

SkateboardXXX3000

Faire de l'analyse de données de skate

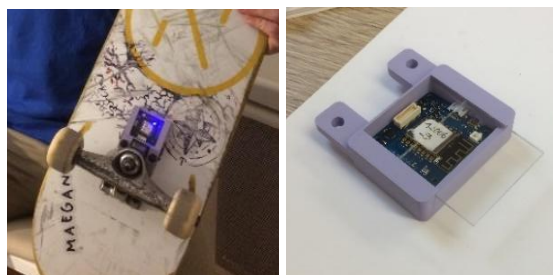


Le projet

SkateboardXXX3000 est un capteur (**Movuino**) **d'accélération** et **gyroscopique (vitesse angulaire)** placé sous le skate permettant d'analyser les **mouvements simples**¹ qu'un skateur peut faire. J'ai démarré ce projet l'année dernière durant ma césure, et je l'ai continué en tant que projet de fin d'étude cette année avec un camarade. Nous avons orienté notre travail sur la **reconnaissance des figures**. Nous sommes très fiers des premiers résultats que nous avons obtenus, mais il reste encore beaucoup à faire et à améliorer.

Les motivations

Quand j'ai parlé de ce projet à mes encadrants de stage, Joël et Adrien, je voyais en lui l'opportunité de challenger vos capacités de futur ingénieur sur un sujet qui me passionne depuis maintenant 2 ans et demi. Très intéressé par la programmation et l'analyse de données, avec l'aide de nombreuses personnes, nous avons monté brique par brique les premières étapes du projet. Ces étapes vont de la confection d'un **boîtