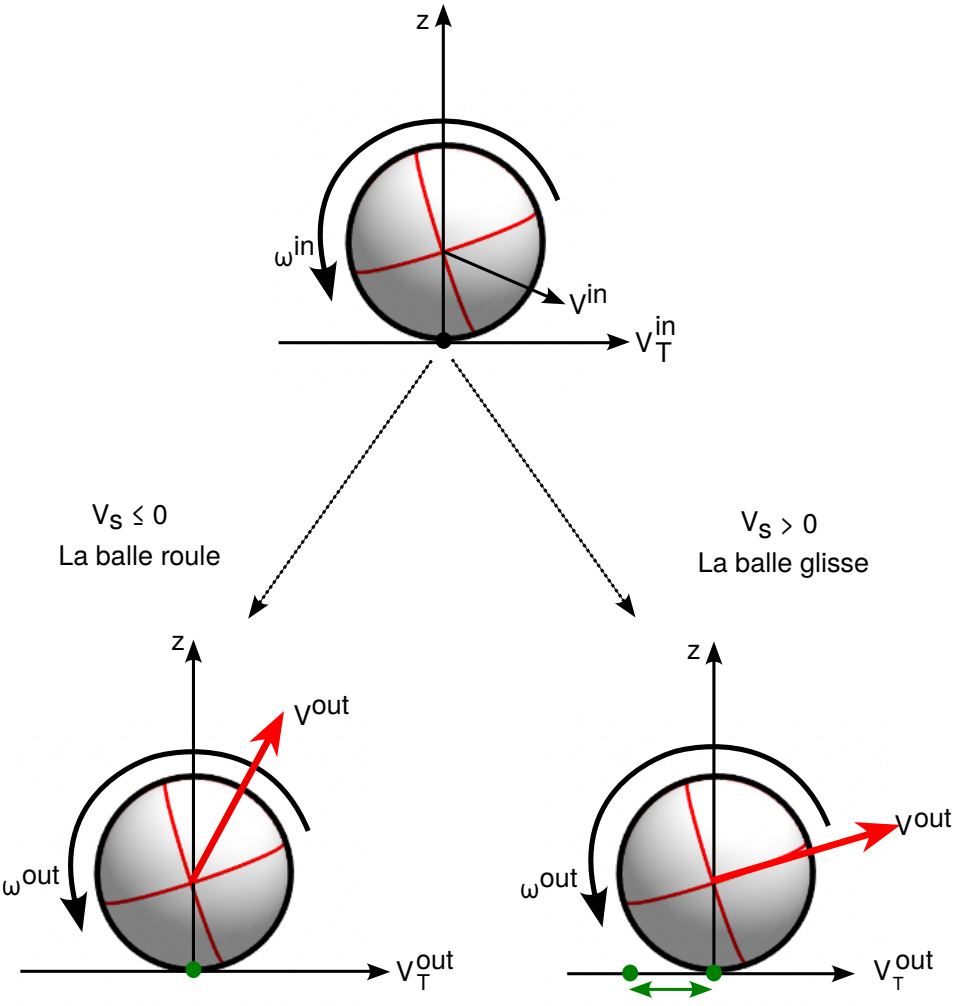


FIGURE 4.4 – Lors d'un impact, la balle se déforme légèrement et peut rouler ($V_S \leq 0$) ou glisser ($V_s > 0$) sur la table (D'après (NONOMURA, NAKASHIMA et HAYAKAWA, 2010))

Dans ce chapitre, nous utiliserons $(\mathbf{V}^{in}, \boldsymbol{\omega}^{in})$ et $(\mathbf{V}^{out}, \boldsymbol{\omega}^{out})$ pour désigner les vecteurs vitesses de translations et rotations avant et après rebond dans le repère de la table, ou repère monde, $\mathcal{R}$. Le centre de la table correspond à l'origine de ce repère.

Différents modèles de rebond, prenant en compte les observations précédentes, ont été proposés dans la littérature. Le modèle le plus simple prenant en compte la rotation de la balle est celui de Y. HUANG et al., 2011. À partir de trajectoires observées dans des vidéos, cet auteur utilise la méthode des moindres carrés pour décrire par une fonction polynomiale la relation entre les couples de vecteurs vitesse de translation et de rotation avant et après rebond dans le référentiel lié à la table.

Ce modèle, bien que facile à utiliser, ne fait pas intervenir de modèle physique à proprement parler, et nous avons préféré utiliser un autre modèle introduit par BAO et al., 2012 ; NONOMURA, NAKASHIMA et HAYAKAWA, 2010. Celui-ci fait plusieurs hypothèses :

— Le rebond est défini en un point de contact unique sur la table. Il ne dépend que des paramètres cinématiques de la balle (rotation, translation, ...) à l'instant de l'impact en $\mathcal{I}_1$. Le décalage éventuel entre point d'impact et point de rebond, mesuré par V_s est ignoré mais l'inclinaison du vecteur après rebond est modifié en conséquence (voir équation 4.5).

— Les composantes en z de la vitesse avant collision v_z^{in} et de la vitesse après collision v_z^{out} sont reliées par l'équation $v_z^{in} = \epsilon_t v_z^{out}$, où ϵ_t est le COR de la table. Ce coefficient dépend de la table, néanmoins, la valeur $\epsilon_t = 0,93$, proposée par NONOMURA, NAKASHIMA et HAYAKAWA, 2010 est couramment utilisée dans la littérature, c'est celle que nous utiliserons par la suite.

— Le rebond est effectué dans la "direction" du vecteur vitesse de translation de la balle (*i.e.* $\mathbf{V}^{in}$ et $\mathbf{V}^{out}$ sont tous les deux dans le plan vertical défini par la trajectoire de la balle).

Dans ce cas, le vecteur de vitesse après rebond $\mathbf{V}^{out}$ est donné par :

$$\mathbf{V}^{out} = \mathbf{A_v}\mathbf{V}^{in} + \mathbf{B_v}\boldsymbol{\omega}^{in} \tag{4.1}$$

où les matrices $\mathbf{A_v}$ et $\mathbf{B_v}$ sont définies par :

$$\mathbf{A_v} = \begin{bmatrix} 1-\beta & 0 & 0 \\ 0 & 1-\beta & 0 \\ 0 & 0 & -\epsilon_t \end{bmatrix}, \mathbf{B_v} = \begin{bmatrix} 0 & \beta r & 0 \\ -\beta r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.2}$$

Le coefficient β dépend du comportement de la balle au moment de l'impact et sera défini au paragraphe suivant.

Le vecteur de rotation après rebond $\boldsymbol{\omega}^{out}$ est, lui, défini par :

$$\boldsymbol{\omega}^{out} = \mathbf{A}_{\boldsymbol{\omega}}\mathbf{V}^{in} + \mathbf{B}_{\boldsymbol{\omega}}\boldsymbol{\omega}^{in} \tag{4.3}$$

où les matrices $\mathbf{A}_\omega$ et $\mathbf{B}_\omega$ sont définies par :

$$\mathbf{A}_\omega = \begin{bmatrix} 0 & -\frac{3\beta}{2r} & 0 \\ \frac{3\beta}{2r} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{B}_\omega = \begin{bmatrix} 1-\frac{3\beta}{2r} & 0 & 0 \\ 0 & 1-\frac{3\beta}{2r} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

Dans NONOMURA, NAKASHIMA et HAYAKAWA, 2010 les auteurs définissent β en considérant le coefficient de friction de la balle sur la table, noté μ, et la vitesse tangentielle à la table $\mathbf{V}_T^{in}$ au moment de l'impact :

$$\mathbf{V}_T^{in} = \begin{bmatrix} v_x^{in} \\ v_y^{in} \\ 0 \end{bmatrix} + \boldsymbol{\omega}^{in} \times \begin{bmatrix} 0 \\ 0 \\ -r \end{bmatrix} = \begin{bmatrix} v_x^{in} - r\omega_y^{in} \\ v_y^{in} + r\omega_x^{in} \\ 0 \end{bmatrix} \tag{4.5}$$

où r est le rayon de la balle. Lors du contact avec la table, la balle peut adopter un des comportements déjà cités : soit elle roule, soit elle glisse. Pour distinguer les deux cas, les auteurs considèrent la quantité $V_s = 1 - \frac{5}{2}\mu(1-\epsilon_t)\frac{|v_z^{in}|}{\|\mathbf{V}_T^{in}\|}$. Si $V_s > 0$, alors la balle glisse sur la table et $\beta = \mu(1-\epsilon_t)\frac{|v_z^{in}|}{\|\mathbf{V}_T^{in}\|}$. Dans le cas contraire, $V_s \leq 0$, la balle roule, et $\beta = \frac{2}{5}$.

4.2.2 Amélioration de l'estimation de la vitesse de rotation en utilisant le rebond balle/table

Dans le chapitre 3 nous avons obtenu une estimation fiable de la vitesse de translation de la balle ainsi qu'une première estimation de sa vitesse de rotation. Comme illustré sur la figure 4.5, ces estimations utilisent uniquement les sections de trajectoires entre deux impacts, sans prise en compte des informations données par le rebond.

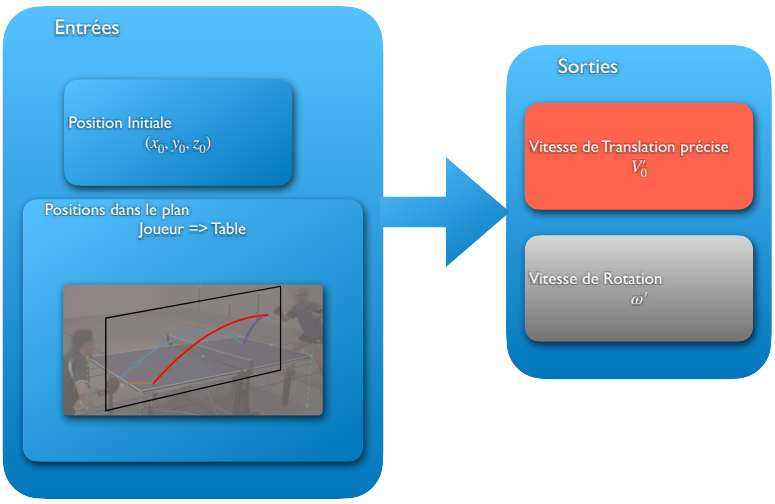


FIGURE 4.5 – En utilisant $\mathcal{S}_0$ seule, l'estimation de la vitesse de translation est précise, mais pas celle de la vitesse de rotation.

Théoriquement, le modèle physique de rebond doit permettre de lier les vitesses avant et après impact et donc d'améliorer globalement leur estimation. Cependant, dans les acquisitions vidéos que nous avons effectuées, le moment de l'impact n'est généralement pas observé car situé entre deux images consécutives. Cela reste vrai même avec des caméras rapides, comme le montre Fa figure 4.6 obtenue avec une vitesse d'acquisition de 240 images par seconde.

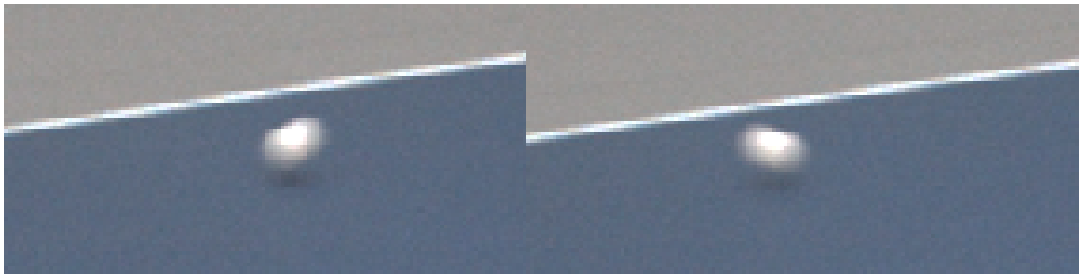


FIGURE 4.6 – Exemple d'images acquises à 240 fps : sur celle de gauche la balle n'a pas touché la table, sur celle de droite elle a déjà rebondi.

Pour obtenir l'angle et la vitesse au moment de l'impact, il faut donc estimer avec précision la position 3D de la balle au moment du rebond (impact $\mathcal{I}_1$). Pour cela, nous devons ré-échantillonner les trajectoires et connaître précisément le temps écoulé entre $t = 0$ et le rebond.

Ré-échantillonage temporel : estimation du moment de l'impact

Au chapitre 3, nous avons étudié une séquence de positions 3D de balles pour obtenir une trajectoire pseudo-parabolique $\mathcal{S}_0$ qui correspond à la première des 4 sections $\mathcal{S}_.$ de la trajectoire considérée dans le chapitre actuel (voir Figure 4.1).

Pour chacune de ces sections, les trajectoires issues des positions successives de la balle au cours du temps sont décrites dans le repère impact $\mathcal{R}'$ introduit au chapitre 3. Dans le plan vertical $x'z'$ elles peuvent être approximées par un polynôme de degré 4. De même, dans le plan $x'y'$ elles peuvent être approximées par un polynôme de degré 2 (LIN, YU et Y. C. HUANG, 2020). Comme nous supposons que la balle se déplace dans un plan vertical, nous ne considérerons que le seul polynôme dans le plan $x'z'$.

Si nous souhaitions prendre en compte les effets latéraux, appelés `Sidespin`, que l'on rencontre notamment dans le cas de l'analyse de services, il faudrait abandonner l'hypothèse du déplacement de la balle dans un plan, et il serait nécessaire d'utiliser deux polynômes, un sur l'axe x', et un autre sur l'axe y' pour analyser l'évolution de la balle. Ceci n'a toutefois pas été fait dans cette thèse et fera partie des perspectives.

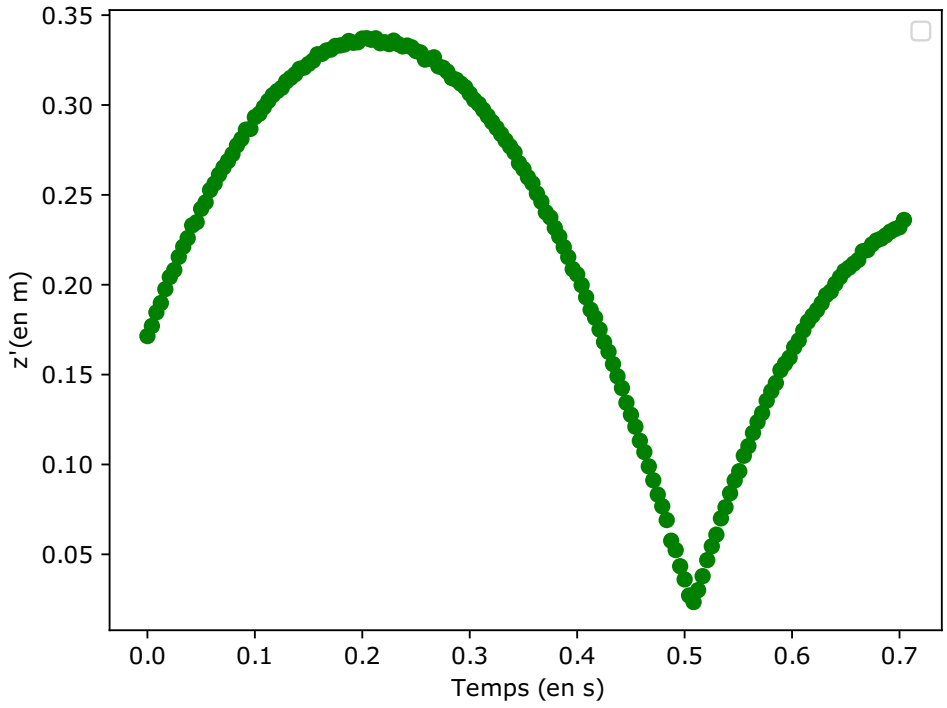


FIGURE 4.7 – Évolution de la position de la balle sur l'axe z' en fonction du temps.

La Figure 4.7 donne un exemple d'évolution de la position du centre de la balle sur l'axe z' en fonction du temps entre les impacts $\mathcal{I}_0$ et $\mathcal{I}_1$. Deux polynômes, P_0 et P_1 de degré 4 sont utilisés pour modéliser chacune des 2 parties de cette trajectoire. P_0 représente la section de trajectoire entre la frappe et le rebond sur la demie-table adverse ($\mathcal{S}_0$), et P_1 représente la partie de trajectoire entre le rebond sur la table et la frappe de l'adversaire ($\mathcal{S}_1$) :

$$\begin{cases} P_0(t) & = at^4 + bt^3 + ct^2 + dt + e \\ P_1(t) & = a't^4 + b't^3 + c't^2 + d't + e' \end{cases} \tag{4.6}$$

La Figure 4.8 illustre les positions sur l'axe z' ainsi que le résultat des régressions polynomiales P_0 et P_1 obtenues en minimisant l'erreur au sens des moindres carrés entre les échantillons $z[i]$ et les valeurs des polynômes aux temps $t[i]$:

$$(\tilde{a}, \tilde{b}..., \tilde{e}), (\tilde{a'}, \tilde{b'}..., \tilde{e'}) = \arg \min_{\substack{(a,b...,e)\in\mathbb{R}^5 \\ (a',b'...,e')\in\mathbb{R}^5}} \sum_{i=0}^{n_1} (P_0(t[i]) - z[i])^2 + \sum_{i=n_1+1}^{n} (P_1(t[i]) - z[i])^2 \tag{4.7}$$

où les $(\tilde{a}, \tilde{b}..., \tilde{e}), (\tilde{a'}, \tilde{b'}..., \tilde{e'})$ désignent les estimations des coefficients des polynômes, n_1 le nombre d'images correspondant à la première portion de la trajectoire, et donc au nombre de points avant l'impact $\mathcal{I}_1$, et n le nombre total d'images acquises entre $\mathcal{I}_0$ et $\mathcal{I}_2$.

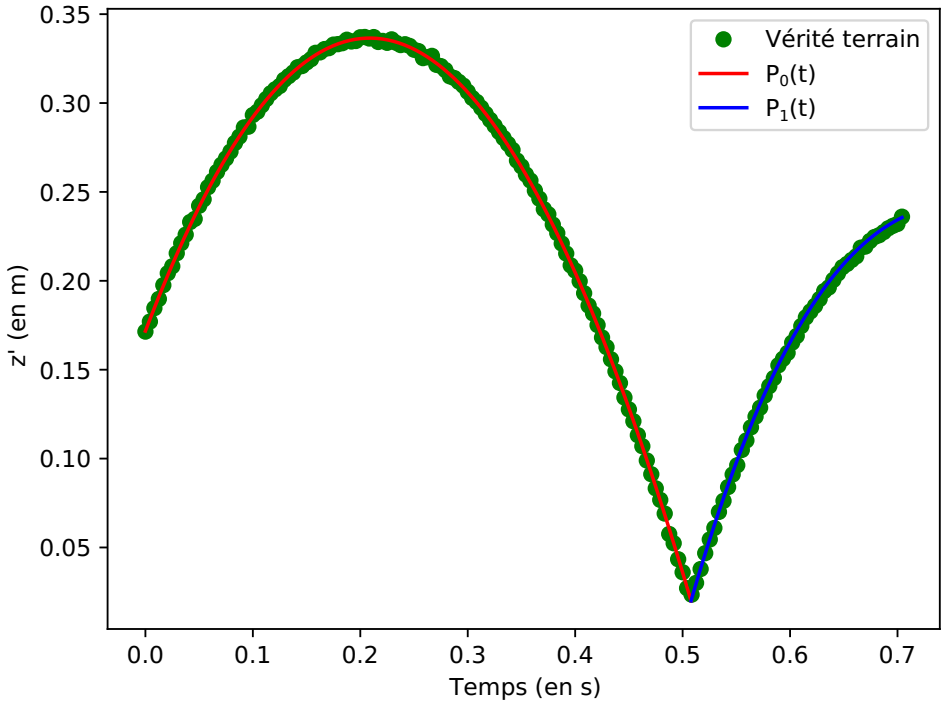


FIGURE 4.8 – Régression polynomiale de degré 4 sur les sections de trajectoires $\mathcal{S}_0$ et $\mathcal{S}_1$.

Le moment de l'impact au point $\mathcal{I}_1$ est noté t_1. Pour le déterminer, nous résolvons l'équation $P_0(t_1) = P_1(t_1)$. Quatre solutions existent, puisque le polynôme $P_0 - P_1$ est de degré 4, cependant, avec une fréquence d'acquisition F_s, fixée ici à 240 fps, nous ne conservons que celles comprises entre $\frac{n}{F_s}$, et $\frac{n+1}{F_s}$.

De plus, comme le rayon r de la balle est de 2 cm, le point d'impact observé devrait être à cette hauteur au-dessus du plateau de la table. En pratique, $P_0(t_1) = P_1(t_1)$ est proche de cette valeur, mais un léger décalage, $\Delta z'$, est souvent constaté. Par exemple, cela est visible sur la Figure 4.9 qui correspond à une acquisition du jeu de données `avec sportifs`. En moyenne, sur ces mêmes données, $P_0(t_1) = P_1(t_1) = r + \Delta z' \approx 2,21\ cm$, soit une erreur d'environ 2 mm, ce qui est faible relativement à la distance de la balle à la caméra, qui, pour fixer les idées, est de $3,65\ m$ pour ces acquisitions. Cette erreur est probablement directement liée aux erreurs de calibration (voir chapitre 2). Afin de pouvoir utiliser notre modèle physique, dans lequel la balle est supposée rebondir à une hauteur de 2 cm, nous corrigeons ce décalage en soustrayant la valeur $\Delta z'$ aux deux polynômes. Sur la figure 4.9, après correction, $P_0(t_1) - \Delta z' = P_1(t_1) - \Delta z' = r = 2\ cm$.

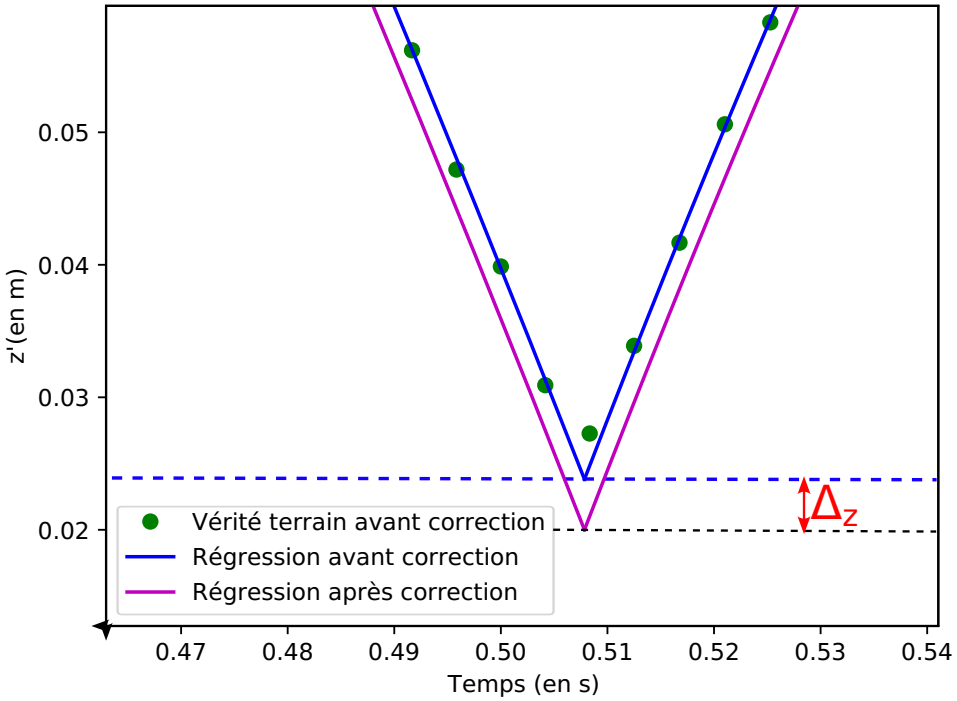


FIGURE 4.9 – Exemple de décalage sur l'axe z' au moment du rebond à l'instant t_1 en $\mathcal{I}_1$.

Après avoir déterminé t_1, nous pouvons ré-échantilloner temporellement les positions de la balle afin d'estimer le moment de l'impact, qui est essentiel pour déterminer l'angle et vitesse d'incidence. Nous prenons $n_e = 100$ échantillons entre $t = 0$ et t_1, en utilisant comme pas $\delta t = \frac{t_1}{n_e}$. Nous utilisons ce même pas temporel pour l'échantillonnage après impact sur la section $\mathcal{S}_1$.

$$z'[i] = \begin{cases} P_0(i\delta t) - \Delta z' & \text{si } 0 \leq i \leq n_e \\ P_1(i\delta t) - \Delta z' & \text{sinon} \end{cases}$$

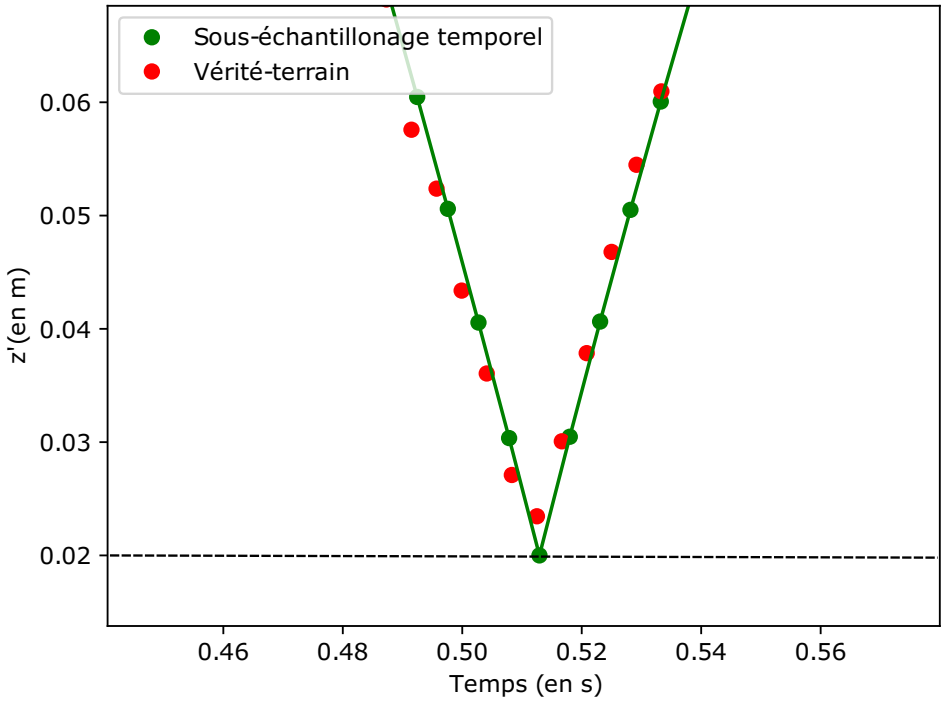


FIGURE 4.10 – Comparaison entre la trajectoire initiale et la trajectoire après ré-échantillonnage temporel avec point d'impact estimé.

Sur la Figure 4.10, nous observons la différence entre les positions de la balle obtenues avec un pas d'échantillonnage temporel de $\frac{1}{F_s} = \frac{1}{240}s$, et les positions avec un pas de $\frac{t_1}{n_e} = \frac{1}{100}$ seconde.

Une fois la correction effectuée, nous estimons, avec les mêmes méthodes qu'en section 4.2, les paramètres cinématiques initiaux, à $t = 0$, à savoir, position de la balle en 3D, vecteur vitesse, et vitesse de rotation.

La condition d'arrêt du modèle physique est le rebond sur la table, c'est-à-dire à $z' = 0,02\ m$.

Comme le montre l'équation 4.1, le modèle de rebond introduit par NONOMURA, NAKASHIMA et HAYAKAWA, 2010 ne dépend que des conditions au moment de l'impact.

Après notre estimation initiale des paramètres cinématiques sur la première section de la trajectoire, nous utilisons les deux derniers points de la trajectoire ré-échantillonnée pour estimer par différences finies $\mathbf{V}^{in}$, le vecteur vitesse au moment de l'impact. Nous avons utilisé l'hypothèse, classique pour le tennis de table, que $\boldsymbol{\omega}^{in}$, vitesse de rotation de la balle, était constante tout au long de la première partie de la trajectoire, et donc égale à celle estimée à $t = 0$.

Après passage dans le repère frappe $\mathcal{R}'$, ces paramètres permettent la simulation du rebond à l'aide de l'équation 4.1, puis, après passage dans le repère monde $\mathcal{R}$ de ré-estimer le vecteur vitesse de translation, noté $\tilde{\mathbf{V}}^{out}$, ainsi que la vitesse de rotation $\tilde{\omega}^{out}$. Nous pouvons alors utiliser le modèle physique introduit chapitre 3 pour simuler la seconde partie de la trajectoire $\mathcal{S}_1$.

Comme précédemment, la minimisation est faite grâce à l'algorithme de Levenberg-Marquardt.

La vitesse de translation $\mathbf{V}_0$ pouvant être estimée sans prise en compte du rebond avec une erreur faible, $0,02\ m/s$ en moyenne dans nos expérimentations du chapitre 3, nous avons fait le choix de conserver cette approche et de ne ré-estimer que la vitesse de rotation. Pour cela, nous fixons cette vitesse de translation, et nous minimisons l'erreur quadratique moyenne entre les positions simulées et réelles sur l'ensemble de la trajectoire avec comme seul paramètre libre la vitesse de rotation de la balle ω_0. Le schéma 4.11 illustre les deux étapes successives permettant l'estimation précise de la vitesse de rotation de la balle.

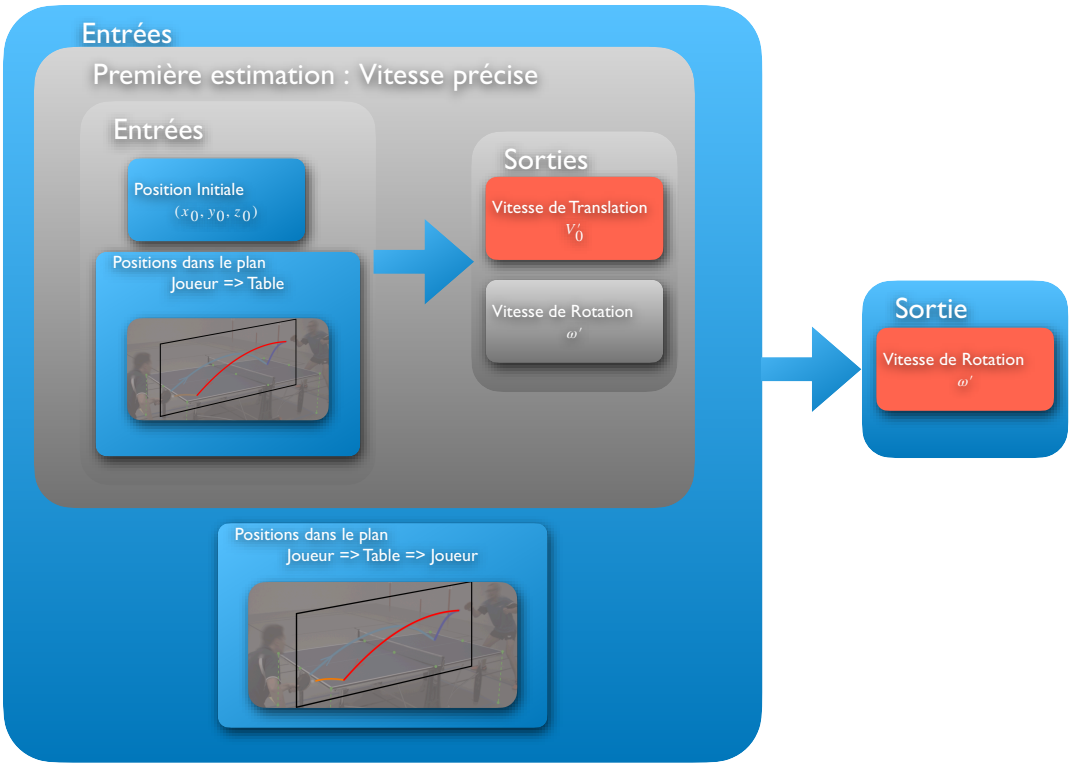


FIGURE 4.11 – Ré-estimation de la vitesse de rotation en utilisant le rebond en $\mathcal{I}_1$.

4.2.3 Résultats expérimentaux

Pour cette section, nous utilisons principalement le jeu de données `Synthétiques` qui est le seul pour lequel nous connaissons les vitesses de rotation.

Afin de montrer le gain apporté par l'utilisation du modèle de rebond, nous rappelons également l'erreur de rotation obtenue au chapitre précédent, c'est-à-dire en utilisant uniquement la trajectoire avant rebond.

Le tableau 4.1 présente les différentes erreurs moyennes obtenues pour chaque type de coup sur le jeu de données `Synthétiques`. Les colonnes "Vitesse de translation", et "Vitesse de rotation sans rebond" correspondent aux erreurs obtenues au chapitre précédent sur la seule section $\mathcal{S}_0$. La dernière colonne correspond aux erreurs obtenues en utilisant le modèle de rebond sur les deux sections $\mathcal{S}_0$ et $\mathcal{S}_1$.

TABLE 4.1 – Erreurs moyennes pour chaque type de coup avec et sans utilisation du modèle de rebond

Coup	Vitesse de translation (m/s)	Vitesse de rotation sans rebond : $\mathcal{S}_0$ (rps)	Vitesse de rotation avec rebond : $\mathcal{S}_0$+$\mathcal{S}_1$(rps)
Top Spin	0,03	10,65	**3,17**
Contre-attaque	0,01	6,47	**1,40**
Poussette	0,02	**1,72**	5,87
Moyenne	0,02	6,61	**3,48**

Pour une `Contre-attaque`, la Figure 4.12 représente le résultat de la rétroprojection de la balle sur une séquence, ainsi que la trajectoire obtenue en utilisant le modèle de rebond pour estimer correctement la vitesse de rotation de la balle. Sur cette séquence, l'erreur de vitesse de translation est de $0,15\ m/s$ et l'erreur de vitesse de rotation est de $0,35\ rps$.

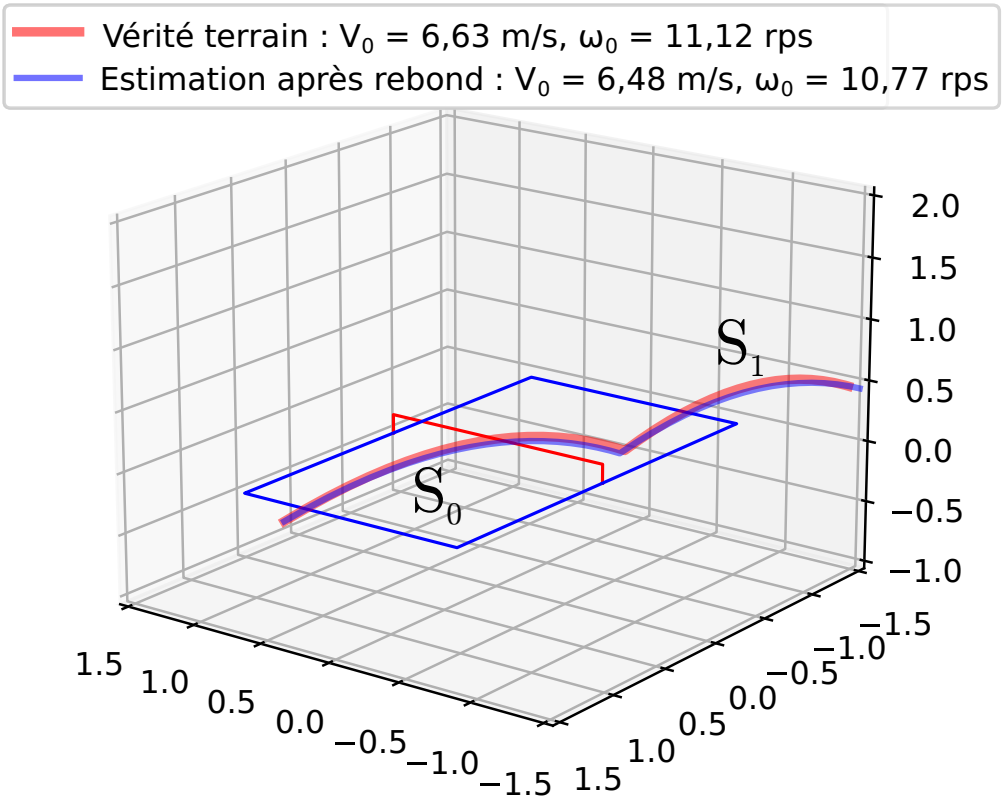


FIGURE 4.12 – Exemple de rétroprojection 3D, et les paramètres cinématiques extraits pour un `Top Spin` en exploitant le rebond sur la table.

Comme nous pouvons le voir, l'exploitation du rebond sur la table permet globalement d'améliorer l'estimation des paramètres cinématiques : nous observons une diminution de l'erreur de vitesse de rotation moyenne de 47%, qui passe de $6,61$ *rps* à $3,48$ *rps*.

Sur les `Top Spin` notamment, dont la rotation est très forte, la vitesse de rotation estimée en utilisant uniquement la section avant rebond n'était pas toujours correcte. Cette erreur, passe de $10,65$ *rps* à $3,17$ *rps* : la rotation étant élevée, elle influence fortement l'angle de sortie. Il est donc logique que la prise en compte du rebond améliore l'estimation des paramètres cinématiques.

Au contraire, pour les `poussettes`, dont la vitesse de rotation est très faible, nous observons que l'erreur estimée en utilisant le rebond est supérieure à l'erreur obtenue en utilisant uniquement la trajectoire avant l'impact. L'erreur de vitesse de rotation passe ainsi de $1,72$ *rps* à $5,87$ *rps*. L'angle de sortie est peu impacté par la rotation. La prise en compte du rebond a donc moins d'importance.

Lorsque nous analysons la trajectoire dans son ensemble, nous effectuons plusieurs étapes consécutives : estimation de positions 3D, estimation de vitesse de translation en utilisant une portion de trajectoire, puis vitesse de rotation en étudiant l'ensemble de la trajectoire. À chaque étape, nous cumulons des erreurs sans réellement ajouter d'information pertinente lorsque la balle tourne lentement.

Au chapitre précédent, nous avions présenté sous forme de diagramme "en violon" la répartition des erreurs de translation et rotation pour chaque type de coup. De la même façon, nous présentons ici la répartition des erreurs de rotation à la Figure 4.13. Afin de représenter les différences d'estimation avant et après rebond, ceux-ci sont mis côte à côte. La partie gauche des violons représente l'estimation de la densité des erreurs de rotation obtenues en utilisant uniquement la première partie de la trajectoire, et la partie droite représente celle obtenue en utilisant l'intégralité de la trajectoire.

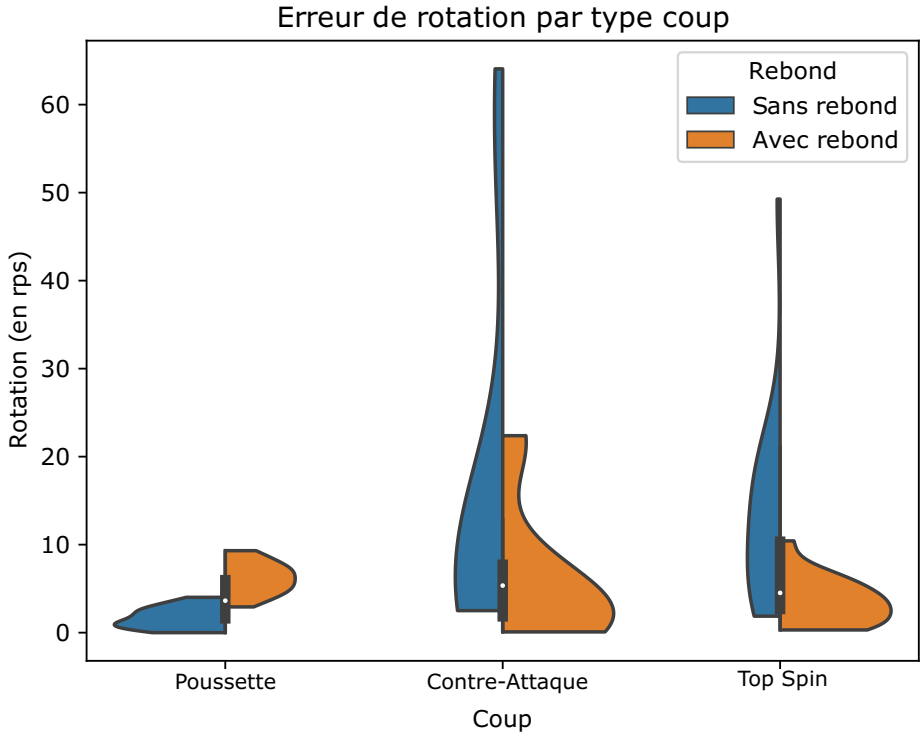


FIGURE 4.13 – Graphique en violon de l'erreur d'estimation de la vitesse de rotation. Nous comparons la répartition des erreurs avant et après la prise en compte du rebond.

En regardant la répartition des données, nous observons plusieurs choses. Comme sur le tableau 4.1, les erreurs commises sur les vitesses de rotation sur les `Poussettes` augmentent légèrement, cependant, leur variance est très

faible, avec un maximum de $9,10\,rps$. Pour les `Top Spins` et `Contre-Attaques`, la première observation est que les erreurs maximales obtenues sont inférieures en exploitant le rebond. L'erreur maximale obtenue pour un `Top Spin` passe de plus de $60\,rps$ à $10,53\,rps$. Le constat est le même pour les `Contre-Attaquess`, qui passent d'une erreur maximale de $50\,rps$ à $3,43\,rps$.

Ensuite, la seconde observation est liée à la forme globale du violon, c'est-à-dire la répartition des erreurs. Avant rebond, les erreurs de rotation étaient réparties de manière plutôt uniforme une large plage de données. Après rebond, la plage de donnée est réduite, mais nous observons également que l'erreur moyenne est proche de zéro.

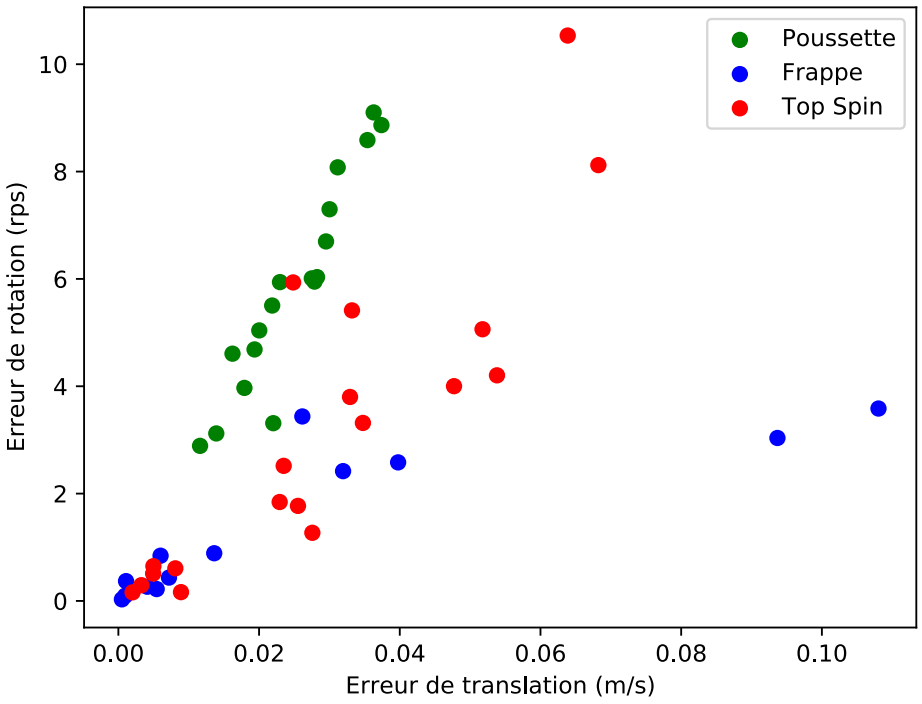


FIGURE 4.14 – Répartition des erreurs de vitesse de translations par rapport aux vitesses de rotations en prenant en compte le rebond

Comme au chapitre précédent, nous illustrons avec la figure 4.14 les erreurs de rotations par rapport aux erreurs de translations pour chacun des trois types de coups. Sans prendre en compte le rebond, il n'y avait pas de corrélation entre l'erreur de rotation et l'erreur de translation obtenue. En prenant en compte le rebond, nous observons une corrélation linéaire entre ces deux erreurs. Nous avons obtenu comme coefficient de corrélation pour les `Top Spin` : $0,87$, pour les `Contre-Attaque` : $0,82$ et pour les `Poussette` :

0,93. On observe également que les séquences ayant les erreurs en translation les plus faibles correspondent également à celles ayant les erreurs les plus faibles en rotation.

Que ce soit avec le tableau 4.1, le graphique en violon Figure 4.13, ou la répartition des erreurs Figure 4.14, il est évident que pour les coups dont la rotation est rapide, comme les `Top Spins`, l'utilisation du rebond apporte un gain significatif dans l'estimation des paramètres cinématiques.

4.2.4 Bilan de l'utilisation du rebond balle/table

Dans cette section, nous avons exploité le rebond de la balle sur la table à l'aide du modèle de NONOMURA, NAKASHIMA et HAYAKAWA, 2010 afin d'analyser la trajectoire dans son ensemble. À partir des positions 3D de la balle, nous avions obtenu, sans utilisation du modèle de rebond, une erreur de vitesse de translation de $0,02\ m/s$, et une erreur de vitesse de rotation de la balle de $6,61\ rps$. Après utilisation du rebond, cette erreur de rotation est quasiment divisée par 2.

L'analyse des résultats obtenus après utilisation du rebond permet de vérifier que globalement les gains sur l'estimation des paramètres cinématiques sont d'autant plus importants que la rotation est forte. Les coups comme les `Top Spins` ou les `Contre-attaques` sont les coups bénéficiant le plus de cette exploitation globale de la trajectoire.

Dans la section suivante, nous exploitons ces paramètres cinématiques pour déduire les forces exercées par la raquette d'un joueur sur la balle lors d'une frappe. Tout comme pour le rebond sur la table, un modèle physique permettra de déduire les paramètres liés à la raquette, c'est-à-dire l'angle de frappe et la vitesse de translation, indicateurs clefs de performance pour les joueurs.

4.3 Prise en compte du rebond balle/raquette dans l'estimation des paramètres cinématiques de la raquette

En utilisant un modèle physique de rebond sur la table, nous avons, dans la section précédente, affiné l'estimation des paramètres cinématiques obtenus à partir d'un ensemble de positions 3D de la balle. De la même manière,

nous allons dans cette section exploiter le rebond de la balle sur la raquette et un modèle physique permettra de faire le lien entre les paramètres cinématiques avant impact et après impact. Dans le cas de l'interaction balle/table, le modèle ne dépendait que des conditions précédant l'impact, c'est-à-dire la vitesse de rotation, la vitesse de translation, et l'angle d'incidence. Dans le cas balle/raquette, d'autres éléments sont à prendre en compte qui rendent plus complexe cette étude. Contrairement à la table, la raquette n'est pas immobile dans le repère monde $\mathcal{R}$, et sa vitesse entre en compte dans les calculs, de plus, son COR dépend des matériaux qui la composent.

Il faut noter que la connaissance des paramètres cinématiques de la balle conjointement à ceux de la raquette permet d'établir une relation entre les vitesses de translation et de rotation de la balle et le geste du joueur. À terme, cela conduira à des indicateurs de performance du joueur.

Dans cette section, nous allons présenter l'estimation de l'orientation de la raquette et sa vitesse lors de la frappe afin de qualifier l'efficacité d'un coup.

4.3.1 Modèle de rebond balle/raquette

Lors d'une frappe effectuée par le joueur, les paramètres cinématiques de la balle après impact dépendent de l'angle entre la surface de la raquette et la trajectoire de la balle avant l'impact et déterminent sa trajectoire après impact.

Les matériaux constituant la raquette sont importants et de différentes natures. La structure en bois, la surface en caoutchouc (ou mousse) M. VARENBERG et A. VARENBERG, 2012 ; J. Q. LIU et al., 2014 ; RINALDI et al., 2019, appelée "plaque", le type de colle utilisé, et même la température de la salle de sport, déterminent le coefficient de friction entre la balle et la raquette. Le choix de ces matériaux est très important pour un joueur de haut niveau, et dépend essentiellement de sa stratégie de jeu. Un joueur offensif aura tendance à privilégier une raquette qui permet de donner beaucoup d'effet lors d'une frappe pour mettre l'adversaire en difficulté. À l'inverse, un joueur plutôt défensif aura tendance à privilégier une surface de frappe qui atténue les effets de la balle ce qui permet un meilleur contrôle de la balle. Notons également qu'une raquette possède deux faces, généralement une face rouge et une face noire et qu'il est possible que chacune des deux surfaces n'ait pas le même COR.

Nous allons, dans cette section, exploiter les paramètres cinématiques de la balle estimés précédemment sur l'ensemble de la trajectoire pour obtenir

avec précision la vitesse et l'angle de frappe de la raquette lors d'une frappe. La Figure 4.15 synthétise les pré-requis nécessaires pour obtenir les paramètres liés à la raquette : angle de frappe et vitesse de translation. Ceux-ci sont obtenus à l'aide d'un modèle de rebond sur la raquette et des paramètres extraits avant et après la frappe du joueur.

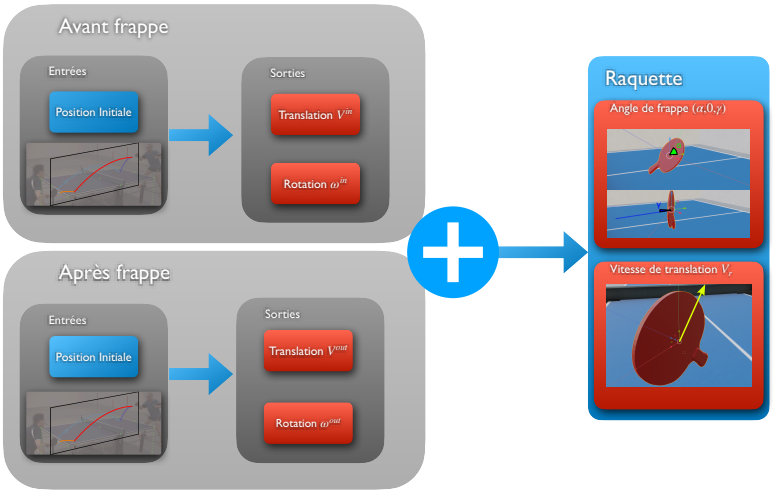


FIGURE 4.15 – Schéma représentant l'obtention des paramètres liés à la raquette avec utilisation du modèle de rebond.

Le modèle de rebond balle/raquette utilisé est celui défini dans (NONOMURA, NAKASHIMA et HAYAKAWA, 2010). L'ensemble des calculs de cette partie est effectué dans un repère lié à la raquette et noté $\mathcal{R}''$, son origine correspond au centre de la plaque, les axes x'' et z'' sont dans le plan de la raquette, le premier horizontal, le second vertical, y'' est orthogonal à la raquette comme illustré Figure 4.16.

Pour définir l'orientation de la raquette en coordonnées sphériques, nous utilisons deux angles α et γ qui correspondent respectivement à la longitude et à la colatitude (voir Figure 4.17) et nous notons $\boldsymbol{\theta} = (\alpha, 0, \gamma)^t$.

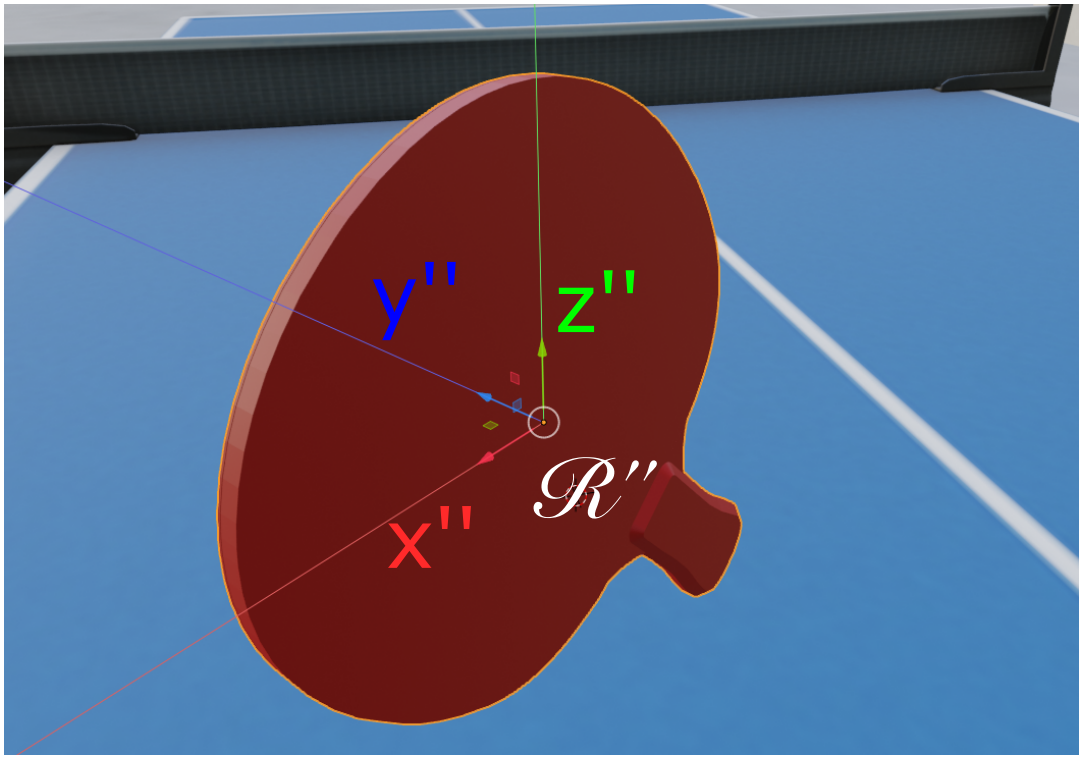


FIGURE 4.16 – Représentation du repère raquette $\mathcal{R}''$.

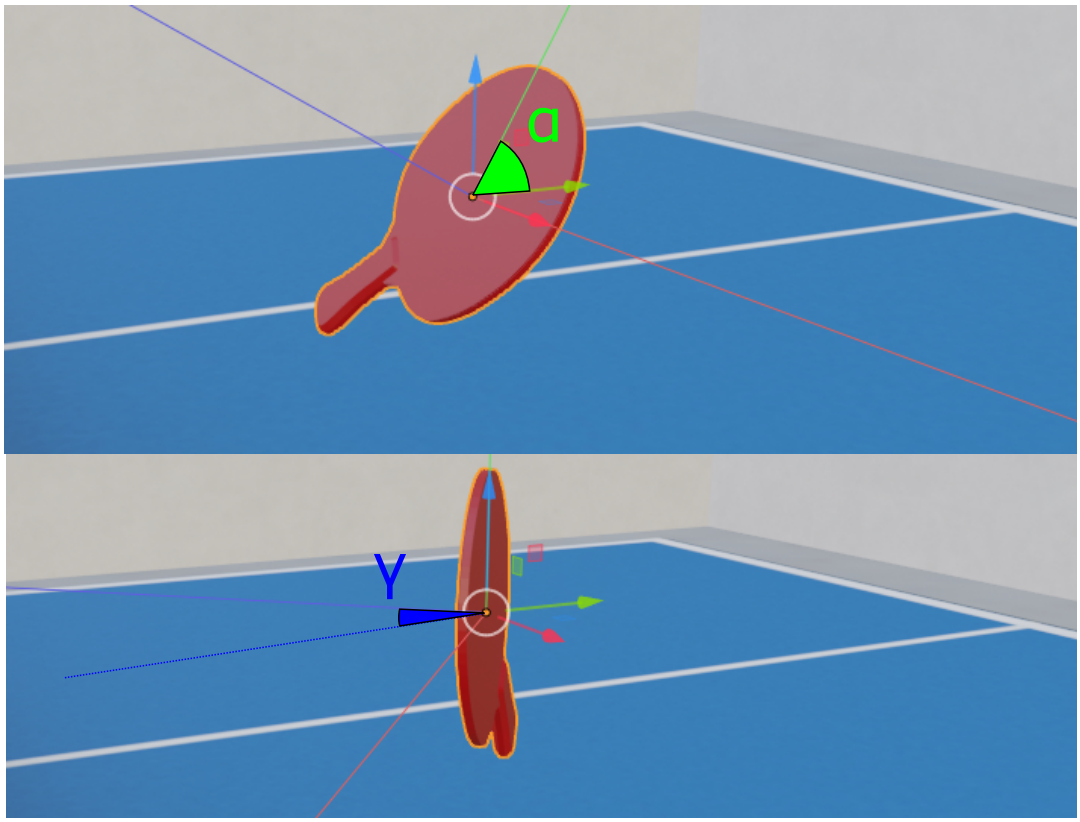


FIGURE 4.17 – Orientation de la raquette. α correspond à l'orientation verticale de la raquette, et γ correspond à l'orientation latérale de la raquette.

Au moment de l'impact sur la raquette, dans le repère raquette $\mathcal{R}''$, la vitesse de la balle est notée $\mathbf{V}^{in''}$, la rotation de la balle $\boldsymbol{\omega}^{in''}$. Ce modèle utilise des hypothèses proches de celles faites dans le cas du rebond sur la table :

— Le point de contact balle/raquette est unique, et le rebond ne dépend que des paramètres cinématiques de la balle à l'instant du contact $\mathcal{I}_2$ et des caractéristiques de la raquette. La surface de la raquette en mousse interdit tout glissement ou roulement. Il n'y a donc pas de décalage entre point d'impact et de rebond contrairement au cas du rebond sur la table.

— Comme dans le chapitre précédent, la balle se déplace dans le plan vertical et la relation entre la vitesse avant et après collision est donnée, pour leurs composantes en z'' par $v_z^{out''} = \epsilon_r \, v_z^{in''}$, où ϵ_r correspond au COR de la raquette. Comme dans le cas de la table, nous prendrons par défaut la valeur $\epsilon_r = 0,81$ proposée dans (NONOMURA, NAKASHIMA et HAYAKAWA, 2010).

— Le rebond est effectué dans la "direction" du vecteur vitesse de translation de la balle (*i.e.* $\mathbf{V}^{in''}$ et $\mathbf{V}^{out''}$ sont tous les deux dans le plan vertical défini par la trajectoire de la balle).

Les vitesses de translation et de rotation avant et après rebond sont liées par les équations suivantes :

$$\mathbf{V}^{out''} = \mathbf{A''}_V \mathbf{V}^{in''} + \mathbf{B''}_V \boldsymbol{\omega}^{in''} \tag{4.8}$$

$$\boldsymbol{\omega}^{out''} = \mathbf{A''}_\omega \mathbf{V}^{in''} + \mathbf{B''}_\omega \boldsymbol{\omega}^{in''} \tag{4.9}$$

où les matrices $\mathbf{A''}_V$ et $\mathbf{B''}_V$ sont définies par :

$$\mathbf{A''}_V = \begin{bmatrix} 1-\frac{k}{m} & 0 & 0 \\ 0 & 1-\frac{k}{m} & 0 \\ 0 & 0 & -\epsilon_r \end{bmatrix}, \mathbf{B''}_V = \frac{k}{m} \begin{bmatrix} 0 & r & 0 \\ -r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.10}$$

et $\mathbf{A''}_\omega$ et $\mathbf{B''}_\omega$ par :

$$\mathbf{A}''_{\omega} = \frac{k}{I}\begin{bmatrix} 0 & -r & 0 \\ r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad ,\mathbf{B}''_{\omega} = \begin{bmatrix} 1-\frac{k}{I}r^2 & 0 & 0 \\ 0 & 1-\frac{k}{I}r^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

où

— $I = \frac{2}{3}\, m.r^2$ est le moment d'inertie qui dépend du rayon $r = 2\ cm$ de la balle et de sa masse $m = 2,7\ g$

— k est le coefficient de frottement de la raquette. Nous avons utilisé dans nos expérimentations la valeur donnée dans NONOMURA, NAKASHIMA et HAYAKAWA, 2010 : $k = 1,9.10^{-3}$

Le coefficient de restitution ϵ_r n'est pas le même en fonction du joueur, et peut varier en fonction de la face utilisée lors de la frappe. Il dépend du type de raquette utilisé, épaisseur de la mousse, type de caoutchouc, ou même la température extérieure. Il en est de même pour k, et nous utiliserons les constantes que nous avons données précédemment pour ces deux coefficients à défaut de les connaître précisément pour chaque joueur. Ce choix sera discuté dans la conclusion de ce chapitre.

Notons, que comme nous travaillons dans le repère raquette $\mathcal{R}''$, la vitesse de celle-ci dans le repère monde $\mathcal{R}$ n'apparaît pas explicitement. Elle sera calculée dans la section 4.3.2.

Afin d'illustrer l'influence de l'orientation de la raquette au moment de la frappe sur la trajectoire de la balle, nous présentons Figure 4.18 les résultats de quelques simulations obtenues en faisant varier la longitude (α) et la colatitude (γ) de la raquette dont le centre est placé à la position $(1,50 \quad 0,20 \quad 0,22)^t\ m$ dans le repère monde $\mathcal{R}$, sa vitesse est nulle : $(0 \quad 0 \quad 0)^t\ m/s$. Les vitesses de translation et de rotation de la balle, toujours dans $\mathcal{R}$, sont fixées :

$\mathbf{V}^{in} = (-0,5 \quad 2,0 \quad 1,0)^t\ m/s$ et $\boldsymbol{\omega}^{in} = (32,64 \quad 0,0 \quad 0,0)^t\ radians/sec.$

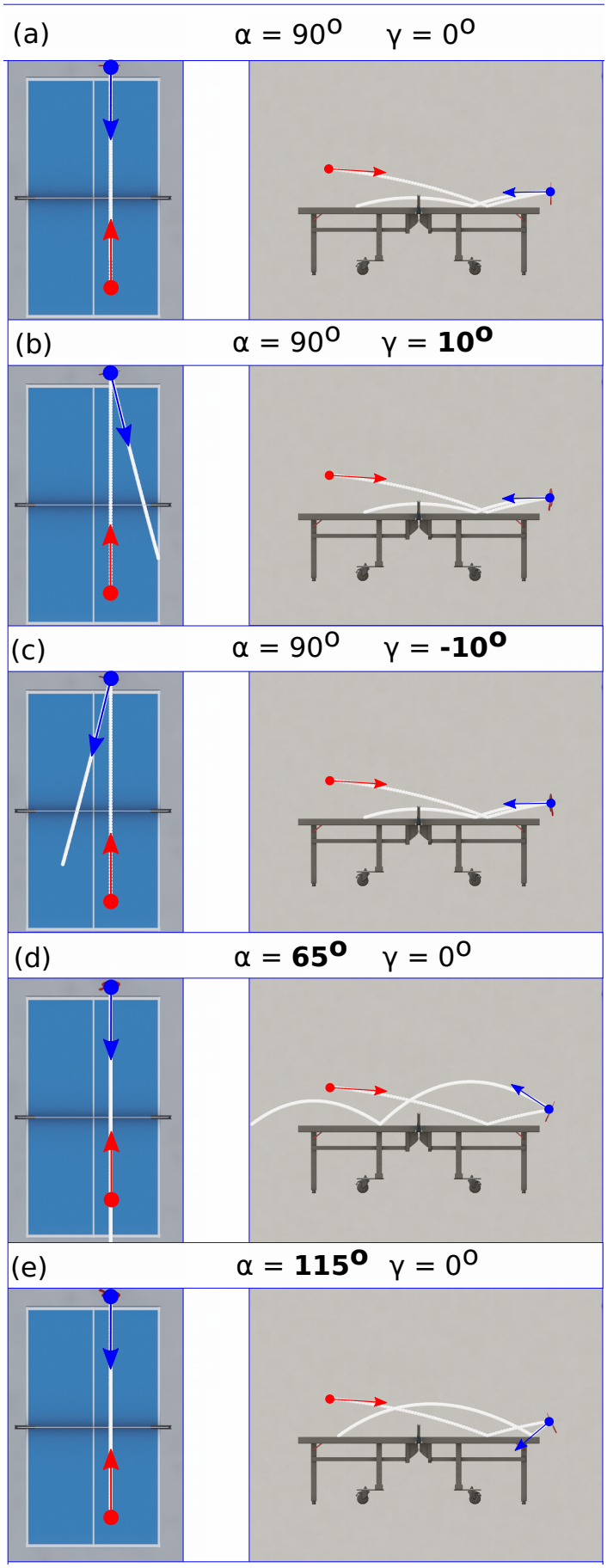


FIGURE 4.18 – Exemple de trajectoires avec rebond balle/raquette puis balle/table pour différents angles de frappe.

De la même façon, nous illustrons l'influence de la vitesse de la raquette sur la trajectoire de la balle sur la figure 4.19. Les vitesses de translation et de rotation avant rebond sur la raquette sont celles utilisées précédemment.

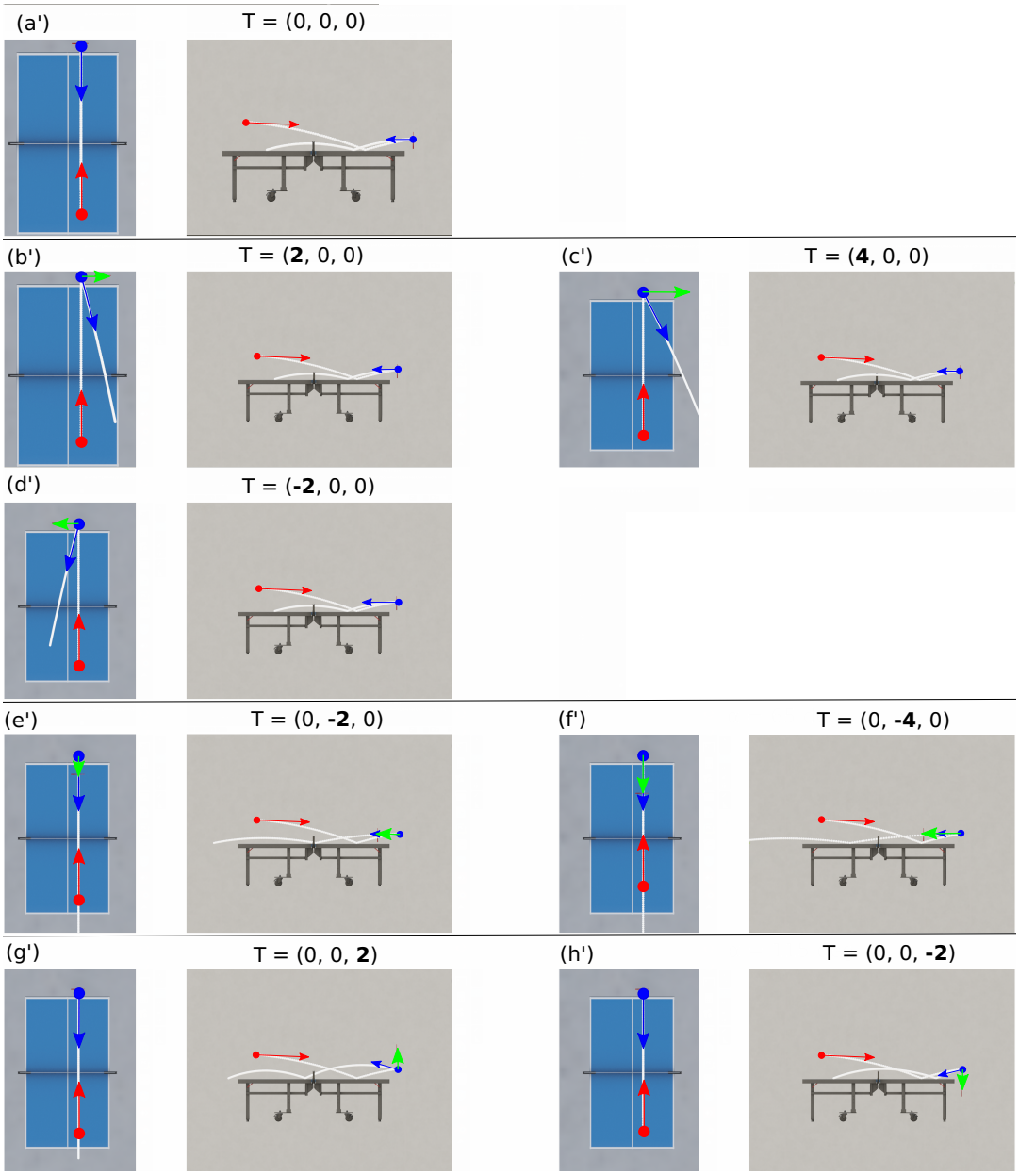


FIGURE 4.19 – Exemple de trajectoires avec rebond balle/raquette puis balle/table pour différentes vitesses de frappe.

Les résultats numériques correspondant à ces figures sont donnés dans le tableau 4.3 pour la partie correspondant aux variations de vitesses et dans le tableau 4.2 pour celle correspondant aux variations d'angles de frappe.

TABLE 4.2 – Vitesses de translation et de rotation d'une balle après impact sur la raquette pour différents angles de frappe. La colonne de gauche indique la direction du coup et la vignette donnant sa visualisation en Figure 4.18, la vitesse de la raquette est toujours nulle.

	Raquette		**Balle** (après impact)	
	Angle θ	**Vitesse $\mathbf{V}_r$**	**Vitesse V^{out}**	**Rotation ω^{out}**
	(deg)	(m/s)	(m/s)	(rad/s)
Orientation de la raquette / filet	$\begin{pmatrix}\alpha\\0\\\gamma\end{pmatrix}$	$\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}$	$\begin{pmatrix}V_x^{out}\\V_y^{out}\\V_z^{out}\end{pmatrix}$	$\begin{pmatrix}\omega_x^{out}\\\omega_y^{out}\\\omega_z^{out}\end{pmatrix}$
Parallèle (a)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}0,0\\-5,0\\-0,3\end{pmatrix}$	$\begin{pmatrix}-15,1\\0,0\\0,0\end{pmatrix}$
Droite (b)	$\begin{pmatrix}90\\0\\\mathbf{10}\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}1,2\\-4,8\\-0,3\end{pmatrix}$	$\begin{pmatrix}-13,8\\\mathbf{-8,2}\\\mathbf{57,0}\end{pmatrix}$
Gauche (c)	$\begin{pmatrix}90\\0\\\mathbf{-10}\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}-1,2\\-4,8\\-0,3\end{pmatrix}$	$\begin{pmatrix}-13,8\\\mathbf{8,2}\\\mathbf{-57,0}\end{pmatrix}$
Bas (d)	$\begin{pmatrix}\mathbf{65}\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}0,0\\-3,9\\2,2\end{pmatrix}$	$\begin{pmatrix}\mathbf{-152,7}\\0,0\\0,0\end{pmatrix}$
Haut (e)	$\begin{pmatrix}\mathbf{115}\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}0,0\\-3,7\\-3,0\end{pmatrix}$	$\begin{pmatrix}\mathbf{124,9}\\0,0\\0,0\end{pmatrix}$

TABLE 4.3 – Vitesses de translation et de rotation d'une balle après impact pour différentes vitesses de frappe (variations sur chacun des trois axes). La colonne de gauche indique la direction du coup et la vignette donnant sa visualisation en Figure 4.19.

	Raquette		**Balle** (après impact)	
	Angle θ	**Vitesse $\mathbf{V}_r$**	**Vitesse V^{out}**	**Rotation ω^{out}**
	(deg)	(m/s)	(m/s)	(rad/s)
Mouvement de la raquette	$\begin{pmatrix}\alpha\\0\\\gamma\end{pmatrix}$	$\begin{pmatrix}v_x\\v_y\\v_z\end{pmatrix}$	$\begin{pmatrix}V_x^{out}\\V_y^{out}\\V_z^{out}\end{pmatrix}$	$\begin{pmatrix}\omega_x^{out}\\\omega_y^{out}\\\omega_z^{out}\end{pmatrix}$
Aucun (a′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\0\end{pmatrix}$	$\begin{pmatrix}0,0\\-5,0\\-0,3\end{pmatrix}$	$\begin{pmatrix}-15,1\\0,0\\0,0]\end{pmatrix}$
Droite (b′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}\mathbf{2}\\0\\0\end{pmatrix}$	$\begin{pmatrix}\mathbf{1,4}\\-5,0\\-0,3\end{pmatrix}$	$\begin{pmatrix}-15,1\\0,0\\\mathbf{-105,5}\end{pmatrix}$
Droite (rapide) (c′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}\mathbf{4}\\0\\0\end{pmatrix}$	$\begin{pmatrix}\mathbf{2,8}\\-5,0\\-0,3\end{pmatrix}$	$\begin{pmatrix}-15,1\\0,0\\\mathbf{-211,1}\end{pmatrix}$
Gauche (d′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}\mathbf{-2}\\0\\0\end{pmatrix}$	$\begin{pmatrix}\mathbf{-1,4}\\-5,0\\-0,3\end{pmatrix}$	$\begin{pmatrix}-15,1\\0,0\\\mathbf{105,5}\end{pmatrix}$
Avant (e′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\\mathbf{-2}\\0\end{pmatrix}$	$\begin{pmatrix}0,0\\\mathbf{-8,6}\\-0,3\end{pmatrix}$	$\begin{pmatrix}-15,1\\0,0\\0,0\end{pmatrix}$
Avant (rapide) (f′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\\mathbf{-4}\\0\end{pmatrix}$	$\begin{pmatrix}0,0\\\mathbf{-12,2}\\-0,3\end{pmatrix}$	$\begin{pmatrix}-15,1\\0,0\\0,0\end{pmatrix}$
Haut (g′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\\mathbf{2}\end{pmatrix}$	$\begin{pmatrix}0,0\\-5,0\\\mathbf{1,0}\end{pmatrix}$	$\begin{pmatrix}\mathbf{90,4}\\0,0\\0,0\end{pmatrix}$
Bas (h′)	$\begin{pmatrix}90\\0\\0\end{pmatrix}$	$\begin{pmatrix}0\\0\\\mathbf{-2}\end{pmatrix}$	$\begin{pmatrix}0,0\\-5,0\\\mathbf{-1,7}\end{pmatrix}$	$\begin{pmatrix}\mathbf{-120,6}\\0,0\\0,0\end{pmatrix}$

4.3.2 Estimation de la vitesse et de l'orientation de la raquette lors de la frappe.

La vitesse et l'orientation de la raquette au moment de la frappe sont deux indicateurs du type et de l'efficacité d'un coup. Ils peuvent être analysés par un joueur ou un entraîneur et permettre de corriger un geste pour le rendre plus efficace.

Grâce aux résultats du chapitre 3 la position 3D ainsi que les vitesses de translation et de rotation de la balle avant et après impact sur la raquette sont connus. Si cette dernière n'est pas directement observée, seule la balle l'est, ses paramètres cinématiques sont donnés par les équations de rebond que nous venons de voir (équations 4.8 et 4.9), qui modélisent ses interactions avec la balle.

Cependant, ces équations sont écrites dans le repère raquette $\mathcal{R}''$, et les paramètres caractérisant la balle sont données dans le repère $\mathcal{R}$. Pour effectuer des calculs utilisant ces deux informations, nous devons donc exprimer la totalité des paramètres dans ce repère monde, qui, de plus, est facilement interprétable pour les entraîneurs ou les sportifs.

Au moment de l'impact de la balle sur la raquette, la position du centre de la balle est $\mathbf{P}_{in} = (x_{in}, y_{in}, z_{in})^t$. Elle correspond à la position de l'origine du repère $\mathcal{R}''$ dans $\mathcal{R}$ et les deux angles sphériques α et γ donnent son orientation. Nous pouvons alors effectuer un changement de base de $\mathcal{R}$ vers $\mathcal{R}''$ grâce à la matrice de passage $\mathbf{M}_{\mathcal{R},\mathcal{R}''}$ qui s'écrit :

$$\mathbf{M}_{\mathcal{R},\mathcal{R}''} = \begin{bmatrix} cos(\gamma) & -sin(\gamma).cos(\alpha) & sin(\gamma).sin(\alpha) \\ sin(\gamma) & cos(\gamma).cos(\alpha) & -cos(\gamma).sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \tag{4.12}$$

et

$$\mathbf{M}_{\mathcal{R}'',\mathcal{R}} = \mathbf{M}_{\mathcal{R},\mathcal{R}''}^{-1} \tag{4.13}$$

est la matrice de passage de la base du repère table à la base du repère monde.

Le passage d'un repère à l'autre peut donc être effectué si l'on connaît une de ces deux matrices ainsi que le vecteur $\mathbf{P}_{in}$ qui permet d'effectuer la translation de l'origine. Si ce vecteur est connu, ce n'est pas le cas de $\boldsymbol{\theta} = (\alpha, 0, \gamma)^t$. Pour l'estimer, nous allons utiliser les vitesses de la balle avant et après rebond sur la raquette dont l'expression est connue dans chacun des deux repères.

Dans $\mathcal{R}$, $(\mathbf{V}^{in}, \boldsymbol{\omega}^{in})$ et $(\mathbf{V}^{out}, \boldsymbol{\omega}^{out})$ sont connus grâce aux résultats du chapitre 3. La vitesse relative de la balle par rapport à la raquette peut être définie

comme la différence entre $\mathbf{V}^{in}$ et la vitesse de la raquette. Cette vitesse, que nous notons $\mathbf{V}_r$, n'est pas connue mais permet d'exprimer la vitesse de la balle au moment de l'impact dans le repère $\mathcal{R}$:

$$\mathbf{V}^{in''} = \mathbf{M}_{\mathcal{R},\mathcal{R}''} \times (\mathbf{V}^{in} - \mathbf{V}_r) \tag{4.14}$$

De même, la rotation de la balle avant impact dans $\mathcal{R}''$ notée $\boldsymbol{\omega}^{in''}$ est égale à :

$$\boldsymbol{\omega}^{in''} = \mathbf{M}_{\mathcal{R},\mathcal{R}'} \times \boldsymbol{\omega}^{in} \tag{4.15}$$

À partir de ces deux vecteurs, nous pouvons utiliser les équations de rebond 4.8 et 4.9 pour obtenir les vitesses de translation et rotation de la balle après impact dans $\mathcal{R}''$ puis d'obtenir les vecteurs correspondants dans $\mathcal{R}$:

$$\begin{aligned} \mathbf{V}^{out} &= \mathbf{M}_{\mathcal{R}'',\mathcal{R}} \times \mathbf{V}^{out''} + \mathbf{V}_r \\ \boldsymbol{\omega}^{out} &= \mathbf{M}_{\mathcal{R}'',\mathcal{R}} \times \boldsymbol{\omega}^{out''} \end{aligned} \tag{4.16}$$

Ces différentes étapes et changements de repères nécessaires au calcul de $(\mathbf{V}^{out}, \boldsymbol{\omega}^{out})$ sont illustrés en figure 4.20.

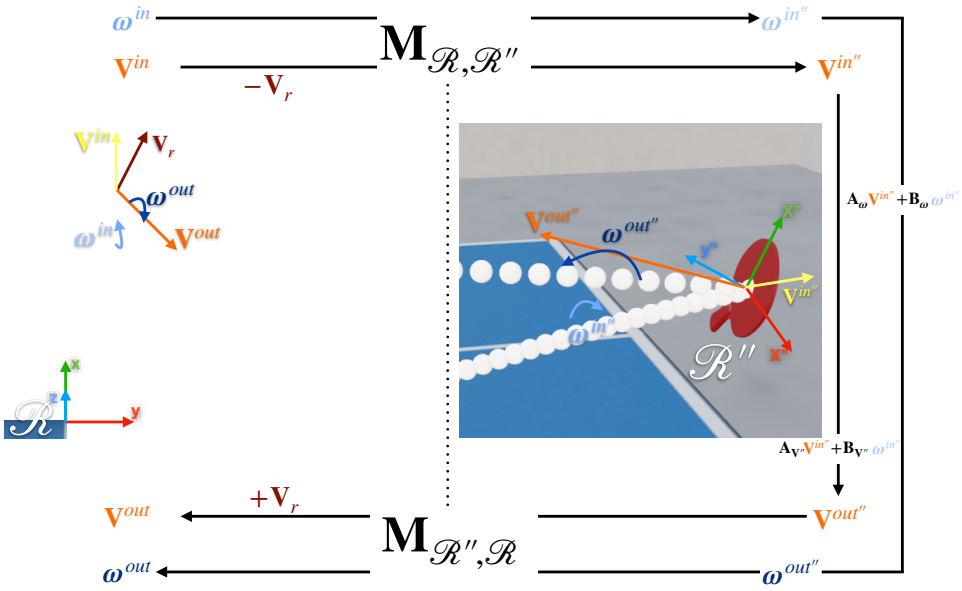


FIGURE 4.20 – Représentation dans $\mathcal{R}$ et $\mathcal{R}''$ des différentes étapes nécessaires au calcul de $(\mathbf{V}^{out}, \boldsymbol{\omega}^{out})$ à partir de $(\mathbf{V}^{in}, \boldsymbol{\omega}^{in})$ en utilisant les matrices de passage et la vitesse de la raquette.

Nous avons donc deux expressions distinctes des paramètres cinématiques de la balle avant et après impact : celle donnée en l'équation 4.16 et celle donnée par 3.8 au chapitre 3. Elles doivent, bien évidemment, être numériquement égales, ce qui va permettre d'estimer les paramètres $\mathbf{V}_r$ et $\boldsymbol{\theta}$ par minimisation de leur différence. Pour cela, nous allons noter $(\tilde{\mathbf{V}}^{out}, \tilde{\boldsymbol{\theta}})$ les paramètres cinématiques de la balle après impact lorsqu'ils sont calculés en utilisant l'équation 4.16 dans laquelle $\mathbf{V}_r$ et $\boldsymbol{\theta}$ doivent être estimés.

Nous définissons l'erreur quadratique $\mathcal{L}(\mathbf{V}_r, \boldsymbol{\theta})$ par :

$$\mathcal{L}(\mathbf{V}_r, \boldsymbol{\theta}) = \frac{E_V}{\sigma_V^2} + \frac{E_\omega}{\sigma_\omega^2} \tag{4.17}$$

où

$$E_V = \|\mathbf{V}^{out} - \tilde{\mathbf{V}}^{out}\|_2^2 \tag{4.18}$$

$$E_\omega = \|\boldsymbol{\omega}^{out} - \tilde{\boldsymbol{\omega}}^{out}\|_2^2 \tag{4.19}$$

σ_V et σ_ω permettent de normaliser, dans $[0, 1]$, E_V et $E\omega$ qui ont des intervalles de variation très différents. Nous avons choisi $\sigma_\omega = 2\pi$ et $\sigma_V = 20$, car la vitesse de la balle est au maximum de 20 m/s (voir chapitre 3). $(\mathbf{V}^{in}$,$\boldsymbol{\omega}^{in})$ et $(\mathbf{V}^{out}, \boldsymbol{\omega}^{out})$ sont donnés par la méthode vue précédemment.

Pour estimer $\boldsymbol{\theta}$ et $\mathbf{V}_r$ nous avons utilisé une méthode *ad-hoc*. En effet, nous n'avons pas réussi à obtenir de résultats satisfaisants en utilisant directement la méthode de Levenberg-Marquardt qui tombait régulièrement dans des minima locaux pour lesquels les estimations n'étaient pas réalistes. Nos tentatives d'initialisations adaptées, comme d'introduction de contraintes sur les paramètres, n'ont pas donné satisfaction. Pour ces raisons, la recherche du minimum de $\mathcal{L}(\mathbf{V}_r, \boldsymbol{\theta})$, est faite en deux étapes : à l'aide d'une recherche par grille, nous fixons une valeur de $\boldsymbol{\theta}$ puis nous cherchons le vecteur $\mathbf{V}_r$ qui minimise l'erreur.

Plus précisément, avec un pas p fixé nous prenons successivement les valeurs de α comprises dans un intervalle $[\alpha_{min}, \alpha_{max}]$, et de même pour γ dans un intervalle $[\gamma_{min}, \gamma_{max}]$, puis nous calculons les matrices $\mathbf{M}_{\mathcal{R},\mathcal{R}'}$ et $\mathbf{M}_{\mathcal{R}',\mathcal{R}}$ correspondantes en utilisant les équations 4.12 et 4.13.

Ensuite, nous initialisons le vecteur $\mathbf{V}_r$ à $(0, 0, 0)^t$. En utilisant les équations 4.14, 4.15 et 4.16, nous obtenons une première estimation de $(\tilde{\mathbf{V}}^{out}, \tilde{\boldsymbol{\theta}})$ et grâce à algorithme de Levenberg-Marquardt nous obtenons la valeur de $\mathbf{V}_r$ qui minimise $\mathcal{L}(\mathbf{V}_r, \boldsymbol{\theta})$ à $\boldsymbol{\theta}$ fixé.

Par itérations successives sur les valeurs de $\boldsymbol{\theta}$ nous obtenons le couple de paramètres $(\mathbf{V}_r, \boldsymbol{\theta})$ qui minimise $\mathcal{L}(\mathbf{V}_r, \boldsymbol{\theta})$. Dans nos expérimentations, nous avons choisi les intervalles de recherche suivants :

$$\begin{cases} [\alpha_{min}, \alpha_{max}] = [-15°, +15°] \\ [\gamma_{min}, \gamma_{max}] = [-30°, 60°] \\ p = 0.5° \end{cases} \tag{4.20}$$

Ces intervalles ont été obtenus expérimentalement comme nous allons le voir dans la section suivante.

4.3.3 Résultats expérimentaux

Afin de valider notre approche, nous avons constitué un nouveau jeu de données `Synthétique avec frappes` qui contient 200 trajectoires de balles générées comme dans le chapitre 3 par le logiciel Blender. Il comprend au total de 62 749 images. Chaque trajectoire est constituée des quatre sections $\mathcal{S}_.$ décrites sur la Figure 4.1. La position et les vitesses de translation et de rotation de la balle au début de la première trajectoire sont tirées au hasard comme pour le jeu de données `Synthétique`, puis le rebond sur la table est simulé en utilisant les équations de la section 4.2. L'impact entre la raquette et la balle a lieu lorsque celle-ci est dans une zone comprise entre 1 m et 1,5 m du centre de la table sur l'axe x. Nous choisissons cette distance aléatoirement en utilisant une loi uniforme.

Lorsque la balle a atteint la distance choisie, nous choisissons un angle de frappe $\boldsymbol{\theta}$ aléatoire. Les intervalles dans lesquels α et γ sont tirés aléatoirement selon une loi uniforme sont ceux des équations 4.20 de la section précédente. Expérimentalement, nous avons observé que lorsque α est choisi dans l'intervalle -15 et 15 degrés, la balle maximise les chances de retomber sur la table. Pour la même raison, l'angle γ est choisi aléatoirement entre -30 et $+60$ degrés. Un angle inférieur à -30 degrés a peu de chances de dépasser le filet lors d'un rebond, et un angle supérieur à 60 degrés a peu de chances de toucher la table.

La vitesse de translation de la raquette est choisie aléatoirement en utilisant une loi uniforme dont les bornes sont 0 m/s et 20 m/s sur chacun des trois axes. La valeur 20 m/s a été choisie car dans les séquences réelles, la vitesse de translation de la balle est en moyenne proche de 10 m/s.

Enfin, nous ne conservons que les trajectoires valides, c'est-à-dire pour lesquelles nous obtenons quatre parties $\mathcal{S}_.$ valides. Si un des rebond n'est pas

valide, car sur la demie-table adverse ou en dehors de la table, nous éliminons la trajectoire et en générons une nouvelle.

Nous sauvegardons la position initiale, les vitesses initiales de translation et de rotation de la balle, les vitesses de translation et de rotation après le rebond sur la raquette, ainsi que la vitesse et l'angle de frappe de la raquette comme vérité terrain.

Le tableau 4.4, obtenu en utilisant la méthode d'optimisation décrite plus haut sur le jeu de données `Synthétique avec frappes`, résume les erreurs moyennes d'estimation des vitesses de translation, de la raquette selon chacun des trois axes (x, y, z), ainsi que celles des angles α et γ.

TABLE 4.4 – Erreurs moyennes d'estimation des vitesses de translation de la raquette, et de l'angle de frappe.

	Vitesse $\mathbf{V}_r$ (m/s)			Angle de frappe $\boldsymbol{\theta}$ (degrés)	
Axe	x	y	z	α	γ
Erreur	0,18	0,04	0,67	5,95	0,21
Moyenne	0,30			3,08	

La plus faible erreur, $0,04\ m/s$, est mesurée sur l'axe y qui correspond au sens de la largeur de la table. Cela est logique puisque notre modèle suppose un déplacement de la balle dans un plan vertical avec peu ou pas d'effets latéraux. Au moment de la frappe, la raquette se déplace donc essentiellement selon les axes x et z et il y a peu de variabilité selon l'axe y.

Nous observons que la vitesse de translation de la raquette est estimée avec une erreur moyenne de $0,18\ m/s$ selon l'axe x qui correspond à la composante du geste dans le sens de la profondeur de la table, c'est-à-dire en direction de l'adversaire. Cette composante de $\mathbf{V}_r$ porte l'essentiel de la vitesse de translation de la balle. Cette dernière étant généralement bien estimée par notre modèle, l'erreur sur sa composante principale est limitée.

L'erreur la plus importante, $0,67\ m/s$, se trouve sur l'axe z qui correspond à un geste vertical et qui permet de donner sa rotation à la balle (voir tableau 4.3). L'essentiel des effets étant donnés par un mouvement de la raquette dans cette direction, il est normal de constater que cette composante du geste est celle qui comprend le plus de variabilité et qui est donc la plus difficile à estimer.

Néanmoins, l'erreur moyenne globale sur les vitesses de translation est bonne et correspond bien aux informations données, pour les types de coups étudiés, par les entraîneurs et les sportifs que nous avons rencontrés.

De la même façon, pour $\boldsymbol{\theta}$, la composante α conditionne l'angle de rebond sur la table selon l'axe z comme nous pouvons le voir sur la Figure 4.18. Les erreurs sur α, et sur z sont donc liées et toutes les deux relativement élevées. Cela est à rapprocher de l'erreur moyenne dans l'estimation des vitesses de rotations de la balle que nous avons obtenue en section 4.2 qui était relativement élevé.

La faible erreur commise sur l'angle γ est à rapprocher de la faible erreur commise selon l'axe y, qui est associé à l'orientation de la trajectoire. Celle-ci, a lieu dans un plan qui est généralement correctement estimé (voir table 4.2).

Les résultats sont donc satisfaisants bien qu'ils fassent apparaître une dissymétrie sur la précision entre les deux familles de paramètres de translation et de rotation. Une des causes est probablement notre méthode d'optimisation qui conduit à une estimation très précise de la vitesse de translation par l'algorithme de Levenberg-Marquardt alors que pour les angles c'est une méthode de recherche par grille avec un pas fixe qui est utilisée. De plus, pour des mesures angulaires, la norme L2 utilisée n'est probablement pas adaptée si l'estimation initiale est trop éloignée de la solution. Une fonction d'erreur utilisant la distance angulaire serait sans doute meilleure en général. Toutefois, dans le cas où les estimations et solutions sont proches, cette approximation reste acceptable.

4.3.4 Bilan de l'utilisation du rebond balle/raquette

Dans cette section, nous avons étudié le comportement de la balle lors d'une frappe. Un impact entre une raquette et une balle se caractérise par des paramètres liés à l'état de la balle (rotation, translation), ainsi que par des paramètres liés à la raquette (angle de frappe, et vitesse de frappe). En utilisant les paramètres cinématiques obtenus précédemment et un modèle physique, nous obtenons sur un jeu de données synthétique de bons résultats avec, en moyenne, une erreur de $0,3\ m/s$ sur les vitesses de frappe et une erreur pour les angles de frappe d'environ $3°$.

Ces résultats sur nos données synthétiques devront être validés sur des données réelles par une série d'expérimentations. À l'heure actuelle, celles que nous avons menées n'ont pas été jugées satisfaisantes car trop peu précises en termes d'angle de frappe : l'inspection visuelle de l'orientation de la raquette dans les séquences réelles montre qu'elle n'est pas suffisamment proche de celle qui est estimée par notre méthode. Nous avons donc décidé de ne pas les inclure dans ce manuscrit. Cependant, nous avons identifié plusieurs raisons à ces difficultés. Tout d'abord, dans nos simulations, les constantes physiques sont connues et non estimées par optimisation. Comme nous l'avons déjà dit, le COR ϵ_r et le coefficient de frottement k dépendent de nombreux paramètres propres à la raquette de chaque joueur et aux conditions de jeux (température, type de raquette, de colle, ...).

L'erreur commise sur l'estimation des angles de frappe et les difficultés rencontrées sur la mise en œuvre sur données réelles nous ont conduit à penser que notre méthode de minimisation de l'erreur doit être améliorée de plusieurs façons :

— Améliorer la fonction d'erreur angulaire pour mieux traiter les cas où l'estimation initiale est éloignée de la solution.
— Remplacer notre schéma d'optimisation par une méthode de descente gradient générique pouvant utiliser une fonction d'erreur mieux adaptée.
— Effectuer l'optimisation sur une séquence longue constituée de plusieurs échanges afin de pouvoir estimer les paramètres liés à la raquette (COR et coefficient de frottement).

4.4 Conclusion

Dans ce chapitre, nous avons étudié le rebond de la balle lorsque celle-ci entre en contact avec une surface.

Dans la section 4.2, nous avons analysé le comportement de la balle lorsque celle-ci touche la table. L'utilisation d'un modèle physique de rebond a montré que la vitesse de rotation avant impact sur la table conditionne l'angle de rebond. Cette liaison a permis d'améliorer significativement notre estimation de la vitesse de rotation, passant ainsi d'une erreur de $6,61\ rps$ à $3,48\ rps$.

Une fois le rebond sur la table analysé, nous sommes capables d'obtenir une estimation satisfaisante des vitesses de translation et de rotation tout au long de la trajectoire de la balle entre deux frappes de joueurs. Dans la section 4.3, nous avons utilisé les paramètres cinématiques juste avant impact et juste après impact pour analyser la frappe du joueur. À l'aide d'un modèle de rebond adapté, et sur des séquences de synthèse, nous avons obtenu une très bonne estimation de la vitesse de la raquette, et donc de la vitesse de frappe du joueur. L'angle de frappe est, lui, moins bien estimé mais les résultats restent bons avec une erreur moyenne de $3,08°$.

Notre approche reste toutefois à valider sur des données réelles, ce qui nécessitera une réécriture partielle de notre méthode d'optimisation.

Les paramètres extraits sont extrêmement intéressants pour les sportifs. En effet, l'angle et la vitesse de frappe d'un joueur sont des informations riches. Dans le cas d'un entraînement régulier, les informations liées à la vitesse de frappe pourraient être utilisées pour évaluer l'énergie déployée par un joueur pour jouer un type de coup et donc juger de son efficacité. Il serait également possible de corriger un geste en étudiant les corrélations entre les coups réussis, c'est-à-dire qui sont gagnants ou correctement placés sur la table, et les angles de frappe. Dans le contexte de compétitions, il est même envisageable d'étudier le geste de l'adversaire pour mieux le contrer. Par exemple, un joueur défensif cherche à forcer l'adversaire à commettre des erreurs et connaître les caractéristiques du jeu adverse lui permettrait d'ajuster son propre jeu. L'exemple le plus simple étant le cas où le défenseur impose une rotation dans le sens opposé au `Top Spin` pour forcer le joueur offensif prendre plus de risque pour accélérer la balle.

Ces informations d'angle et de vitesse de frappe ont donc du sens pour

les sportifs, et pourraient servir pour suggérer des entraînements personnalisés et détecter ou combler les lacunes d'un sportif. Ces paramètres pourraient également être utilisés pour caractériser les coups, et améliorer les outils existants de reconnaissance automatique d'actions. En effet, la reconnaissance d'actions à grain fin, dans le cas des sports de balle reste un domaine complexe. Pour un humain non-expert du domaine, la distinction entre deux coups pourtant différents du point de vue de la cinématique peut être difficile.

Nous avons montré qu'en utilisant des modèles physiques intégrants les rebonds balle/table et balle/raquette, il est possible d'estimer correctement les paramètres cinématiques de la balle, mais également de la raquette, ce qui ouvre des perspectives sur l'estimation de paramètres physiques liés au joueur : énergie dépensée, efficacité du geste.

Deuxième partie

Partie 2 - Reconnaissance d’actions humaines - Benchmark MediaEval

Chapitre 5

Reconnaissance d'actions sportives dans des vidéos - Workshop MediaEval

Sommaire

5.1 Introduction

Nous avons eu l'occasion de participer dans le cadre de cette thèse au Workshop MediaEval pendant 3 ans. Plus précisément, au sein du projet CRISP, nous organisons depuis 2018 la tâche `Sports Video` à MediaEval. Ce projet est une collaboration avec le STAPS de l'université de Bordeaux, le LaBRI de l'Université de Bordeaux et le laboratoire MIA à L'Université de La Rochelle.

Les principales productions liées à ce chapitre sont :

— Un article en tant que co-éditeur de MediaEval
— Trois articles internationaux en tant qu'organisateur pour la tâche de reconnaissance d'actions sportives
— Deux articles internationaux en tant que participant (voir annexes C.1 et C.2)

Dans un premier temps, nous allons présenter en section 5.2 les objectifs du benchmark MediaEval. Section 5.3 nous présenterons en détails la tâche que nous organisons, et le jeu de donnée `TTStroke-21` utilisé, ainsi que l'évolution de la tâche sur les trois années.

Ensuite, nous présenterons section 5.4 un récapitulatif de nos contributions en tant que participants en 2019 et 2020.

Nous terminerons section 5.5 par la conclusion de ces trois années en tant qu'organisateurs et participants, ainsi que les perspectives d'évolution de la tâche `Sports Video`.

5.2 Le benchmark MediaEval

Le benchmark MediaEval [1] (multiMEDIA EVALuation) propose des tâches liées à la recherche, à l'analyse et à l'exploration de contenus multimédia. La participation est ouverte à tout chercheur intéressé pas les domaines du multimédia. MediaEval se concentre spécifiquement sur les aspects humains et sociaux du multimédia, et sur les systèmes multimédias au service des utilisateurs. Les tâches de MediaEval offrent aux chercheurs la possibilité de relever des défis qui réunissent plusieurs modalités comme le domaine visuel (images, vidéos ...), l'aspect textuel (e-mails, textes, tweets, ...) ainsi que l'aspect audio (musiques, extraits sonores ...).

Les tâches proposées pour 2022 sont les suivantes :

- DisasterMM : Multimedia Analysis of Disaster-Related Social Media Data
- Emotional Mario : A Game Analytics Challenge
- FakeNews : Fake News Detection

1. https://multimediaeval.github.io/ et avant 2020 : http://www.multimediaeval.org/

- Medico : Medical Multimedia Task : Transparent Tracking of Spermatozoa
- Musti : Multimodal Understanding of Smells in Texts and Images
- NewsImages : Relating news articles and images
- NjordVid : Fishing Trawler Video Analytics Task
- Memorability : Predicting Video Memorability

→ **Sports Video : Fine Grained Action Detection and Classification of Table Tennis Strokes from videos**

- SwimTrack : Swimmers and Stroke Rate Detection in Elite Race Videos
- Urban Air : Urban Life and Air Pollution

5.3 La tâche Sports Video

La détection et la classification d'actions sportives sont des défis très actuels dans le domaine de l'analyse vidéo. Les principaux jeux de données disponibles centrés sur la classification d'actions humaines comme (SOOMRO, ZAMIR et SHAH, 2012; ABU-EL-HAIJA et al., 2016; KUEHNE et al., 2013) consistent principalement à déterminer à quel sport appartient une séquence d'images.

Le contexte de la tâche Sports Videos[2] est l'analyse et la reconnaissance à grain fin d'actions sportives. La reconnaissance d'action à grain fin est complexe, car la variabilité intra-classe (un même coup pour différents joueurs) peut être forte et la variabilité inter-classe faible (beaucoup de coups sont très similaires).

5.3.1 Jeu de données utilisé pour la tâche

Dans notre cas d'application qui est le tennis de table, il n'existait pas de jeu de données public pour la reconnaissance fine de coups, dans des conditions écologiques, c'est-à-dire non-invasives pour le joueur. Dans le cadre du projet CRISP, a été créée la première base de données annotée sur le tennis de table, nommée TTStrokes21 (P.-E. MARTIN, 2020). La base TTStrokes21 a été élaborée sous la direction de professeurs de sport du STAPS, qui ont établi une taxonomie de 20 coups majeurs, comme le Service coup droit coupé ou le Revers lifté, ainsi qu'une classe de rejet, soit 21 labels.
La figure 5.1 présente quelques séquences pour les coups ainsi que la classe Négatif, de rejet.

2. https://multimediaeval.github.io/editions/2021/tasks/sportsvideo/

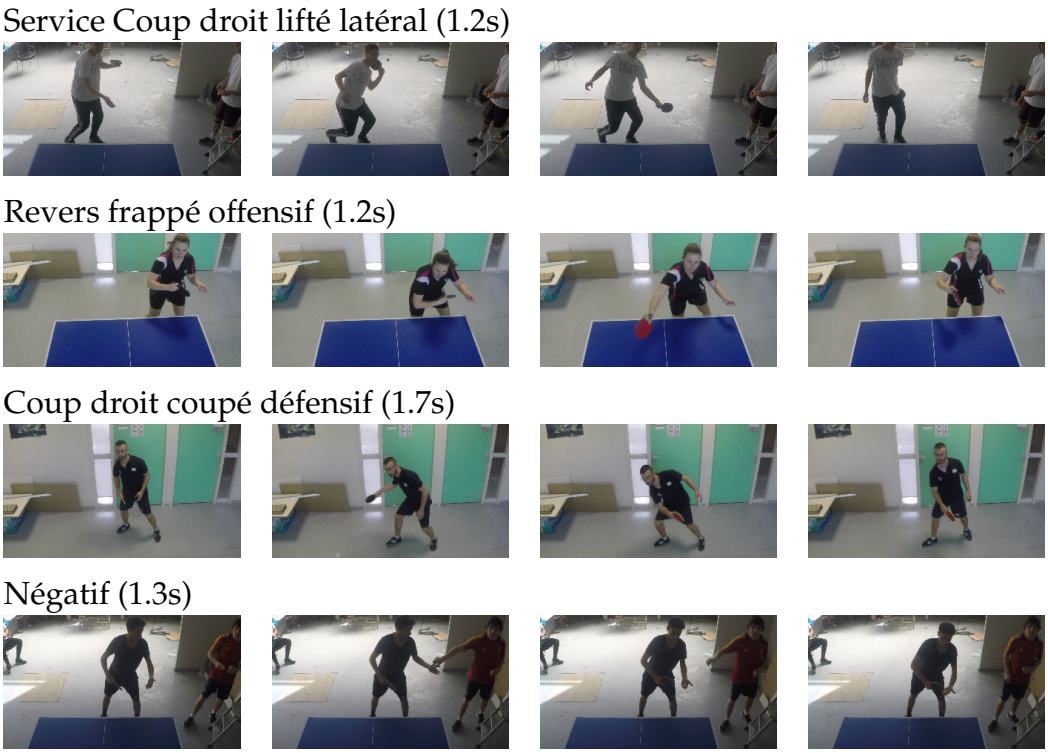


FIGURE 5.1 – Trois exemples de coups de la base `TTStroke-21` ainsi que la classe de rejet (`Négatif`). Sur chaque ligne est affichée une image correspondant respectivement au début, au 1/3, au 2/3 et à la fin de la séquence. On peut noter les conditions très différentes de scènes et d'acquisition de chaque séquence.

La base a été continuellement enrichie avec des séquences de joueurs différents, pour différentes fréquences d'images, résolutions spatiales et points de vue de la caméra. Ces séquences sont enregistrées dans une salle de sport, sans marqueurs, et en éclairage artificiel. Le joueur est filmé dans deux situations : lors de gammes d'entraînement (répétition du même coup) ou dans un contexte de match. Ces vidéos ont été annotées par des joueurs de tennis de table et des experts de la Faculté des sports de l'Université de Bordeaux. Le processus d'annotation a été conçu comme une méthode de crowdsourcing. Les sessions sont supervisées par des joueurs de tennis de table professionnels et des enseignants. Une plate-forme Web à cet effet a été développée pour l'annotation, où l'annotateur repère et étiquette les coups présents dans une séquence. Actuellement, la base comprend 241 vidéos, ce qui représente 369 minutes de séquences de tennis de table à 120 fps et un total de 2 152 annotations. La base est librement disponible sur demande à des fins de recherche.

5.3.2 Objectifs de la tâche

En cours depuis 2019, la tâche `Sports Video` s'est concentrée les deux premières années sur la classification de vidéos segmentées temporellement extraites de la base `TTStroke-21`. Depuis l'édition 2021 de la tâche, deux sous-tâches sont proposées. Le jeu de données a également été enrichi de nouveaux échantillons de coups plus diversifiés.

— *La sous-tâche 1* est une tâche de **classification** : les participants doivent construire un système de classification qui étiquette automatiquement les segments vidéo en fonction du coup effectué. Il y a 20 classes d'attaque possibles.

— *La sous-tâche 2* est une nouvelle tâche proposée depuis 2021 : l'objectif est de **détecter** si une attaque a été réalisée, quelle que soit sa classe, et d'en extraire les **limites temporelles**. L'objectif est de pouvoir distinguer les moments d'intérêt dans un jeu (les joueurs exécutant des coups) des moments non-pertinents (ramasser la balle, faire une pause...). Cette sous-tâche peut être une étape préliminaire à la reconnaissance ultérieure d'un coup qui a été effectué.

Plus de détails sont disponibles en annexe dans la description de la tâche (voir annexe B).

5.4 Nos participations à la tâche

En plus d'être organisateurs de la tâche, nous y avons aussi participé en 2019 et 2020. Lors de ces deux éditions, la tâche ne comportait que la sous-tâche 1 (tâche de classification). Pour l'édition 2019, nous avons proposé une méthode exploitant les singularités du flot optique, appelés *points critiques* (sous-section 5.4.1).
Pour l'édition 2020, nous avons proposé une méthode utilisant des Images Dynamiques, qui permettent de "résumer" l'information de mouvement d'une vidéo en une seule image (sous-section 5.4.2).

5.4.1 Une méthode utilisant les singularités du flot optique

Nos travaux étant principalement axés sur l'analyse visuelle du mouvement, nous avons fait le choix la première année d'exploiter le flot optique. Nous présentons ici un récapitulatif de notre participation à l'édition 2019 (CALANDRE, PÉTERI et MASCARILLA, 2019). Pour plus de détails sur ces travaux, l'article est disponible en annexe C.1.

Nous avons proposé une approche basée sur le flot optique. Nous avons utilisé l'implémentation récente par apprentissage profond de (SUN et al., 2018), qui permet d'avoir des contours de mouvement très nets, et qui est rapide à calculer. Nous illustrons le résultat d'un calcul du flot optique obtenu sur une séquence test par la figure 5.2.

FIGURE 5.2 – Représentation du mouvement entre deux images sur `TTStroke-21` à l'aide du flot optique. La couleur encode le sens du vecteur mouvement et la saturation sa norme.

Nous obtenons ainsi en chaque pixel (x_1, x_2) les composantes horizontales et verticales du vecteur mouvement (respectivement noté $U(x_1, x_2)$ et $V(x_1, x_2)$). L'utilisation du flot optique est très employée pour la reconnaissance d'actions humaines. Il est utilisé en tant que descripteur pour encoder le mouvement apparent dans le voisinage de points d'intérêt préalablement détectés (`HOF` (LADJAILIA et al., 2020), `MBH` (WANG et al., 2011), etc.). Dans l'approche présentée ci-dessous, le flot optique est utilisé comme un outil permettant la détection d'éléments d'intérêt dans des vidéos. Ces éléments d'intérêt correspondent aux points décrivant une forte déformation du champ vectoriel associé au flot optique. Plus précisément, nous nous intéressons aux singularités du champ de vecteurs, aussi appelées points critiques (BLANC, LINGRAND et PRECIOSO, 2017), qui correspondent à des points particuliers où le flot optique présente une forte divergence ou une forte rotation.

Ces points critiques sont obtenus en projetant localement $U(x_1, x_2)$ et $V(x_1, x_2)$ sur une base de Legendre de degré 1 :

$$\begin{pmatrix} U(x_1, x_2) \\ V(x_1, x_2) \end{pmatrix} \simeq \mathbf{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \mathbf{b}$$

.

A et b sont respectivement la matrice et le vecteur de projection sur la base de Legendre. Pour plus de détails, veuillez vous référer à l'annexe C.1.

Les points critiques du flot optique peuvent ensuite être localement analysés à partir de A, notamment de ses valeurs propres (λ_1, λ_2), son déterminant $\Delta(A)$ et sa trace $tr(A)$. Les différents types de singularités sont illustrés sur la figure 5.3.

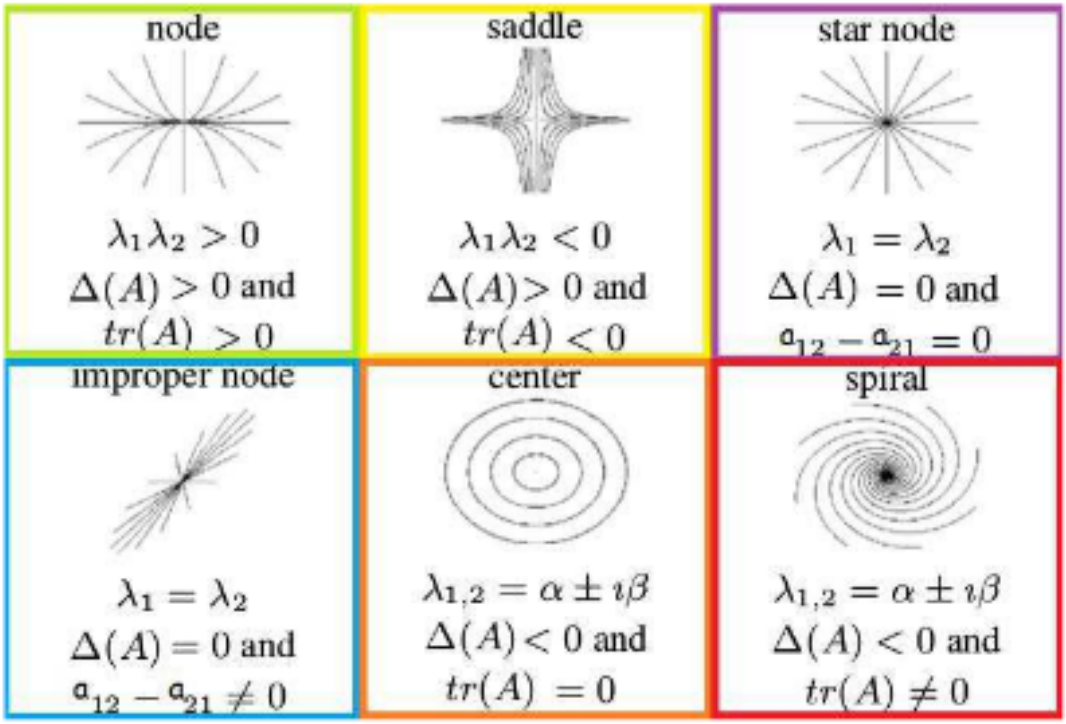


FIGURE 5.3 – Différents types de singularités du flot optique (BLANC, LINGRAND et PRECIOSO, 2017)

Pour effectuer la classification d'une séquence contenant un coup de tennis de table, nous utilisons une approche de type `Bag of Word` sur les points critiques extraits, avec :

— Un K-Means à 6 clusters pour les coefficients de projections sur les polynômes de Legendre
— Un K-Means à 8 clusters sur les descripteurs HoG
— Une information spatiale indiquant dans lequel des 4 quadrants se situe le point critique

Nous obtenons ainsi un vecteur descripteur de taille compris entre 6 (essai 1 avec uniquement les coefficients de projection) et 18 (utilisation de l'ensemble des informations), et effectuons la classification de ce vecteur à l'aide d'un SVM avec validation croisée.

TABLE 5.1 – Taux de classsification pour chacune des méthodes. Toutes utilisent les coefficients de Legendre, et un SVM, et la dernière méthode utilise un SVM équilibré.

Méthode	Entraînement	Test
Coeff. Legendre puis SVM	20,29%	7,06%
Coeff. Legendre + Position puis SVM	56,50%	12,99%
Coeff. Legendre + Position + HoG puis SVM	69,50%	12,99%
Coeff. Legendre + Position + HoG puis SVM Équilibré	54,77%	16,10%

Le nombre d'instances de chaque classe n'étant pas identique, certaines classes étant rares, nous effectuons également une pondération du SVM pour compenser la disparité des classes.

Nous présentons dans le tableau 5.1 les résultats des quatre essais soumis à MediaEval 2019, avec des méthodes différentes en fonction de l'information utilisée dans le descripteur.

Le meilleur résultat est obtenu avec les coefficients de Legendre, le descripteur HoG, et un SVM équilibré pour compenser les classes rares dans le jeu de données.

La classification pour chaque classe est illustrée dans la figure 5.4.

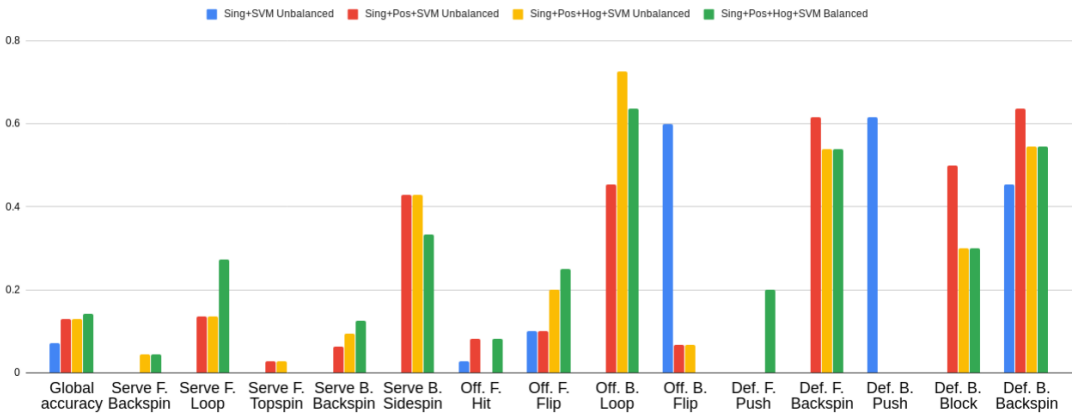


FIGURE 5.4 – Répartition des erreurs de classification pour chaque type de coup pour chacune des méthodes utilisées.

La précision globale sur la base de test est de 16,10%. Nous observons également une précision de 54,77% sur la base d'entraînement. Cette approche, bien qu'intéressante d'un point de vue méthodologique, ne permet pas de classifier avec précision les différentes actions présentes dans le jeu de données. Cela peut s'expliquer pour différentes raisons : premièrement, nous utilisons les coefficients des singularités pour classifier les actions. Cependant, certains joueurs sont droitiers, et d'autres gauchers. Les singularités

pour un même coup entre deux joueurs dont la main dominante est opposée n'auront donc pas les mêmes coefficients. Les singularités ne sont donc pas invariantes aux symétries axiales. Une autre raison expliquant les faibles résultats est le manque de robustesse des singularités aux changements de points de vue. Un mouvement de rotation d'un joueur ou de la raquette peut être visible sur une vue de face, mais pas observable sur une vue latérale.

Pour l'édition suivante, nous avons changé d'approche et utiliser une approche permettant de surmonter les limites précédentes évoquées.

5.4.2 Une méthode utilisant un réseau multi-flux et les Images Dynamiques

Pour la seconde participation en 2020, nous avons proposé une méthode utilisant des Images Dynamiques (DI) (BILEN et al., 2018). Pour plus de détails sur ces travaux, l'article est disponible en annexe C.2. L'idée d'une DI est de résumer l'information d'une vidéo en une seule image. Une DI capture l'évolution temporelle des images observées, en modifiant les intensités de pixels au fil du temps.

Les DI peuvent être calculées sur des images RGB, mais aussi sur des flots optiques. Nous représentons figure 5.5 des exemples de DI calculées sur les images RGB et sur les flots optiques (Dynamic Optical Flow, abrégé en DOF).

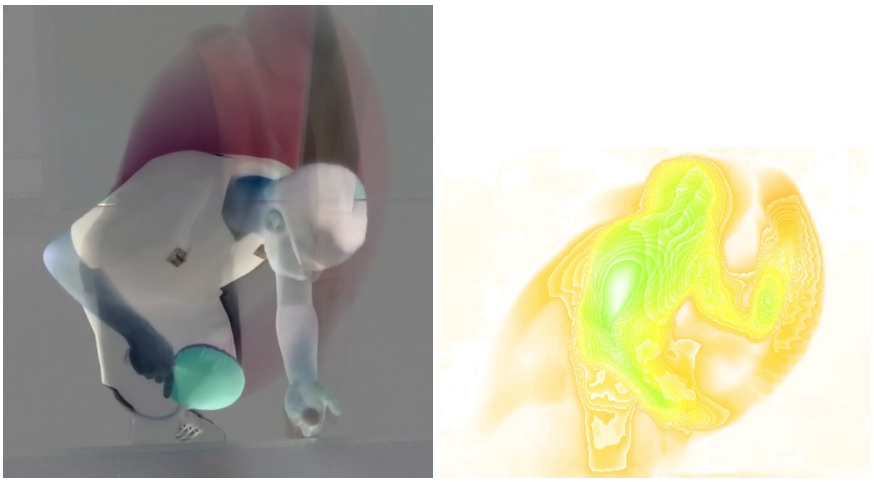


FIGURE 5.5 – Représentation du mouvement sur l'ensemble d'une séquence à l'aide d'image dynamique (DI) à gauche, et de flot optique dynamique (DOF) à droite

En reconnaissance d'action sur des vidéos, les deux principales méthodes d'apprentissage profonde utilisées sont :

— Les Two-Stream Network (SIMONYAN et ZISSERMAN, 2014) consistent à utiliser un CNN pour extraire les informations d'une image RGB, et un CNN qui prend en entrée tous les flots optiques successifs. Les flots optiques permettent de ne garder que l'information de mouvement, sans redondance temporelle. Les sorties des deux réseaux sont ensuite fusionnées.

— Les 3D-CNN (ARUNNEHRU, CHAMUNDEESWARI et BHARATHI, 2018) consistent à prendre en entrée plusieurs images, et utilisent des filtres convolutionnels 3D au lieu de filtres 2D. Les 3D-CNN permettent de garder l'information de l'ensemble de la séquence d'images. Ceux-ci ont donc généralement de bonnes performances mais nécessitent beaucoup de paramètres, de temps de calcul, et un jeu de donnée de grande taille.

Comme les DI sont des images, nous pouvons utiliser de nombreux réseaux pré-entraînés pour la classification d'images. À titre d'exemple, en utilisant un ResNet-50, BILEN et al., 2018 obtient une précision de 86,6% sur UCF-101 (SOOMRO, ZAMIR et SHAH, 2012) en utilisant des images dynamiques. Nous avons donc fait le choix d'utiliser des réseaux ResNet pré-entraînés sur ImageNet (RUSSAKOVSKY et al., 2015). Pour chacune des cinq tentatives envoyées à MediaEval 2020, nous utilisons un ResNet avec une entrée image différente. Les différents types d'images qui sont passés au réseau sont :

— Image RGB au centre temporel de la séquence

— DI calculée sur l'ensemble des images RGB

— DI calculée sur une moitié de séquence (première ou seconde moitié temporelle)

— DI calculée sur l'ensemble des flots optiques, ou Flot Optique Dynamique (DOF)

La figure 5.6 illustre le réseau profond utilisé pour la méthode soumise ayant les meilleurs résultats de classification. C'est celui qui comprend le plus d'entrées, avec deux images dynamiques calculées sur les images RGB, une image dynamique sur du flot optique, et une image RGB. Chaque entrée est ensuite passée à un ResNet, et les informations de sorties des ResNets sont ensuite fusionnées en utilisant une couche totalement connectée.

TABLE 5.2 – Présentation des cinq tentatives, et résultats sur la base d'entraînement, de validation et de test.

Méthode	Entraînement	Validation	Test
DI	25,00%	25,70%	11,58%
DI + RGB	30,34%	23,65%	10,17%
2* DI	62,28%	36,48%	11,58%
2*DI + RGB	63,05%	36,48%	11,51%
2*DI + RGB + DOF	79,21%	44,58%	12,99%

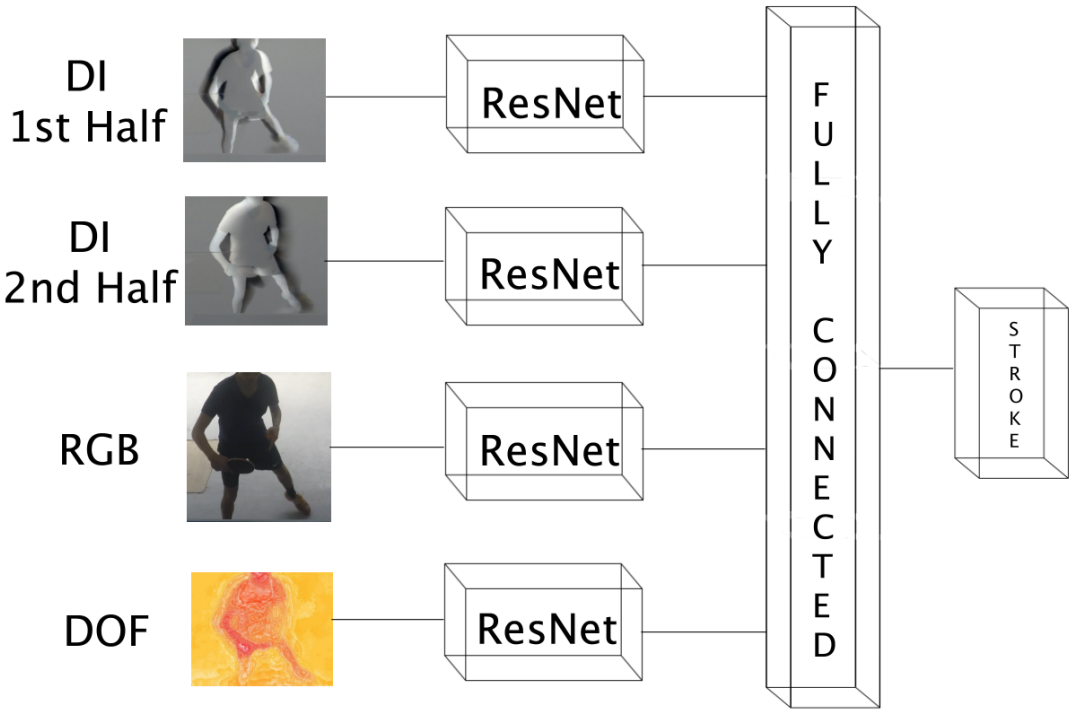


FIGURE 5.6 – Réseau pour la tentative 5, utilisant quatre ResNet pour analyser une image RGB, les DI calculée sur les demies-séquences, et la DOF

Les résultats des cinq tentatives sont présentés dans le tableau 5.2. Plus de détails sur les cinq réseaux sont présentés à l'annexe C.2.

Le meilleur résultat de classification obtenu est de 12, 99%, ce qui est inférieur au résultat que nous avions obtenus précédemment à l'aide des singularités (voir sous-section 5.4.1). Nous observons cependant une augmentation des résultats sur les bases d'entraînement, et les bases de validation. La différence entre le taux de bonnes classifications sur les bases d'entraînement et de test implique que le réseau fait du sur-apprentissage.

L'ensemble d'entraînement est certainement trop petit pour généraliser correctement, en particulier en ce qui concerne le point de vue de la caméra. Il serait donc peut être intéressant de faire une augmentation de données plus poussée, notamment sur les déformations géométriques liées à la position de la caméra.

5.5 Conclusion

Nous sommes organisateurs de la tâche `Sports Video` à MediaEval depuis 2019. Sur nos deux années en tant que participants, les résultats soumis sur les séquences d'évaluations ont été mitigés. Toutefois, l'objectif de MediaEval est avant tout de tester et de comparer différentes approches, notamment pour les étudiants en doctorant. Le bilan est donc positif, puisqu'il nous a à la fois permis comme participant de toucher les limites de certaines approches, et de discuter avec les autres participants des approches les plus prometteuses. Sur nos deux années de participation à MediaEval, un ensemble de 8 équipes ont aussi pris part à la tâche. Le meilleur résultat de classification est de 31,35% (NGUYEN-TRUONG et al., 2020), en utilisant un Channel-Separated Convolutional Networks (CSN) (TRAN et al., 2019). Toutes les autres approches avaient des résultats de classification entre 5% et 26%, ce qui souligne la difficulté de la tâche proposée.

Depuis 2021, la base de données utilisée pour la tâche a été enrichie. La séparation des données en base d'entraînement/test est différente, plus équilibrée parmi les classes, avec des points de vue commun, et un seul type de coup présent sur chaque séquence. Cette modification du jeu de données a permis d'améliorer fortement les résultats des participants. À titre comparatif, l'équipe avec le meilleur score a obtenu un taux de 74,21% (YIJUN et al., 2021), et a utilisée une approche de type Transformers. Nous avons également proposé un challenge sur la détection d'actions, qui est une sous-tâche plus difficile et qui n'a eu jusqu'à présent qu'une seule participation.

Nous continuons d'organiser ces deux sous-tâches en 2022, avec comme nouveauté cette année une mise en avant de l'explicabilité dans les résultats d'algorithmes d'apprentissage, via une nouvelle sous-tâche `Quest for Insight`. Il est aussi prévu dans le futur une extension de la tâche à la natation, dans le cadre d'une collaboration avec le laboratoire LIRIS de Lyon.

Chapitre 6

Conclusion générale et perspectives

Sommaire

6.1 Bilan

Tout le long de ce manuscrit, nous avons présenté un ensemble de méthodes pour l'analyse du geste sportif par la vision dans un contexte non-intrusif, avec comme application le Tennis de Table. La principale motivation était d'extraire à partir de vidéos des caractéristiques de performance (comme par exemple la vitesse de frappe, l'angle de frappe, et les effets donnés à la balle) pouvant aider les entraîneurs et les sportifs.

Dans le chapitre 1, nous avons introduit la problématique de l'analyse fine des gestes sportifs, et l'apport croissant de l'analyse vidéo dans ce contexte. En effet, il est possible en laboratoire d'utiliser des traceurs (cellules photosensibles, marqueurs actifs) ou des « exosquelettes » pour d'effectuer des analyses biomécaniques du sportif. Toutefois, pour étudier les pratiques physiques de pleine nature (dispositifs sportifs – salle de sport, parcours de santé, ...), il semble plus naturel d'utiliser la vidéo, sans marqueurs ou appareillages pouvant gêner le sportif dans sa performance et sa pratique. C'est le cadre dans lequel nous nous sommes placés dans les travaux présentés dans ce manuscrit, avec comme sport d'étude, le tennis de table.

Nous nous sommes intéressé à l'étude trajectographique de la balle lors d'échanges au tennis de table, ainsi qu'à l'extraction de ses paramètres cinématiques (vitesse de translation et de rotation). Ces paramètres sont très importants puisqu'ils déterminent les effets donnés à la balle, et donc l'efficacité du coup réalisé. Nous commençons dans le chapitre 2 par présenter les différentes méthodes de la littérature pour estimer la position 3D d'un objet dans une scène. La première étape dans toutes les approches est la calibration qui permet d'obtenir des informations relatives à la caméra et à sa position dans la scène : les paramètres intrinsèques et extrinsèques. Notre méthode de reconstruction 3D est présentée dans la deuxième section de ce chapitre. Elle utilise une approche mono-vision moins contraignante pour les sportifs. Nous utilisons un réseau convolutif pour estimer le diamètre de la balle sur l'image ce qui permet d'obtenir sa distance à la caméra puis sa position 3D. Grâce au suivi de la balle au long de la séquence vidéo, nous obtenons une séquences de positions 3D. Nous avons créé un jeu de données `avec sportifs` ainsi qu'un jeu de données `synthétiques`. Pour le premier jeu de données, la vérité terrain sur les positions 3D est obtenue par stéréovision. Le réseau convolutif est pré-entraîné sur le jeu de données`synthétique` puis entraîne par transfert d'apprentissage sur le jeu de données `avec sportifs`. Au final, nous obtenons sur le jeu de données `avec sportifs` une très bonne précision sur la distance balle-caméra.

Après l'estimation des positions 3D successives de la balle dans le référentiel monde, nous traitons dans le chapitre 3 l'analyse trajectographique de la balle entre sa frappe et son rebond sur la table. Nous avons proposé une méthode utilisant un modèle physique prenant en compte la gravité, les frottements de l'air ainsi que la rotation (effet Magnus). Cette trajectoire dépend de paramètres cinématiques, que sont les vitesses intiales de translation

V_0 ou de rotation ω_0 de la balle. Ces dernières sont intéressantes pour caractériser et quantifier la performance d'un coup effectué par un joueur, et nous nous sommes focalisés sur leur estimation dans ce chapitre.
La vérité-terrain des vitesses de balle sur les séquences `avec sportifs` n'étant pas connue, nous estimons dans un premier temps ces paramètres cinématiques sur des trajectoires simulées avec Blender. La génération de séquences `synthétiques` à partir du modèle physique introduit a été détaillée. Nous avons présenté ensuite l'évaluation des paramètres cinématiques extraits sur ce jeu de données `synthétiques`, en comparant avec la vérité terrain sur les vitesses de rotation et de translation initiales. La méthode proposée utilise une minimisation avec l'algorithme de Levenberg-Marquardt. Elle permet de réduire le temps de calcul d'un facteur proche de 3 par rapport à une approche de recherche d'optimum par grille, tout en améliorant l'estimation du vecteur vitesse V_0 initial. L'erreur sur ω_0 est cependant plus importante par rapport à la méthode de recherche par grille. Enfin, sur le jeu de données `avec sportifs`, dont la vérité terrain est inconnue, nous avons extrait les paramètres cinématiques pour effectuer une classification sur 3 classes de coups, et démontrer leur pertinence.

La balle rebondit sur la table ou la raquette entre chaque section de la trajectoire. Dans le chapitre 4, nous étudions ces deux cas. Après un impact sur une surface, la vitesse de translation et de rotation de la balle sont conditionnées par le type de matériau constituant cette surface. La première section de ce chapitre correspond à l'analyse du rebond sur la table. En utilisant les sections de trajectoire avant et après le rebond ainsi qu'un modèle physique nous ré-estimons la vitesse de rotation de la balle, ce qui conduit à une diminution de l'erreur obtenue au chapitre précédent de presque 50 %. La deuxième section de ce chapitre est consacrée à l'analyse du rebond entre la balle et la raquette. À partir des paramètres cinématiques extraits avant l'impact sur la raquette et la section de trajectoire après l'impact raquette, nous avons déterminé la vitesse de la raquette et son orientation. Ces informations donnent directement les caractéristiques de la frappe du joueur. Sur notre jeu de données `synthétiques avec frappe`, en utilisant une méthode d'optimisation dédiée, nous avons obtenu une erreur de $0,30\ m/s$ pour la vitesse, et de $3,08^\circ$ pour l'orientation. La validation sur des données réelles, reste à faire, ce qui nécessitera une réécriture partielle de notre méthode d'optimisation.

À l'issue de ces travaux, nous avons développé une chaîne de traitement complète d'analyse fine du geste sportif, allant de l'acquisition de séquences vidéo avec les sportifs à l'extraction de la vitesse et angle de frappe de la raquette en se basant sur l'analyse de la trajectoire des balles.

Dans la deuxième partie de ce manuscrit, nous nous sommes intéressés à la classification d'actions humaines à travers la participation au benchmark MediaEval. Nous sommes aussi membre de l'équipe organisatrice de la tâche `Sports Videos` depuis 2019. Lors de nos deux participations, nous avons proposé des approches basées sur l'analyse et la représentation du mouvement, la première utilisant les points critiques du champ de mouvement, la deuxième approche utilisant les images dynamiques.
Il pourrait être intéressant de pouvoir utiliser dans une future participation à MediaEval les paramètres cinématiques extraits et présentés dans ce manuscrit. Si les conditions d'acquisition sont données avec la tâche, il serait possible de reconstruire la trajectoire de la balle lors de l'échange. Les paramètres cinématiques liés à la balle (vitesse de translation, vitesse de rotation) ou à la raquette (vitesse de frappe, angle de frappe) peuvent apporter des informations pertinentes pour reconnaître l'action du joueur.

6.2 Perspectives

Dans ces travaux, nous avons apporté des contributions à l'analyse du geste sportif par un système mono-caméra. Beaucoup de pistes restent ouvertes : en premier lieu, des améliorations liées à la méthode proposée dans ce manuscrit, et en second lieu des applications à d'autres domaines.

6.2.1 Protocole d'acquisition, et reconstruction de trajectoires

Calibration automatique

L'étape initiale de notre méthode est toujours la reconstruction 3D des positions successives de la balle. Il est donc nécessaire de connaître les paramètres intrinsèques et extrinsèques de la caméra. Cela est fait dans notre cas en détectant de manière automatique des mires de calibration et des points remarquables de la table. Dans le cas où notre méthode serait appliquée dans différentes halles sportives, ou que nous souhaitons effectuer d'autres séquences d'acquisition, une calibration est nécessaire pour chaque nouvelle scène. Des méthodes basées sur des réseaux profonds pourraient être utilisées pour automatiser la détection de la table et extraire les paramètres de

la caméra dans le cadre mono-vision. Des approches convaincantes ont été proposées dans des sports comme le football ou le basket SHA et al., 2020 .

Utilisations de caméra événementielles pour supprimer le flou de déplacement

Lors de l'acquisition de nos séquences `avec sportifs`, nous utilisons actuellement deux caméras synchronisées qui ont une fréquence d'acquisition à 240 fps pour une image de taille 2048×1088 pixels. Sur nos caméras, nous pouvons baisser la fréquence d'acquisition afin d'obtenir une meilleure résolution spatiale de la balle, mais au prix d'une augmentation du flou de déplacement. Inversement, augmenter la fréquence d'acquisition réduirait le flou de déplacement, mais diminuerait la résolution spatiale de nos images (et donc rendrait complexe l'estimation précise du diamètre apparent de la balle). L'utilisation d'une caméra *événementielle* serait une approche alternative prometteuse qui permettrait une très grande vitesse d'acquisition tout en conservant une résolution élevée. Dans ce type de caméra, ce sont les variations d'intensité lumineuse qui sont détectées. Chaque pixel étant indépendant, la fréquence d'acquisition maximale obtenue dépasse celle des caméras traditionnelles, pouvant atteindre des centaines de milliers d'images par seconde et donc s'affranchir du flou de mouvement. La résolution spatiale de ces caméras a augmenté progressivement ces dernières années, et l'exploitation de ce type de caméra pourrait être intéressante pour la reconstruction de la trajectoire de la balle en 3D. La sortie d'une caméra événementielle étant une matrice très creuse, elles sont adaptées aux Spiking Neural Networks (SNNs) (GHOSH-DASTIDAR et ADELI, 2009), à basse consommation énergétique, ce qui pourrait également être intéressant pour déployer des modules tout-en-un dans des halles de sport. Des membres du laboratoire MIA sont en contact avec l'équipe Fox du laboratoire CRISTAL de l'Université de Lille, qui est spécialisée dans ce domaine de recherche.

Estimation globale de la trajectoire et des paramètres cinématiques

Notre approche étudie la trajectoire d'une balle en analysant ses positions successives au cours du temps. Nous détectons la balle, puis nous utilisons un algorithme de suivi pour obtenir ses positions successives, et enfin nous déterminons son diamètre apparent sur chaque image. Bien que naturelle, cette approche image par image est sensible au bruit, car elle suppose une

estimation précise du diamètre pour estimer de façon fiable la distance à la caméra. Si cette distance est imprécise, tous les paramètres extraits sont, eux aussi, imprécis.
Notre idée serait donc d'utiliser comme entrée d'un réseau profond soit un cube vidéo ou soit une Image Dynamique, représentant une trajectoire pour obtenir en sortie les paramètres cinématiques sur cette trajectoire. En effet, chaque section de trajectoire est entièrement déterminée par les vecteurs vitesses initiaux et les coefficients associés à la table et à la balle : il y a donc unicité de la trajectoire pour un jeu de paramètre donné. Il s'agirait donc, pour obtenir une plus grande robustesse, d'estimer ces paramètres à partir des sections de trajectoires, et non d'une seule image.

6.2.2 Extension du modèle physique

Prise en compte des effets non planaires

Jusqu'à présent, notre jeu de données s'est composé de trois types de coups pour lesquels il n'y a presque pas d'effets latéraux, et nous avons donc pu faire l'hypothèse que la trajectoire de la balle s'effectuait dans un plan. Cependant, pour d'autres types de coups comme les services, les effets latéraux sont très importants, et nous dévions de l'hypothèse planaire. Il conviendrait donc pour pouvoir traiter ces catégories de coups de remplacer la projection planaire (Chapitre 2) par une projection plus générale sur une surface supposée régulière.

Analyse plus précise des paramètres de la raquette

Dans notre méthode, nous extrayons les paramètres de la raquette (vitesse de translation, et angle au moment de l'impact) depuis un modèle de rebond utilisant un coefficient de restitution (COR) fixe. En réalité, comme nous l'avons déjà évoqué, ce coefficient dépend de nombreux facteurs liés aux conditions de jeu et aux choix tactiques du joueur. Pour déterminer le COR de la raquette d'un joueur, plusieurs approches peuvent être envisagées. La première, qui est dans la continuité de nos travaux actuels, consiste à intégrer le COR dans les paramètres à optimiser dans notre fonction d'erreur. Pour que la méthode d'optimisation donne des résultats fiables, il faudra utiliser davantage de données et donc un ensemble de trajectoires complètes.

Une approche complémentaire serait d'utiliser une méthode de détection de la raquette, par exemple avec un réseau convolutif. Une source d'inspiration pourrait être les approches utilisées pour la détection du regard humain qui détectent l'ellipse de l'iris, mais également déterminent la direction du regard (KOTHARI et al., 2021). S'il semble illusoire de penser qu'une telle approche permette à elle seule de déterminer précisément l'orientation de la raquette, elle pourrait être une aide à l'initialisation de la méthode d'optimisation.

Une troisième approche serait d'utiliser des capteurs de type `inertial measurement unit` (IMU) qui pourraient être intégrés dans des raquettes. Si cette approche s'éloigne de notre parti-pris d'être non intrusif, cela permettrait de valider nos résultats par des approches complémentaires. De plus, le COR et les autres paramètres de la raquette pourraient être déterminés une seule fois, durant une séance d'entraînement dédiée. Actuellement, dans le cadre de EU Conexus, des membres du MIA sont en contact avec des membres de l'Institut Waterford en Irlande dont le domaine de recherche est précisément ce type de capteurs.

6.2.3 Extension du domaine d'application

Extension à d'autres sports de balle

Nos travaux pourraient être étendus à d'autres sports de balle. Pour cela, il faudrait utiliser des paramètres physiques différents prenant en compte les différents matériaux constituant la balle ou les surfaces de rebond. Pour citer quelques exemples, au tennis, la balle possède des micro-poils, qui augmentent la résistance avec l'air ; dans le cas du football, ou du baseball, la balle possède des coutures pouvant provoquer des effets différents de ceux observés au Tennis de Table. Bien que nos travaux se soient concentrés sur un seul sport, il devrait être possible d'étendre l'analyse trajectographique de balle à d'autres disciplines, afin d'obtenir des indicateurs de performance pertinents.

Projets futurs impliquant l'analyse du mouvement humain

Les résultats de cette thèse, ainsi que les travaux qui vont suivre, s'inscrivent dans un nouveau cadre : le projet SMART, Sport Mouvement Ambition Recherche Technologie, financé par l'Idex de Bordeaux. Il s'agit d'un projet de construction d'une salle de sport instrumentée et connectée sur le campus bordelais. Ce bâtiment fait partie des espaces de préparation pour les championnats du monde de rugby 2023, et doit être une base de préparation d'équipes pour les Jeux Olympiques de 2024 notamment pour les athlètes paralympiques.

La pratique et l'étude de la performance sportive sont donc des objectifs principaux de ce projet. Le bâtiment SMART sera équipé d'un gymnase augmenté (Réalité Virtuelle), ainsi que d'un maillage de caméras reliées à des salles informatiques surplombant les salles de sport. Il hébergera des projets de recherche scientifique et de développement technologique sur l'étude du mouvement humain. Nous serons pleinement impliqués dans la mise en place des méthodes d'analyse des sportifs par la Vision, et cela sera l'occasion de tester et d'étendre les travaux présentés dans ce manuscrit.

Enfin, on peut envisager à plus long terme, qu'une application potentielle serait le cadre de l'Usine du Futur. En effet, la réduction des troubles musculo-squelettiques (TMS) et l'amélioration de la performance au poste de travail sont des thématiques majeurs de l'usine de demain. Lors d'un travail à la chaîne, le stress et les troubles musculo-squelettiques provoquent de nombreuses maladies professionnelles chez l'opérateur. Pouvoir analyser la posture de l'opérateur et étudier ses déplacements permettra de réduire ces risques. Nous pensons que certains aspects méthodes développés dans cette thèse pourraient être adaptés à ce nouveau cadre. De plus, la vision est une source d'information non-invasive, et qui évite de gêner le travailleur dans sa tâche.

Annexe A

Protocole d'acquisition

Matériel

Matériel nécessaire pour la calibration et les acquisitions dans le gymnase

1. Mire de calibration (Checkerboard)

   Mire de calibration de la taille la plus grande possible pour faire la calibration avec une distance similaire à celle de notre application. Les carrés doivent être suffisamment gros sur les images (30+pixels) pour être précis, mais suffisamment petits pour avoir le plus de carrés possib visibles.

   Taille(mm) = Distance_camera_scène(mm) * Taille_image(pixels) / Distance_focale.

   Exemple : Avec une distance focale de 4k5 et une distance de 5m, les carrés doivent faire 38mm

   Tick all that apply.

   - [ ] A4 à l'unité
   - [ ] A3 à l'unité
   - [ ] Avec trièdre

2. Acquisition

   Tick all that apply.

   - [ ] Ordinateur avec le logiciel Coreview (Linux/Windows) (+alimentation)
   - [ ] Boîtier d'acquisition (+cable d'alimentation + connection)
   - [ ] Caméras *2 (+cables)
   - [ ] Trépieds *2
   - [ ] Multiprise
   - [ ] Ralonge (rouleau)
   - [ ] Vignettes/disques colorés (pour simplifier l'annotation du sol sous les coins de table)
   - [ ] Pointeur lumineux (pour positionner les vignettes)
   - [ ] Balles blanches
   - [ ] Balles avec rotation visible
   - [ ] Scotch
   - [ ] Stylo/crayon
   - [ ] Feuille

Mise en place

Avant l'arrivée des sportifs, préparation de la scène

1. Visibilité

Tick all that apply.

- [ ] Bonne lumière naturelle
- [ ] Limiter l'usage des néons (si nécessaire, essayer de limiter les néons possitionés bas pour limiter les réverbérations)

2. Mesures au sol

Tick all that apply.

- [ ] Choix de la position de la table par rapport aux lignes au sol, et la lumière (même luminosité dans les deux caméras)
- [ ] Mesurer les lignes visibles : schéma

3. Positionner la table

Tick all that apply.

- [ ] Placer la table au repère souhaité
- [ ] Placer au sol (Z=0) des vignettes sous les coins de la table (pour l'annotation)

4. Configuration des cameras

Tick all that apply.

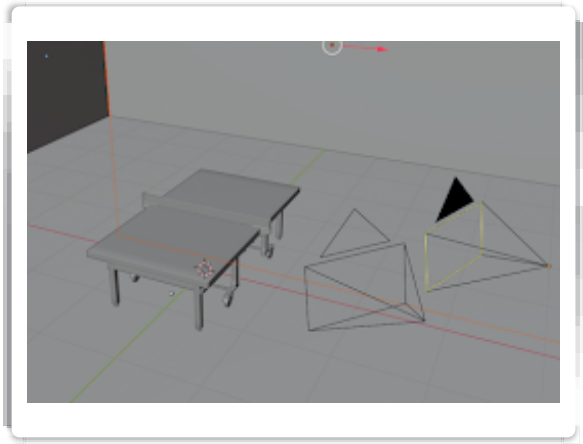
- [ ] Brancher le boîtier, les caméras, et l'ordinateur
- [ ] Résolution : 1920x1080 (suppression des bords) : Menu Flare => Window => 1920x1080 => Center => Apply
- [ ] Couleurs : Flare => Exposure => Image => Lookup table (IOI Slog)
- [ ] Balance des blanc : Flare => Balance => Auto white balance
- [ ] Synchronisation (menu) : Véifier que le trigger est activé dans le Control Signal Manager
- [ ] Synchronisation (activation) : Flare => Exposure => Edge-triggered => Trigger Input 1

Scènes

Liste des scènes à enregistrer (au moins une, voir plus selon le temps disponible).
Pour chaque scène, une calibration doit être effectuée, et la liste des acquisitions (feuille 4) doit être suivie

1. Caméras du même côté

Mark only one oval.



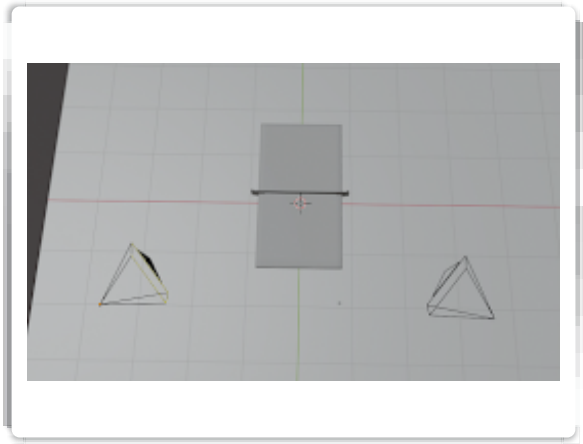
Vue Globale

Cam 1

Cam 2

2. Caméras des deux côtés

Tick all that apply.



- [ ] Vue Globale

- [ ] Cam 1

- [ ] Cam 2

3. Calibration

Tick all that apply.

- [ ] Position des mires A4 sur différentes positions sur la table (4 coins + devant le filet)
- [ ] Enlever les mires A4
- [ ] Position des mires A3 sur différentes positions sur la table (4 coins + devant filet)
- [ ] Enlever les mires A3
- [ ] Utilisation du trièdre, et déplacer celui-ci pour faire une calibration a hauteur d'utilisation

Acquisitions

Liste des coups à effectuer pour chaque scène

1. Nom des participants

Tick all that apply.

- [ ] Joueur 1 :
- [ ] Joueur 2 :
- [ ] Joueur 3 :
- [ ] Joueur 4 :

2. Gamme de coups

Tick all that apply.

- [ ] Top Spins joueur 3 (15 droits, 15 revers)
- [ ] Contre-Attaques joueur 3 (15 droits, 15 revers)
- [ ] Poussettes joueur 3 (15 droits, 15 revers)
- [ ] Blocs joueur 3 (15 droits, 15 revers)
- [ ] Défense joueur 3 (15 droits, 15 revers)
- [ ] Top Spins joueur 4 (15 droits, 15 revers)
- [ ] Contre-Attaque joueur 4 (15 droits, 15 revers)
- [ ] Poussettes joueur 4 (15 droits, 15 revers)
- [ ] Blocs joueur 4 (15 droits, 15 revers)
- [ ] Défenses joueur 4 (15 droits, 15 revers)

3. Gamme de services

Tick all that apply.

- [ ] Service lifté joueur 1 (15 droits, 15 revers)
- [ ] Service coupé joueur 1 (15 droits, 15 revers)
- [ ] Service marteau joueur 1 (15 droits, 15 revers)
- [ ] Service lifté joueur 2 (15 droits, 15 revers)
- [ ] Service coupé joueur 2 (15 droits, 15 revers)
- [ ] Service marteau joueur 2 (15 droits, 15 revers)

4. Extrait de match

Tick all that apply.

- [ ] Focus Joueur 1 (1-2 min)
- [ ] Focus Joueur 2 (1-2 min)
- [ ] Focus Joueur 3 (1-2 min)
- [ ] Focus Joueur 4 (1-2 min)

Annexe B

MediaEval Workshop

Sports Video : Fine Grained Action Detection and Classification of Table Tennis Strokes from videos

B.1 Task Description

This task offers researchers an opportunity to test their fine-grained classification methods for detecting and recognizing strokes in table tennis videos. The low inter-class variability makes the task more difficult than with usual general datasets like UCF-101. The task offers two subtasks :

Subtask 1 : Stroke Detection Participants are required to build a system that detects whether a stroke has been performed, whatever its class, and to extract its temporal boundaries. The aim is to be able to distinguish between moments of interest in a game (players performing strokes) from irrelevant moments (picking up the ball, having a break…). This subtask can be a preliminary step for later recognizing a stroke that has been performed.

Subtask 2 : Stroke Classification Participants are required to build a classification system that automatically labels video segments according to a performed stroke. There are 21 possible stroke classes.

Compared with Sports Video 2020, this year we extend the task in the direction of detection and also enrich the data set with new and more diverse stroke samples.

Participants are encouraged to make their code public with their submission.

B.2 Motivation and background

Action detection and classification are one of the main challenges in visual content analysis and mining. Sport video analysis has been a very popular

research topic, due to the variety of application areas, ranging from analysis of athletes' performances and rehabilitation to multimedia intelligent devices with user-tailored digests. Datasets focused on sports activities or datasets including a large amount of sport activity classes are now available and many research contributions benchmark on those datasets. A large amount of work is also devoted to fine-grained classification through the analysis of sport gestures using motion capture systems. However, body-worn sensors and markers could disturb the natural behavior of sports players. Furthermore, motion capture devices are not always available for potential users, be it a University Faculty or a local sports team. Giving end-users the possibility to monitor their physical activities in ecological conditions through simple equipment is a challenging issue. The ultimate goal of this research is to produce automatic annotation tools for sports faculties, local clubs and associations to help coaches better assess and advise athletes during training.

B.3 Target group

The task is of interest to researchers in the areas of machine learning, visual content analysis, computer vision and sport performance. We explicitly encourage researchers focusing specifically in domains of computer-aided analysis of sport performance.

B.4 Data

Our focus is on recordings that have been made by widespread and cheap video cameras, e.g., GoPro. We use a dataset specifically recorded at a sport faculty facility and continuously completed by students and teachers. This dataset is constituted of player-centered videos recorded in natural conditions without markers or sensors. It comprises 20 table tennis strokes, and a rejection class. The problem is hence a typical research topic in the field of video indexing : for a given recording, we need to label the video by recognizing each stroke appearing in it.

B.5 Evaluation methodology

Twenty stroke classes are considered according to the rules of table tennis. This taxonomy was designed with professional table tennis teachers. We

are working on videos recorded at the Faculty of Sports of the University of Bordeaux. Students are the sportsmen filmed and the teachers are supervising exercises conducted during the recording sessions. The dataset has been recorded in a sport faculty facility using a light-weight equipment, such as GoPro cameras. The recordings are markerless and allow the players to perform in natural conditions from different viewpoints. These sequences were manually annotated, and the annotation sessions were supervised by professional players and teachers using a crowdsourced annotation platform.

The training dataset shared for each subtask is composed of videos of table tennis matches with temporal borders of performed strokes supplied in an xml file, with the corresponding stroke label.

Subtask 1 : Stroke Detection Participants are asked to temporally segment regions where a stroke is performed on unknown videos of matches. The IoU metric on temporal segments will be used for evaluation.

Subtask 2 : Stroke Classification Participants produce an xml file where each stroke of test sequences is labeled according to the given taxonomy. Submissions will be evaluated in terms of accuracy per class and global accuracy.

For each subtask, participants may submit up to five runs. We also encourage participants to carry out a failure analysis of their results in order to gain insight into the mistakes that their classifiers make.

B.6 Task organizers

You can email us directly at mediaeval.sport.task (at) diff.u-bordeaux.fr
— Jordan Calandre, MIA, University of La Rochelle, France
— Pierre-Etienne Martin, Max Planck Institute for Evolutionary Anthropology, Germany
— Jenny Benois-Pineau, Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, France
— Renaud Péteri, MIA, University of La Rochelle, France
— Boris Mansencal, CNRS, Bordeaux INP, LaBRI, France
— Julien Morlier, IMS, University of Bordeaux, France
— Laurent Mascarilla, MIA, University of La Rochelle, France

B.7 Task Schedule

— 1 August 2021 : Data release
— 25 October 2021 : Runs due

— 8 November 2021 : Results returned
— 22 November 2021 : Working notes paper
— 6-8 December 2021 : MediaEval 2020 Workshop

B.8 Acknolwedgments

We would like to thank all the players and annotators for their involvement in the acquisition and annotation processes and Alain Coupet from sport faculty of Bordeaux, expert and teacher in table tennis, for the proposed table tennis strokes taxonomy.

Annexe C

Participations à MediaEval

C.1 MediaEval 2019 - Singularités du flot optique

Optical Flow Singularities for Sports Video Annotation: Detection of Strokes in Table Tennis

Jordan Calandre[1], Renaud Péteri [2], Laurent Mascarilla[3]
[1]MIA Laboratory, La Rochelle University, France
{jordan.calandre1,renaud.peteri,lmascari}@univ-lr.fr

ABSTRACT

Over the past few years, Action Recognition task has drawn considerable interests, leading to intensive researches. This is mainly due to the variety of related applications, from autonomous car to human behavior analysis.

Up to now, most of researches aim to identify various sport actions such as UCF-101 dataset[11], but, due to the exponential number of online videos and the necessity to be more and more accurate, the need of finer analysis arises.

In this working note, results for the MediaEval 2019 Sports Video Annotation "Detection of Strokes in Table Tennis" task [9] are presented. As in sport videos displacement flow appears to be one of the most useful information for stroke identification, especially to differentiate quite similar strokes, this proposal relies on a combination of spatial information and Optical Flow's singularities identification. As a result, most relevant regions of video frames for the classification task are detected.

1 INTRODUCTION

The selected task requires to analyze a single sport, which means that the analysis has to be even more precise than high inter-class variance datasets. The dataset, aiming at representing real-life sportsman training situations, is made up of videos recorded using standard cameras with unbalanced number of training samples for each stroke. No depth maps or data issued from motion capture suits are available.

This working note provides a description of the methods proposed by the team MIA on this task. Only handcrafted features extracted from video frames and optical flow are used: Histogram of oriented Gradients (HoG)[6] features and dense Optical Flow singularities's coefficients projected on Legendre basis. These features are represented by a Bag-of-Words model and the final classification is obtained by mean of a linear SVM.

2 OUR APPROACH

The great success and popularity of Deep Learning methods for 2D images recognition tasks, led many researchers to adapt these architectures to video analysis using 3D filters instead of 2D filters commonly known as 3DCNN[13].

For both manual and deep learning methods, the Optical Flow was also proved relevant, with the arrival of two-stream network architectures[10] or Siamese Network[8]. Because the automatically calculated filters of deep-learning methods could have no real human meaning compared to handcrafted approaches, we decided to extract interesting regions around the player based only on the optical flow's singularities [1–3] and did complementary analysis on this areas.

Figure 1: Extracted Optical Flow using PWC-Net

As already said, the proposed approach relies on dense accurate Optical Flow. Nowadays, one of the most popular method is probably the Farneback [7] method which starts by generating an image pyramid of different resolutions, and uses polynomial expansion to match the pixel from one resolution to another. The main issue with this method is that when an object of uniform color is moving, only the borders of that object are detected. Using Farneback provides good edges, but empty objects.

More recent methods are trying to overcome this drawback, especially, the PWC-Network [12] that use CNN pyramidal feature extraction, warping layers, and cost volume layers to match features of the first image and warped features of the second one. Our method uses such a network pre-trained using the Sintel dataset [4], an open source animated short film, to give clean boundaries like in Figure 1. Compared to the Sintel dataset, the task dataset presents a lot of compression artifacts, consequently, Gaussian blur is applied before Optical Flow extraction, and frames are resized to speed up consequent processing.

2.1 Optical Flow Singularities

Given the horizontal and vertical components U and V of the optical flow, regions of high rotation or divergence are detected by the following stage. For each frame, using a sliding window, the optical flow is locally approximated using a Legendre polynomial basis. The polynomial basis P is defined as:

$P_{K,L}(x_1, x_2) = \sum_{k=0}^{K} \sum_{l=0}^{L} x_1^k x_2^l$

To obtain precise results, a small sliding window of 50 pixels is chosen. The resulting computational cost is therefore limited as a one-dimensional polynomial basis is precise enough in such a case

MediaEval'19, 27-29 October 2019, Sophia Antipolis, France

Table 1: Global accuracy

Method	Train set	Test set
Unbalanced SVM	153/754	25/354
Position + Unbalanced SVM	426/754	46/354
Position + HoG + Unbalanced SVM	524/754	46/354
Position + Hog + Balanced SVM	485/754	50/354

:

$$U = u_{0,0}P_{0,0} + u_{0,1}P_{0,1} + u_{1,0}P_{1,0}$$
$$V = v_{0,0}P_{0,0} + v_{0,1}P_{0,1} + v_{1,0}P_{1,0}$$

After the projection, the two components are efficiently calculated on a canonical basis by approximating U and V flows as follows :

$$\begin{pmatrix} U \\ V \end{pmatrix} \simeq A \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + b = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + b_1 \\ a_{21}x_1 + a_{22}x_2 + b_2 \end{pmatrix}$$

Each pixel region is then represented by a $2x2$ matrix made of canonical projection coefficients of the flow. Significant region are selected by a simple threshold:
$\Delta(A) = tr(A)^2 - 4 * det(A), \Delta(A) < 0.05$

2.2 BoW and SVM for Action Recognition

The classification task follow the Bag of Word (BoW) approach: K-Means are used to classify the various singularities (each singularity being originally represented by the four projection coefficients) into six clusters.

Except for the first run, the relative spatial positions of the singularities in the frames are also used. The frames are divided in four-squared grids and the number of singularities on each of these four regions are analysed.

For the last two runs, HoG Features, as represented by a height bins BoW, are also used but only on regions where significant singularities have been selected. This aims at quantifying the relative importance of optical flow-based and gradient based features.

As a result, each stroke is represented by an histogram with at most 18 bins (6 singularities, 8 HoG, and 4 spatial regions).

Classification is done by a cross-validated linear SVM[5], thus avoiding overfitting.

The given dataset being seriously unbalanced, a balanced SVM is used on the last run, giving penalties for the most common classes, to increate the retention rate of rare strokes.

3 RESULTS AND ANALYSIS

The proposed method leads to four runs, using only singularities for the first one, and adding additional information like HoG or the position of the singularities region for the others. The accuracy of the four runs are presented in Table 1 for both training and testing set.

The last three runs with the singularities and spatial/pixel information have pretty similar results for the test set, but the run using only the projection coefficients gives a lower global accuracy. That proves that using movement-based analyze, without using other data is not sufficient to have a good enough interpretation of a stroke, and focusing only on the flow information results in high information loss.

Figure 2: Accuracy of the predicted classes

The second and third run, with singularity positions and unbalanced SVM have similar results both in terms of overall accuracy and predicted classes. This behavior is unexpected as one of the run uses Hog features, while the others does not. Maybe, because only one sport is present in the dataset, the players edges are not sufficient to differentiate strokes. We used HoG on each frame, knowing that one frame alone isn't enough to know what stroke class it belongs to. We stacked them over the whole sequence without taking into account the temporal data, and that's probably why the HoG have no impact on the results overall.

On the other hand, the only run with balanced SVM provides a better overall accuracy. As said in the introduction, the dataset is heterogeneously balanced. Standard unbalanced SVM predicts the classes to increase the overall result. On this dataset, it overpredicts the most frequent classes. By using weights, balanced SVM increases its accuracy on the rare classes, resulting in a worst overall result, but in better results on rare classes.

4 DISCUSSION AND OUTLOOK

This paper presents an approach for the Sports Video Annotation on single-sport dataset task. Due to the difficulty of the task, the rare classes samples, missing metadata about right or left handed players, and different camera viewpoints, didn't achieved high performance scores, but it gives an insight of what is missing in the proposed Optical Flow's Singularities features.

There is a still rooms for improvement, mostly due to the lack of long term temporal information and the variations between two optical flows of the same stroke class when recorded by cameras on different viewpoints.

REFERENCES

[1] Cyrille Beaudry, Renaud Péteri, and Laurent Mascarilla. 2014. Action recognition in videos using frequency analysis of critical point trajectories. *2014 IEEE International Conference on Image Processing, ICIP 2014*. https://doi.org/10.1109/ICIP.2014.7025289

[2] Cyrille Beaudry, Renaud Péteri, and Laurent Mascarilla. 2016. An efficient and sparse approach for large scale human action recognition in videos. *Machine Vision and Applications* 27, 4 (2016), 529–543.

[3] Katy Blanc, Diane Lingrand, and Frédéric Precioso. 2017. SINGLETS: Multi-Resolution Motion Singularities for Soccer Video Abstraction. In *Workshop CVsports (in conjunction with CVPR) (Proceedings of the Workshop CVsports (in conjunction with CVPR))*. Honolulu (Hawaii), United States. https://hal.archives-ouvertes.fr/hal-01540342

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. 2012. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV) (Part IV, LNCS 7577)*, A. Fitzgibbon et al. (Eds.) (Ed.). Springer-Verlag, 611–625.

[5] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* 2 (2011), 27:1–27:27. Issue 3. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[6] N. Dalal and B. Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 886–893 vol. 1. https://doi.org/10.1109/CVPR.2005.177

[7] Gunnar Farnebäck. 2003. Two-frame Motion Estimation Based on Polynomial Expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA'03)*. Springer-Verlag, Berlin, Heidelberg, 363–370. http://dl.acm.org/citation.cfm?id=1763974.1764031

[8] P. Martin, J. Benois-Pineau, R. Péteri, and J. Morlier. 2018. Sport Action Recognition with Siamese Spatio-Temporal CNNs: Application to Table Tennis. In *2018 International Conference on Content-Based Multimedia Indexing (CBMI 2018)*. 1–6. https://doi.org/10.1109/CBMI.2018.8516488

[9] Pierre-Etienne Martin, Jenny Benois-Pineau, Boris Mansencal, Renaud Péteri, Laurent Mascarilla, Jordan Calandre, and Julien Morlier. 2019. Sports Video Annotation: Detection of Strokes in Table Tennis task for MediaEval 2019. *Proc. of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-29 October 2019.*

[10] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. *CoRR* abs/1406.2199 (2014). arXiv:1406.2199 http://arxiv.org/abs/1406.2199

[11] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* abs/1212.0402 (2012). arXiv:1212.0402 http://arxiv.org/abs/1212.0402

[12] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. 2017. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. *CoRR* abs/1709.02371 (2017). arXiv:1709.02371 http://arxiv.org/abs/1709.02371

[13] Shuiwang Ji ; Wei Xu ; Ming Yang ; Kai Yu. 2013. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (Jan 2013), 221–231. https://doi.org/10.1109/TPAMI.2012.59

C.2 MediaEval 2020 - Images Dynamiques

Four-stream network and Dynamic Images for Sports Video Classification: Classification of Strokes in Table Tennis

Jordan Calandre[1], Renaud Péteri [2], Laurent Mascarilla[3]
[1]MIA Laboratory, La Rochelle University, France
{jordan.calandre1,renaud.peteri,lmascari}@univ-lr.fr

ABSTRACT

In this working note, results for the MediaEval 2020 Sports Video Annotation "Detection of Strokes in Table Tennis" task are presented. Fine-grained action classification remains a complex task due to the low variance between two strokes, especially in natural conditions. Our proposal is therefore based on motion, which is the most obvious representation of what players are doing. Motion information is captured at the image level by optical flow streams and summarized at the sequence level by Dynamic Images that encode temporal information. A multiple stream architecture is presented, combining RGB-based Dynamic Images, Dynamic Images based on optical flow, and RGB frames to classify table tennis strokes.

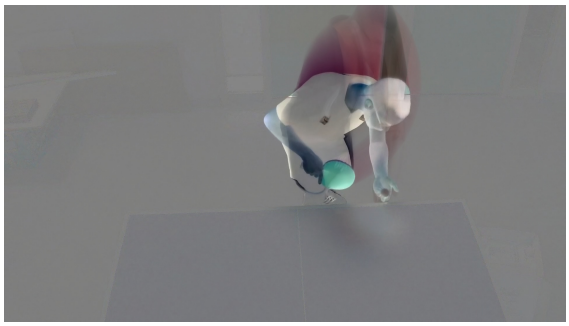


Figure 1: Dynamic Image

1 INTRODUCTION

Fine-grained action recognition in natural conditions remains difficult even after the success of CNN architectures for image and video processing. Datasets like UCF-101 [11], or HMDB [7] are useful for benchmarking methods classifying human action into a given set of sport classes, however the fine-grained recognition of gestures of a specific sport leads to new challenges.

The dataset `TTStroke-21` [10] is made up for this purpose and is much more challenging than most previous datasets. Acquisition is done using standard cameras, without depth maps or motion capture information. The number of strokes are also heavily unbalanced, which can lead to overfitting when training deep neural networks.

Deep learning methods for 2D images recognition tasks led to the spread of CNN network for video analysis. Popular methods, like 3D-CNN, using 3D filters instead of 2D filters on video frames, require huge datasets to be trained efficiently. An alternative method is to use the optical flow. These approaches like two-stream networks or Siamese Networks have been very successful. The optical flow represents the movement between two consecutive frames, but without estimating long term dependencies. The movement being the obvious representation of a stroke, we focus on this feature to enhance our previous proposal [3]. Optical flow and Dynamic Images [1] are used to capture image motion information.

2 OUR APPROACH

We have participated to MediaEval 2019 [9] with a method using optical flow singularities [3], and have noticed that temporal data were not fully exploited with this approach. Our new proposal for MediaEval 2020 [8] is to use Dynamic Images [1] to summarize each sequence based on RGB, along with optical flow obtained by the so-called PWC-Network [12]. A multiple-stream architecture with late fusion is then used to process the different network inputs.

2.1 Dynamic Images (DI)

A Dynamic Image [1] (DI) is a representation of an image sequence in a single frame. This frame is obtained by representing the video using a ranking function on its frames [5]. A pixel pooling operation is applied with the ranking function to average the pixel values over time.

2.2 Dynamic Optical Flow (DOF)

The optical flow being a two dimensional vector field that represents the apparent motion between two consecutive frames, it does not capture long-term motion. When combining the flow of each frame of a video sequence using the same approach as for DI, the motion of an entire stroke into a single image is aggregated, and thus long-term interactions can be captured.

To obtain a dense flow with clean boundaries, the PWC-Network [12] has been selected as it achieves suitable results at decent speed. It has been trained using the Sintel dataset [2].

Since the videos at hand contain compression artifacts, a Gaussian filter is applied before estimating the optical flow.

2.3 CNN Architecture

The proposed CNN architecture is composed of up to four branches. Each branch corresponds to a ResNet[6] with 152 layers, pretrained on ImageNet [4] but the input type varies according to the branch. The five possible inputs for the branches are: A Dynamic Image (DI) computed on the whole sequence; the RGB frame from the middle of the sequence; two Dynamic Images computed on each half of the sequence (DIHalf); a Dynamic Image computed on the optical flow (DOF). The input type, of the branches, for each run is presented in Table. 1. Every input is a `224x224` pixel image, cropped around the

MediaEval'20, December 14-15 2020, Online

Figure 2: Dynamic Optical Flow (DOF)

Table 1: Run results

Method	Train set	Val Set	Test set
DI	25.00%	25.70%	11.58%
DI + RGB	30.34%	23.65%	10.17%
2*DIHalf	62.28%	36.48%	11.58%
2*DIHalf + RGB	63.05%	36.48%	11.51%
2*DIHalf + DOF + RGB	79.21%	44.58%	12.99%

player using `Detectron2` [13]. We modified the last fully connected layer to have 20 neurons, which is the number of considered classes. To combine the branches outputs, a late fusion is applied followed by a fully connected layer that results in the final stroke classification score.

The network was trained over 100 epochs, with a learning rate of 0.05 and a momentum of 0.9 using 10-folds cross validation. All the video sequences of the dataset with at least two different strokes are used in the validation set.

3 RESULTS AND ANALYSIS

The accuracy, for each of the five allowed runs of the task, is presented in Table. 1 for training validation and and testing sets.

To our surprise, the scores are quite similar for runs using one DI or two DIHalf. By averaging the features only on half the sequence, the use of two DIHalf (runs 3,4 and 5) was expected to better represent the movement. This seems to have no real impact on the overall result, nor the adding of the RGB frame located at the middle of the sequence. The only run with a better score is the one with DOF (Dynamic Optical Flow). The DOF encodes the movement but unlike the dynamic RGB images, it provides an insight of the direction of the players hands/grip when in action.

In last year task challenge, using optical flow singularities [3], our best score was 50/354 by adding a weight on the predicted strokes to compensate the unbalanced dataset. We obtained 46/354 correctly classified moves for the two best runs without class-weighted SVM.

Compared to last year, our network has a better estimate of the drive's type (Forehand vs Backhand) presented in Table. 2. We also considerably increased player's stroke estimate (Serve/Offensive/Defensive). This metric increased from 48.87% to 65.25%. The confusion matrix for the drive and stroke estimation, for run 5, is presented in Fig.3.

Table 2: Comparison of the accuracy between our MediaEval 2019 and MediaEval 2020 submissions

Metric	2019	2020
Drive(Forehand/Backhand)	61.58%	65.25%
Group(Serve/Offensive/Defensive)	48.87%	65.82%
Group and Drive	29.10%	49.15%
Total accuracy	14.12%	12.99%

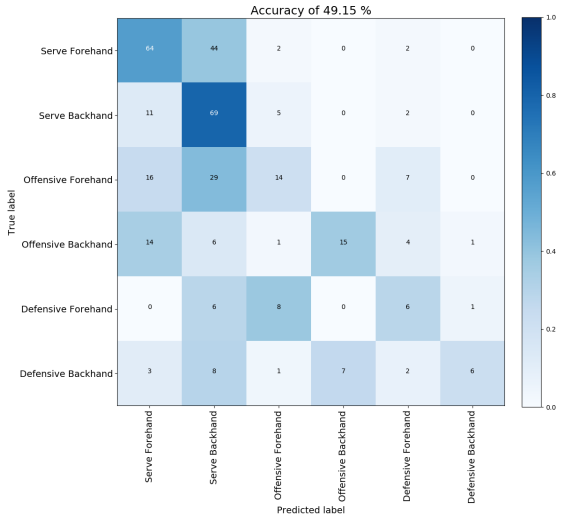


Figure 3: Accuracy of the predicted drive and stroke estimates

4 DISCUSSION AND OUTLOOK

This paper presents the approach of the MIA laboratory for the Sports Video Annotation on single-sport dataset task. Due to the difficulty of the task, such as rare classes samples and different camera viewpoints, the overfit obtained during the training sessions leads to a low score, but it gives an insight of what kind of information is missing in the proposed Dynamic Images. RGB frames and Dynamic Images are arbitrarily split in the middle of each sequence, but an impact detection of the ball could be used to make a more meaningful splitting. Lastly, unbalanced data must be better handled as prediction is clearly biased toward some stroke classes.

5 ACKNOWLEDGMENTS

The research is supported by the Region of Nouvelle Aquitaine through the CRISP project and by the CNRS MIRES federation.

REFERENCES

[1] Hakan Bilen, Basura Fernando, Efstratios Gavves, Andrea Vedaldi, and Stephen Gould. 2016. Dynamic Image Networks for Action Recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2016-December. 3034–3042.

https://doi.org/10.1109/CVPR.2016.331

[2] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. 2012. A naturalistic open source movie for optical flow evaluation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-33783-3_44

[3] Jordan Calandre, Renaud Péteri, and Laurent Mascarilla. 2019. Optical flow singularities for sports video annotation: Detection of strokes in table tennis. In *CEUR Workshop Proceedings*, Vol. 2670.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[5] Basura Fernando and Stephen Gould. 2017. Discriminatively Learned Hierarchical Rank Pooling Networks. *International Journal of Computer Vision* (2017). https://doi.org/10.1007/s11263-017-1030-x arXiv:1705.10420

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.90 arXiv:1512.03385

[7] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre Thomas. 2013. Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering 12: Transactions of the High Performance Computing Center, Stuttgart (HLRS) 2012*. IEEE Computer Society, 571–582. https://doi.org/10.1007/978-3-642-33374-3

[8] Pierre-Etienne Martin, Jenny Benois-Pineau, Boris Mansencal, Renaud Péteri, Laurent Mascarilla, Jordan Calandre, and Julien Morlier. 2020. Sports Video Classification: Classification of Strokes in Table Tennis for MediaEval 2020. In *Proc. of the MediaEval 2020 Workshop, Online, 14-15 December 2020*.

[9] Pierre Etienne Martin, Jenny Benois-Pineau, Boris Mansencal, Renaud Péteri, Laurent Mascarilla, Jordan Calandre, and Julien Morlier. 2019. Sports video annotation: Detection of strokes in table tennis task for mediaeval 2019. In *CEUR Workshop Proceedings*.

[10] Pierre Etienne Martin, Jenny Benois-Pineau, Renaud Péteri, and Julien Morlier. 2020. Fine grained sport action recognition with Twin spatio-temporal convolutional neural networks: Application to table tennis. *Multimedia Tools and Applications* (2020). https://doi.org/10.1007/s11042-020-08917-3

[11] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* abs/1212.0402 (2012). arXiv:1212.0402 http://arxiv.org/abs/1212.0402

[12] Deqing Sun, Xiaodong Yang, Ming Yu Liu, and Jan Kautz. 2018. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* abs/1709.0 (2018), 8934–8943. https://doi.org/10.1109/CVPR.2018.00931 arXiv:1709.02371

[13] Uxin Wu, Alexander Kirillov, Francisco Massa, Wan-YenLo, and Ross Girshick. 2019. Detectron2. *https://github.com/facebookresearch/detectron2* (2019).

Bibliographie

ABU-EL-HAIJA, Sami, Nisarg KOTHARI, Joonseok LEE, Paul NATSEV, George TODERICI, Balakrishnan VARADARAJAN et Sudheendra VIJAYANARASIMHAN (2016). « YouTube-8M : A Large-Scale Video Classification Benchmark ». In : *arXiv :1609.08675*. arXiv : 1609.08675. URL : http://arxiv.org/abs/1609.08675.

ARUNNEHRU, J., G. CHAMUNDEESWARI et S. Prasanna BHARATHI (2018). « Human Action Recognition using 3D Convolutional Neural Networks with 3D Motion Cuboids in Surveillance Videos ». In : *Procedia Computer Science* 133, p. 471-477. ISSN : 18770509. DOI : 10.1016/j.procs.2018.07.059.

BAO, Han, Xiaopeng CHEN, Zhan Tao WANG, Min PAN et Fei MENG (2012). « Bouncing model for the table tennis trajectory prediction and the strategy of hitting the ball ». In : *2012 IEEE International Conference on Mechatronics and Automation*. IEEE, p. 2002-2006.

BARTELSEN, Jan, Helmut MAYER, Heiko HIRSCHMÜLLER, Andreas KUHN et Mario MICHELINI (2012). « Orientation and dense reconstruction of unordered terrestrial and aerial wide baseline image sets ». In : *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. T. 1, p. 25-30. DOI : 10.5194/isprsannals-I-3-25-2012.

BEWLEY, Alex, Zongyuan GE, Lionel OTT, Fabio RAMOS et Ben UPCROFT (2016). « Simple online and realtime tracking ». In : *Proceedings - International Conference on Image Processing, ICIP*. T. 2016-August. IEEE, p. 3464-3468. ISBN : 9781467399616. DOI : 10.1109/ICIP.2016.7533003. arXiv : 1602.00763.

BIAN, Tianling, Yang HUA, Tao SONG, Zhengui XUE, Ruhui MA, Neil ROBERTSON et Haibing GUAN (2020). « VTT : Long-term visual tracking with transformers ». In : *Proceedings - International Conference on Pattern Recognition*. IEEE, p. 9585-9592. ISBN : 9781728188089. DOI : 10.1109/ICPR48806.2021.9412156.

BILEN, Hakan, Basura FERNANDO, Efstratios GAVVES et Andrea VEDALDI (2018). « Action Recognition with Dynamic Image Networks ». In : *IEEE*

Transactions on Pattern Analysis and Machine Intelligence 40.12, p. 2799-2813. ISSN : 19393539. DOI : 10.1109/TPAMI.2017.2769085. arXiv : 1612.00738.

BLACKMAN, Samuel S. (2004). « Multiple hypothesis tracking for multiple target tracking ». In : *IEEE Aerospace and Electronic Systems Magazine* 19.1 II, p. 5-18. ISSN : 08858985. DOI : 10.1109/MAES.2004.1263228.

BLANC, Katy, Diane LINGRAND et Frederic PRECIOSO (juill. 2017). « Singlets : Multi-resolution Motion Singularities for Soccer Video Abstraction ». In : *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. T. 2017-July. Proceedings of the Workshop CVsports (in conjunction with CVPR). Honolulu (Hawaii), United States, p. 66-75. ISBN : 9781538607336. DOI : 10.1109/CVPRW.2017.15. URL : https://hal.archives-ouvertes.fr/hal-01540342.

BLANK, Peter, Benjamin H. GROH et Bjoern M. ESKOFIER (2017). « Ball speed and spin estimation in table tennis using a racket-mounted inertial sensor ». In : *Proceedings - International Symposium on Wearable Computers, ISWC*. T. Part F130534, p. 2-9. ISBN : 9781450351881. DOI : 10.1145/3123021.3123040.

BLENDER ONLINE COMMUNITY (2013). *Blender - a 3D modelling and rendering package*. Stichting Blender Foundation, Amsterdam. URL : http://www.blender.org/.

BLESER, Gabriele, Bertram TAETZ, Markus MIEZAL, Corinna A. CHRISTMANN, Daniel STEFFEN et Katja REGENSPURGER (2017). « Development of an Inertial Motion Capture System for Clinical Application ». In : *I-Com* 16.2, p. 113-129. ISSN : 1618-162X. DOI : 10.1515/icom-2017-0010. URL : https://doi.org/10.1515/icom-2017-0010.

BRIGGS, Lyman J. (1959). « Effect of Spin and Speed on the Lateral Deflection (Curve) of a Baseball ; and the Magnus Effect for Smooth Spheres ». In : *American Journal of Physics* 27.8, p. 589-596. ISSN : 0002-9505. DOI : 10.1119/1.1934921.

CALANDRE, Jordan, Renaud PÉTERI et Laurent MASCARILLA (2019). « Optical flow singularities for sports video annotation : Detection of strokes in table tennis ». In : *CEUR Workshop Proceedings*. T. 2670.

CALANDRE, Jordan, Renaud PÉTERI, Laurent MASCARILLA et Benoit TREMBLAIS (jan. 2020a). « Extraction and analysis of 3D kinematic parameters of Table Tennis ball from a single camera ». In : *Proceedings - International Conference on Pattern Recognition*. Milano, Italy, p. 9468-9475. ISBN : 9781728188089. DOI : 10.1109/ICPR48806.2021.9412391. URL : https://hal.archives-ouvertes.fr/hal-02975085.

— (2020b). « Extraction et analyse de trajectoires de balle de Tennis de Table à partir d'une seule caméra pour l'aide à la performance sportive ». In : *Reconnaissance des Formes, Image, Apprentissage et Perception*, p. 27-29.

— (2021). « Table Tennis ball kinematic parameters estimation from non-intrusive single-view videos ». In : *Proceedings - International Workshop on Content-Based Multimedia Indexing*. T. 2021-June. IEEE, p. 1-6. ISBN : 9781665442206. DOI : 10.1109/CBMI50038.2021.9461884.

CHATTERJEE, Avishek (2016). « Geometric Calibration and Shape Refinement for 3D Reconstruction [eindwerk] ». Thèse de doct. URL : https://www.researchgate.net/publication/307600956.

CHEN, Hua Tsung, Chien Li CHOU, Wen Jiin TSAI et Suh Yin LEE (2011). « 3D ball trajectory reconstruction from single-camera sports video for free viewpoint virtual replay ». In : *2011 IEEE Visual Communications and Image Processing, VCIP 2011*. IEEE, p. 1-4. ISBN : 9781457713200. DOI : 10.1109/VCIP.2011.6115930.

CHEN, Ying et Anthony VETRO (2014). « Next-generation 3D formats with depth map support ». In : *IEEE Multimedia* 21.2, p. 90-94. ISSN : 1070986X. DOI : 10.1109/MMUL.2014.31.

CUNHA, Augusto, Axelle POCHET, Hélio LOPES et Marcelo GATTASS (2020). « Seismic fault detection in real data using transfer learning from a convolutional neural network pre-trained with synthetic seismic data ». In : *Computers and Geosciences* 135, p. 104344. ISSN : 00983004. DOI : 10.1016/j.cageo.2019.104344.

DANELLJAN, Martin, Gustav HÄGER, Fahad Shahbaz KHAN et Michael FELSBERG (2014). « Accurate scale estimation for robust visual tracking ». In : *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*. Bmva Press. DOI : 10.5244/c.28.65.

DORMAND, John R et Peter J PRINCE (1980). « A family of embedded Runge-Kutta formulae ». In : *Journal of computational and applied mathematics* 6.1, p. 19-26.

FALANGA, Davide, Suseong KIM et Davide SCARAMUZZA (2019). « How Fast Is Too Fast? the Role of Perception Latency in High-Speed Sense and Avoid ». In : *IEEE Robotics and Automation Letters* 4.2, p. 1884-1891. ISSN : 23773766. DOI : 10.1109/LRA.2019.2898117.

FANG, Yang, Geun Sik JO et Chang Hee LEE (2020). « RSinet : Rotation-scale invariant network for online visual tracking ». In : *Proceedings - International Conference on Pattern Recognition*. IEEE, p. 4153-4160. ISBN : 9781728188089. DOI : 10.1109/ICPR48806.2021.9412862. arXiv : 2011.09153.

FEICHTENHOFER, Christoph, Axel PINZ et Andrew ZISSERMAN (2017). *Detect to Track and Track to Detect*. DOI : 10.1109/ICCV.2017.330. arXiv : 1710.03958.

FERNANDO, Tharindu, Simon DENMAN, Sridha SRIDHARAN et Clinton FOOKES (2018). « Tracking by Prediction : A Deep Generative Model for Mutli-person Localisation and Tracking ». In : *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*. T. 2018-January. IEEE, p. 1122-1132. ISBN : 9781538648865. DOI : 10.1109/WACV.2018.00128. arXiv : 1803.03347.

FERRARI, Vittorio, Frederic JURIE et Cordelia SCHMID (2010). « From images to shape models for object detection ». In : *International Journal of Computer Vision* 87.3, p. 284-303. ISSN : 09205691. DOI : 10.1007/s11263-009-0270-9.

FORSYTH, David (2014). « Object detection with discriminatively trained part-based models ». In : *Computer* 47.2, p. 6-7. ISSN : 00189162. DOI : 10.1109/MC.2014.42.

FORTMANN, Thomas E., Yaakov BAR-SHALOM et Molly SCHEFFE (1983). « Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association ». In : *IEEE Journal of Oceanic Engineering* 8.3, p. 173-184. ISSN : 15581691. DOI : 10.1109/JOE.1983.1145560.

FTDLYC (2019). *libcalib*. URL : https://github.com/ftdlyc/libcalib.

— (2020). *libcbdetect*. URL : https://github.com/ftdlyc/libcbdetect.

GALLEGO, Guillermo, Jon E.A. LUND, Elias MUEGGLER, Henri REBECQ, Tobi DELBRUCK et Davide SCARAMUZZA (2018). « Event-Based, 6-DOF Camera Tracking from Photometric Depth Maps ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.10, p. 2402-2412. ISSN : 19393539. DOI : 10.1109/TPAMI.2017.2769655. arXiv : 1607.03468.

GARWIN, Richard L (1969). « Kinematics of an ultraelastic rough ball ». In : *American Journal of Physics* 37.1, p. 88-92.

GEIGER, Andreas, Frank MOOSMANN, Oemer CAR et Bernhard SCHUSTER (2012). « A toolbox for automatic calibration of range and camera sensors using a single shot ». In : *International conference on robotics and automation (ICRA)*.

GHOSH-DASTIDAR, Samanwoy et Hojjat ADELI (2009). « Spiking neural networks ». In : *International journal of neural systems* 19.04, p. 295-308.

GU, Chunhui, Chen SUN, David A. ROSS, Carl VONDRICK, Caroline PANTOFARU, Yeqing LI, Sudheendra VIJAYANARASIMHAN, George TODERICI, Susanna RICCO, Rahul SUKTHANKAR, Cordelia SCHMID et Jitendra MALIK (2018).

« AVA : A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions ». In : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* abs/1705.0, p. 6047-6056. ISSN : 10636919. DOI : 10.1109/CVPR.2018.00633. arXiv : 1705.08421.

HA, Hyowon, Michal PERDOCH, Hatem ALISMAIL, In So KWEON et Yaser SHEIKH (oct. 2017). « Deltille Grids for Geometric Camera Calibration ». In : *Proceedings of the IEEE International Conference on Computer Vision*. T. 2017-October, p. 5354-5362. ISBN : 9781538610329. DOI : 10.1109/ICCV.2017.571.

HALLERT., Bertil (1960). *Camera Calibration*. DOI : 10.1111/j.1477-9730.1960.tb01300.x. URL : https://docs.opencv.org/4.5.3/dc/dbb/tutorial%7B%5C_%7Dpy%7B%5C_%7Dcalibration.html.

HARTLEY, Richard et Andrew ZISSERMAN (2003). *Multiple view geometry in computer vision*. Cambridge university press.

HAVALDAR, Parag (2006). « Course notes : Performance driven facial animation ». In : *SIGGRAPH 2006 - ACM SIGGRAPH 2006 Courses*, p. 5. DOI : 10.1145/1185657.1185845.

HENRIQUES, Joao F., Rui CASEIRO, Pedro MARTINS et Jorge BATISTA (2015). « High-speed tracking with kernelized correlation filters ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.3, p. 583-596. ISSN : 01628828. DOI : 10.1109/TPAMI.2014.2345390. arXiv : 1404.7584. URL : http://arxiv.org/abs/1404.7584.

HUANG, Yanlong, De XU, M. TAN et Hu SU (sept. 2011). « Trajectory prediction of spinning ball for ping-pong player robot ». In : p. 3434-3439. DOI : 10.1109/IROS.2011.6095044.

HURL, Braden, Krzysztof CZARNECKI et Steven WASLANDER (2019). « Precise synthetic image and LiDAR (PreSIL) dataset for autonomous vehicle perception ». In : *IEEE Intelligent Vehicles Symposium, Proceedings*. T. 2019-June. IEEE, p. 2522-2529. ISBN : 9781728105604. DOI : 10.1109/IVS.2019.8813809. arXiv : 1905.00160.

IINO, Yoichi et Takeji KOJIMA (2009). « Kinematics of table tennis topspin forehands : effects of performance level and ball spin ». In : *Journal of Sports Sciences* 27.12, p. 1311-1321.

KOTERA, Jan, Denys ROZUMNYI, Filip SROUBEK et Jiri MATAS (2019). « Intra-frame object tracking by deblatting ». In : *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, p. 2300-2309. ISBN : 9781728150239. DOI : 10.1109/ICCVW.2019.00283. arXiv : 1905.03633.

KOTHARI, Rakshit S, Aayush K CHAUDHARY, Reynold J BAILEY, Jeff B PELZ et Gabriel J DIAZ (2021). « Ellseg : An ellipse segmentation framework for robust gaze tracking ». In : *IEEE Transactions on Visualization and Computer Graphics* 27.5, p. 2757-2767.

KU, Jason, Melissa MOZIFIAN, Jungwook LEE, Ali HARAKEH et Steven L. WASLANDER (2018). « Joint 3D Proposal Generation and Object Detection from View Aggregation ». In : *IEEE International Conference on Intelligent Robots and Systems*. IEEE, p. 5750-5757. ISBN : 9781538680940. DOI : 10.1109/IROS.2018.8594049. arXiv : 1712.02294.

KUEHNE, Hilde, Hueihan JHUANG, Rainer STIEFELHAGEN et Thomas SERRE THOMAS (2013). « Hmdb51 : A large video database for human motion recognition ». In : *High Performance Computing in Science and Engineering : Transactions of the High Performance Computing Center, Stuttgart (HLRS) 2012*. {IEEE} Computer Society, p. 571-582. ISBN : 9783642333743. DOI : 10.1007/978-3-642-33374-3.

KUSUBORI, Seiji, Kazuto YOSHIDA et Hiroshi SEKIYA (2012). « the Functions of Spin on Shot Trajectory in Table Tennis ». In : *International Symposium on Biomechanics in Sports : Conference Proceedings Archive 2012*. T. 30. 42, p. 245-248.

LADJAILIA, Ammar, Imed BOUCHRIKA, Hayet Farida MEROUANI, Nouzha HARRATI et Zohra MAHFOUF (2020). « Human activity recognition via optical flow : decomposing activities into basic actions ». In : *Neural Computing and Applications* 32.21, p. 16387-16400.

LAPTEV, Ivan, Marcin MARSZALEK, Cordelia SCHMID et Benjamin ROZENFELD (2008). « Learning realistic human actions from movies ». In : *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, p. 1-8.

LEVINE, Evan, Manuel MARTINELLO et Mahdi NEZAMABADI (2016). « High-precision multi-view camera calibration using a rotating stage ». In : *Proceedings - International Conference on Image Processing, ICIP*. T. 2016-August, p. 1175-1179. ISBN : 9781467399616. DOI : 10.1109/ICIP.2016.7532543.

LI, Jianquan, Yingjie YIN, Xilong LIU, De XU et Qingyi GU (2017). « 12,000-fps Multi-object detection using HOG descriptor and SVM classifier ». In : *IEEE International Conference on Intelligent Robots and Systems*. T. 2017-September. IEEE, p. 5928-5933. ISBN : 9781538626825. DOI : 10.1109/IROS.2017.8206487.

LI, Jing, Makoto TSUBOKURA et Masaya TSUNODA (2017). « Numerical Investigation of the Flow Past a Rotating Golf Ball and Its Comparison with

a Rotating Smooth Sphere ». In : *Flow, Turbulence and Combustion* 99.3-4, p. 837-864. ISSN : 15731987. DOI : 10.1007/s10494-017-9859-1.

LI, Larry et al. (2014). « Time-of-flight camera–an introduction ». In : *Technical white paper* SLOA190B.

LI, Qingpeng, Lichao MOU, Qingjie LIU, Yunhong WANG et Xiao Xiang ZHU (2018). « HSF-Net : Multiscale deep feature embedding for ship detection in optical remote sensing imagery ». In : *IEEE Transactions on Geoscience and Remote Sensing* 56.12, p. 7147-7161. ISSN : 01962892. DOI : 10.1109/TGRS.2018.2848901.

LI, Weixin, Qian YU, Harpreet SAWHNEY et Nuno VASCONCELOS (2013). « Recognizing activities via bag of words for attribute dynamics ». In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 2587-2594.

LIN, Hsien I., Zhangguo YU et Yi Chen HUANG (2020). « Ball tracking and trajectory prediction for table-tennis robots ». In : *Sensors (Switzerland)* 20.2, p. 333. ISSN : 14248220. DOI : 10.3390/s20020333.

LIU, Jia Qiang, Bin WANG, Xuan ZHAO et Yan DOU (mars 2014). « The Application of Rubber Materials on Table Tennis Racket ». In : *Mechanical Engineering, Intelligent System and Applied Mechanics*. T. 473. Applied Mechanics and Materials. Trans Tech Publications Ltd, p. 116-120. DOI : 10.4028/www.scientific.net/AMM.473.116.

LIU, Wei, Dragomir ANGUELOV, Dumitru ERHAN, Christian SZEGEDY, Scott REED, Cheng Yang FU et Alexander C. BERG (2016). « SSD : Single shot multibox detector ». In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9905 LNCS, p. 21-37. ISSN : 16113349. DOI : 10.1007/978-3-319-46448-0_2. arXiv : 1512.02325. URL : http://dx.doi.org/10.1007/978-3-319-46448-0%7B%5C_%7D2.

LUKEZIC, Alan, Tomás VOJÍŘ, Luka CEHOVIN ZAJC, Jiří MATAS et Matej KRISTAN (2018). « Discriminative Correlation Filter Tracker with Channel and Spatial Reliability ». In : *International Journal of Computer Vision*. T. 126. 7, p. 671-688. DOI : 10.1007/s11263-017-1061-3.

MA, Wenchi, Yuanwei WU, Feng CEN et Guanghui WANG (2020). « MDFN : Multi-scale deep feature learning network for object detection ». In : *Pattern Recognition* 100, p. 107149. ISSN : 00313203. DOI : 10.1016/j.patcog.2019.107149. arXiv : 1912.04514.

MARTIN, Pierre Etienne, Jenny BENOIS-PINEAU, Boris MANSENCAL, Renaud PÉTERI et Julien MORLIER (2020). « Classification of strokes in table tennis

with a three stream spatio-temporal CNN for MediaEval 2020 ». In : *CEUR Workshop Proceedings*. T. 2882.

MARTIN, Pierre-Etienne (déc. 2020). « Fine-grained action detection and classification from videos with spatio-temporal convolutional neural networks : Application to Table Tennis. » Theses. Université de Bordeaux. URL : https://tel.archives-ouvertes.fr/tel-03128769.

MARTIN, Pierre-Etienne, Jenny BENOIS-PINEAU, Boris MANSENCAL, Renaud PÉTERI, Laurent MASCARILLA, Jordan CALANDRE et Julien MORLIER (2019). « Sports Video Annotation : Detection of Strokes in Table Tennis task for MediaEval 2019 ». In : *Proc. of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-29 October 2019.*

MEHTA, R. D. et J. M. PALLIS (2001). « The aerodynamics of a tennis ball ». In : *Sports Engineering* 4.4, p. 177-189. ISSN : 1369-7072. DOI : 10.1046/j.1460-2687.2001.00083.x.

MI, Tzu-Wei et Mau-Tsuen YANG (2019). « Comparison of tracking techniques on 360-degree videos ». In : *Applied Sciences* 9.16, p. 3336.

MIYAZAKI, Takeshi, Wataru SAKAI, Tatsuro KOMATSU, Naoya TAKAHASHI et Ryutaro HIMENO (déc. 2017). « Lift crisis of a spinning table tennis ball ». In : *European Journal of Physics* 38.2, p. 24001. ISSN : 13616404. DOI : 10.1088/1361-6404/aa51ea. URL : https://doi.org/10.1088%7B%5C%%7D2F1361-6404%7B%5C%%7D2Faa51ea.

NATHAN, Alan M., Joe HOPKINS, Lance CHONG et Hank KACZMARSKI (2006). « The effect of spin on the flight of a baseball ». In : *The Engineering of Sport 6* 1.2, p. 23-28. DOI : 10.1007/978-0-387-46050-5_5.

NEWVILLE, Matthew et Till STENSITZKI (2018). « Non-Linear Least-Squares Minimization and Curve-Fitting for Python ». In : *Non-Linear Least-Squares Minimization and Curve-Fitting for Python*, p. 65. URL : http://cars9.uchicago.edu/software/python/lmfit/lmfit.pdf.

NGUYEN, Trong Nguyen, Huu Hung HUYNH et Jean MEUNIER (2018). « 3D Reconstruction with Time-of-Flight Depth Camera and Multiple Mirrors ». In : *IEEE Access* 6, p. 38106-38114. ISSN : 21693536. DOI : 10.1109/ACCESS.2018.2854262.

NGUYEN-TRUONG, Hai, San CAO, NA Khoa NGUYEN, Bang-Dang PHAM, Hieu DAO, Minh-Quan LE, Hoang-Phuc NGUYEN-DINH, Hai-Dang NGUYEN et Minh-Triet TRAN (2020). « HCMUS at MediaEval 2020 : Ensembles of Temporal Deep Neural Networks for Table Tennis Strokes Classification Task. » In : *MediaEval*.

NIEBLES, Juan Carlos, Chih Wei CHEN et Li FEI-FEI (2010). « Modeling temporal structure of decomposable motion segments for activity classification ». In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. T. 6312 LNCS. PART 2, p. 392-405. ISBN : 3642155510. DOI : 10.1007/978-3-642-15552-9_29.

NONOMURA, Junko, Akira NAKASHIMA et Yoshikazu HAYAKAWA (2010). « Analysis of effects of Rebounds and aerodynamics for trajectory of table tennis ball ». In : *Proceedings of the SICE Annual Conference*. IEEE, p. 1567-1572. ISBN : 9784907764364.

PASZKE, Adam, Sam GROSS, Francisco MASSA, Adam LERER, James BRADBURY, Gregory CHANAN, Trevor KILLEEN, Zeming LIN, Natalia GIMELSHEIN, Luca ANTIGA, Alban DESMAISON, Andreas KÖPF, Edward YANG, Zach DEVITO, Martin RAISON, Alykhan TEJANI, Sasank CHILAMKURTHY, Benoit STEINER, Lu FANG, Junjie BAI et Soumith CHINTALA (2019). « PyTorch : An imperative style, high-performance deep learning library ». In : *Advances in Neural Information Processing Systems*. T. 32. arXiv : 1912.01703.

PEROT, Etienne, Pierre de TOURNEMIRE, Davide NITTI, Jonathan MASCI et Amos SIRONI (2020). « Learning to detect objects with a 1 megapixel event camera ». In : *Advances in Neural Information Processing Systems* 2020-December. ISSN : 10495258. arXiv : 2009.13436.

PLACHT, Simon, Peter FÜRSATTEL, Etienne Assoumou MENGUE, Hannes HOFMANN, Christian SCHALLER, Michael BALDA et Elli ANGELOPOULOU (2014). « ROCHADE : Robust checkerboard advanced detection for camera calibration ». In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. T. 8692 LNCS. PART 4, p. 766-779. ISBN : 9783319105925. DOI : 10.1007/978-3-319-10593-2_50.

RADIUK, Pavlo M (2017). « Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets ». In : *Information Technology and Management Science* 20.1, p. 20-24.

REDMON, Joseph, Santosh DIVVALA, Ross GIRSHICK et Ali FARHADI (2016). « You only look once : Unified, real-time object detection ». In : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. T. 2016-December, p. 779-788. ISBN : 9781467388504. DOI : 10.1109/CVPR.2016.91. arXiv : 1506.02640.

RINALDI, Renaud G., Lionel MANIN, Sébastien MOINEAU et Nicolas HAVARD (2019). « Table Tennis Ball Impacting Racket Polymeric Coatings : Experiments and Modeling of Key Performance Metrics ». In : *Applied Sciences* 9.1. ISSN : 2076-3417. URL : https://www.mdpi.com/2076-3417/9/1/158.

ROTH, Peter M (2008). « On-line Conservative Learning ». Thèse de doct.

ROZUMNYI, Denys (2017). « Tracking , Learning and Detection over a Large Range of Speeds ». Thèse de doct.

ROZUMNYI, Denys, Jan KOTERA, Filip ŠROUBEK, Lukáš NOVOTNÝ et Jiří MATAS (2017). « The world of fast moving objects ». In : *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. T. 2017-January, p. 4838-4846. ISBN : 9781538604571. DOI : 10.1109/CVPR.2017.514. arXiv : 1611.07889.

RUSSAKOVSKY, Olga, Jia DENG, Hao SU, Jonathan KRAUSE, Sanjeev SATHEESH, Sean MA, Zhiheng HUANG, Andrej KARPATHY, Aditya KHOSLA, Michael BERNSTEIN, Alexander C. BERG et Li FEI-FEI (2015). « ImageNet Large Scale Visual Recognition Challenge ». In : *International Journal of Computer Vision* 115.3, p. 211-252. ISSN : 15731405. DOI : 10.1007/s11263-015-0816-y. arXiv : 1409.0575.

SCHNEIDER, Ralf, Lars LEWERENTZ, Karl LÜSKOW, Marc MARSCHALL et Stefan KEMNITZ (déc. 2018). « Statistical analysis of table-tennis ball trajectories ». In : *Applied Sciences (Switzerland)* 8.12. ISSN : 20763417. DOI : 10.3390/app8122595.

SCHONBEIN, Miriam, Tobias STRAUS et Andreas GEIGER (2014). « Calibrating and centering quasi-central catadioptric cameras ». In : *Proceedings - IEEE International Conference on Robotics and Automation*, p. 4443-4450. ISBN : 9781479936854. DOI : 10.1109/ICRA.2014.6907507.

SECCO, Emanuele Lindo et Andualem Maereg TADESSE (2020). « A Wearable Exoskeleton for Hand Kinesthetic Feedback in Virtual Reality ». In : *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*. Sous la dir. de Gregory M P O'HARE, Michael J O'GRADY, John O'DONOGHUE et Patrick HENN. T. 320 LNICST. Cham : Springer International Publishing, p. 186-200. ISBN : 9783030492885. DOI : 10.1007/978-3-030-49289-2_15.

SEEMANTHINI, K. et S. S. MANJUNATH (2018). « Human Detection and Tracking using HOG for Action Recognition ». In : *Procedia Computer Science* 132, p. 1317-1326. ISSN : 18770509. DOI : 10.1016/j.procs.2018.05.048.

SEITZ, Steven M., Brian CURLESS, James DIEBEL, Daniel SCHARSTEIN et Richard SZELISKI (juin 2006). « A comparison and evaluation of multi-view

stereo reconstruction algorithms ». In : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. T. 1, p. 519-526. ISBN : 0769525970. DOI : 10.1109/CVPR.2006.19.

SHA, Long, Jennifer HOBBS, Panna FELSEN, Xinyu WEI, Patrick LUCEY et Sujoy GANGULY (juin 2020). « End-to-End Camera Calibration for Broadcast Videos ». In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

SHAPIRO, Robert (1978). « Direct linear transformation method for three-dimensional cinematography ». In : *Research Quarterly of the American Alliance for Health, Physical Education and Recreation* 49.2, p. 197-205. ISSN : 10671315. DOI : 10.1080/10671315.1978.10615524.

SHEN, Lejun, Qing LIU, Lin LI et Haipeng YUE (2016). « 3D reconstruction of ball trajectory from a single camera in the ball game ». In : *Advances in Intelligent Systems and Computing*. T. 392, p. 33-39. ISBN : 9783319245584. DOI : 10.1007/978-3-319-24560-7_5.

SIMONYAN, Karen et Andrew ZISSERMAN (2014). « Two-stream convolutional networks for action recognition in videos ». In : *Advances in Neural Information Processing Systems*. T. 1. January, p. 568-576. arXiv : 1406.2199.

SMITS, A J et D R SMITH (2021). « A new aerodynamic model of a golf ball in flight ». In : *Science and Golf II*. Taylor & Francis, p. 433-442. DOI : 10.4324/9780203474709-58.

SOOMRO, Khurram, Amir Roshan ZAMIR et Mubarak SHAH (2012). « UCF101 : A dataset of 101 human actions classes from videos in the wild ». In : *arXiv preprint arXiv :1212.0402*.

SRAVYAPRANATI, Bh, D. SUMA, Ch MANJULATHA et Sudhakar PUTHETI (2021). « Large-Scale Video Classification with Convolutional Neural Networks ». In : *Smart Innovation, Systems and Technologies*. T. 196, p. 689-695. DOI : 10.1007/978-981-15-7062-9_69.

STEIN, Manuel, Halldor JANETZKO, Andreas LAMPRECHT, Thorsten BREITKREUTZ, Philipp ZIMMERMANN, Bastian GOLDLÜCKE, Tobias SCHRECK, Gennady ANDRIENKO, Michael GROSSNIKLAUS et Daniel A. KEIM (2018). « Bring It to the Pitch : Combining Video and Movement Data to Enhance Team Sport Analysis ». In : *IEEE Transactions on Visualization and Computer Graphics* 24.1, p. 13-22. ISSN : 10772626. DOI : 10.1109/TVCG.2017.2745181.

SUN, Deqing, Xiaodong YANG, Ming Yu LIU et Jan KAUTZ (2018). « PWC-Net : CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume ». In : *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* abs/1709.0, p. 8934-8943. ISSN : 10636919.

DOI : 10.1109/CVPR.2018.00931. arXiv : 1709.02371. URL : http://arxiv.org/abs/1709.02371.

SZENBERG, Flávio, Paulo Cezar Pinto CARVALHO et Marcelo GATTASS (2001). « Automatic camera calibration for image sequences of a football match ». In : *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. T. 2013. Springer, p. 301-310. ISBN : 3540417672. DOI : 10.1007/3-540-44732-6_31.

TANG, Kevin, Li FEI-FEI et Daphne KOLLER (2012). « Learning latent temporal structure for complex event detection ». In : *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, p. 1250-1257.

TEBBE, Jonas, Lukas KLAMT, Yapeng GAO et Andreas ZELL (mai 2020). « Spin Detection in Robotic Table Tennis ». In : *Proceedings - IEEE International Conference on Robotics and Automation*, p. 9694-9700. ISSN : 10504729. DOI : 10.1109/ICRA40945.2020.9196536. arXiv : 1905.07967. URL : http://dx.doi.org/10.1109/ICRA40945.2020.9196536.

TOSHEV, Alexander, Ben TASKAR et Kostas DANIILIDIS (2012). « Shape-based object detection via boundary structure segmentation ». In : *International Journal of Computer Vision* 99.2, p. 123-146. ISSN : 09205691. DOI : 10.1007/s11263-012-0521-z.

TRAN, Du, Heng WANG, Lorenzo TORRESANI et Matt FEISZLI (2019). « Video Classification with Channel-Separated Convolutional Networks ». In : *CoRR* abs/1904.02811. arXiv : 1904.02811. URL : http://arxiv.org/abs/1904.02811.

VARENBERG, Michael et A VARENBERG (2012). « Table tennis rubber : tribological characterization ». In : *Tribology Letters* 47.1, p. 51-56.

VO, Xuan Thuy, Tien Dat TRAN, Duy Linh NGUYEN et Kang Hyun JO (2021). « Stair-Step Feature Pyramid Networks for Object Detection ». In : *Communications in Computer and Information Science*. T. 1405, p. 168-175. ISBN : 9783030816377. DOI : 10.1007/978-3-030-81638-4_13.

VOEIKOV, Roman, Nikolay FALALEEV et Ruslan BAIKULOV (2020). « TTNet : Real-time temporal and spatial video analysis of table tennis ». In : *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. T. 2020-June, p. 3866-3874. ISBN : 9781728193601. DOI : 10.1109/CVPRW50498.2020.00450. arXiv : 2004.09927.

WANG, Heng, Alexander KLASER, Cordelia SCHMID et Cheng-Lin LIU (juin 2011). « Action recognition by dense trajectories ». In : *CVPR. 2011 IEEE Conference on*, p. 3169-3176. DOI : 10.1109/CVPR.2011.5995407.

WANNER, Gerhard et Ernst HAIRER (1996). *Solving ordinary differential equations II*. T. 375. Springer Berlin Heidelberg New York.

WENG, Juyang, Paul COHER et Marc HERNIOU (1992). « Camera Calibration with Distortion Models and Accuracy Evaluation ». In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.10, p. 965-980. ISSN : 01628828. DOI : 10.1109/34.159901.

WIKIPEDIA (2021). *Calibration de caméra — Wikipédia*. URL : https://fr.wikipedia.org/wiki/Calibration_de_cam%C3%A9ra.

WOJKE, Nicolai, Alex BEWLEY et Dietrich PAULUS (2018). « Simple online and realtime tracking with a deep association metric ». In : *Proceedings - International Conference on Image Processing, ICIP*. T. 2017-September. IEEE, p. 3645-3649. ISBN : 9781509021758. DOI : 10.1109/ICIP.2017.8296962. arXiv : 1703.07402.

WU, Uxin, Alexander KIRILLOV, Francisco MASSA, WAN-YENLO, Ross GIRSHICK, Yuxin WU, Alexander KIRILLOV, Francisco MASSA, Wan-Yen LO et Ross GIRSHICK (2019). « Detectron2 ». In : *https ://github.com/facebookresearch/detectron2*. URL : https://github.com/facebookresearch/detectron2.

XIA, Shihong, Lin GAO, Yu Kun LAI, Ming Ze YUAN et Jinxiang CHAI (2017). « A Survey on Human Performance Capture and Animation ». In : *Journal of Computer Science and Technology* 32.3, p. 536-554. ISSN : 18604749. DOI : 10.1007/s11390-017-1742-y.

XU, Bing, Naiyan WANG, Tianqi CHEN et Mu LI (2015). « Empirical evaluation of rectified activations in convolutional network ». In : *arXiv preprint arXiv :1505.00853*.

YANG, Wenying, Juming LU, Jianping WENG, Weiping JIA, Linong JI, Jianzhong XIAO, Zhongyan SHAN, Jie LIU, Haoming TIAN, Qiuhe JI, Dalong ZHU, Jiapu GE, Lixiang LIN, Li CHEN, Xiaohui GUO, Zhigang ZHAO, Qiang LI, Zhiguang ZHOU, Guangliang SHAN et Jiang HE (2010). « Prevalence of Diabetes among Men and Women in China ». In : *New England Journal of Medicine* 362.12, p. 1090-1101. ISSN : 0028-4793. DOI : 10.1056/nejmoa0908292. URL : http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.6578%7B%5C&%7Drep=rep1%7B%5C&%7Dtype=pdf.

YIJUN, Qian, Yu LIJUN, Liu WENHE et Alexander G. HAUPTMANN (2021). « Learning Unbiased Transformer for Long-Tail Sports Action Classification. » In : *MediaEval*.

ZHANG, Harry (2004). « The optimality of Naive Bayes ». In : *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004*. T. 2, p. 562-567. ISBN : 1577352017.

ZHANG, Ning, Jingen LIU, Ke WANG, Dan ZENG et Tao MEI (2020). « Robust visual object tracking with two-stream residual convolutional networks ». In : *Proceedings - International Conference on Pattern Recognition*. IEEE, p. 4123-4130. ISBN : 9781728188089. DOI : 10.1109/ICPR48806.2021.9413110. arXiv : 2005.06536.

ZHU, Xizhou, Yujie WANG, Jifeng DAI, Lu YUAN et Yichen WEI (2017). « Flow-Guided Feature Aggregation for Video Object Detection ». In : *Proceedings of the IEEE International Conference on Computer Vision*. T. 2017-October, p. 408-417. ISBN : 9781538610329. DOI : 10.1109/ICCV.2017.52. arXiv : 1703.10025.

ZITA, Aleš et Filip ŠROUBEK (2020a). « Learning-based Tracking of Fast Moving Objects ». In : *arXiv preprint arXiv :2005.01802*. arXiv : 2005.01802. URL : http://arxiv.org/abs/2005.01802.

— (2020b). « Tracking fast moving objects by segmentation network ». In : *Proceedings - International Conference on Pattern Recognition*. IEEE, p. 10312-10319. ISBN : 9781728188089. DOI : 10.1109/ICPR48806.2021.9413129.

Analyse non intrusive du geste sportif dans des vidéos par apprentissage automatique

Résumé Dans cette thèse, nous nous intéressons à la caractérisation et à l'analyse fine de gestes sportifs dans des vidéos, et plus particulièrement à l'analyse non-intrusive 3D en vision mono caméra. Notre cas d'étude est le tennis de table.

Nous proposons une méthode de reconstruction des positions 3D des balles en utilisant une caméra rapide (240 fps) calibrée. Pour cela, nous définissons et entraînons un réseau convolutif qui permet d'extraire des images le diamètre apparent de la balle. La connaissance du diamètre réel de la balle permet de calculer la distance caméra/balle puis de positionner cette dernière dans un repère 3D lié à la table.

Ensuite, nous utilisons un modèle physique, prenant en compte l'effet Magnus, pour estimer les paramètres cinématiques de la balle à partir de ses positions 3D successives. La méthode proposée segmente les trajectoires à partir des impacts de la balle sur la table ou la raquette, ce qui permet, en utilisant un modèle physique de rebond, d'affiner les estimations des paramètres cinématiques de la balle puis de calculer la vitesse et l'angle de la raquette lors de la frappe et d'en déduire des indicateurs de performance pertinents.

Deux bases de données ont été construites : la première est constituée d'acquisitions de séquences réelles de jeu et la seconde, synthétique, reproduit les conditions d'acquisition de la première et permet de valider nos méthodes, les paramètres physiques utilisés pour la générer étant connus.

Enfin, nous présentons notre participation à la tâche Sport\&Vision du challenge MediaEval sur la classification d'actions humaines, par des approches basées sur l'analyse et la représentation du mouvement.

Mots clefs : Analyse vidéo, Reconstruction 3D monovision, Apprentissage automatique, Performance sportive, Reconnaissance d'actions humaines, Tennis de table

Non-intrusive analysis of sports gestures in videos using machine learning

Abstract : In this thesis, we are interested in the characterization and fine grained analysis of sports gestures in videos, and more particularly in non-intrusive 3D analysis using a single camera. Our case study is table tennis.

We propose a method for reconstructing 3D ball positions using a high-speed calibrated camera (240 fps). For this, we propose and train a convolutional network that extracts the apparent diameter of the ball from the images. The knowledge of the real diameter of the ball allows us to compute the distance between the camera and the ball, and then to position the latter in a 3D coordinate system linked to the table.

Then, we use a physical model, taking into account the Magnus effect, to estimate the kinematic parameters of the ball from its successive 3D positions. The proposed method segments the trajectories from the impacts of the ball on the table or the racket. This allows, using a physical model of rebound, to refine the estimates of the kinematic parameters of the ball. It is then possible to compute the racket's speed and orientation after the stroke and to deduce relevant performance indicators.

Two databases have been built: the first one is made of real game sequence acquisitions. The second is a synthetic dataset that reproduces the acquisition conditions of the previous one. This allows us to validate our methods as the physical parameters used to generate it are known.

Finally, we present our participation to the Sport\&Vision task of the MediaEval challenge on the classification of human actions, using approaches based on the analysis and representation of movement.

Keywords : Video analysis, 3D monovision reconstruction, Machine learning, Sports performance, Human action recognition, Table tennis

Laboratoire Mathématiques, Image et Applications
Avenue Michel Crépeau
17042 La Rochelle Cedex 01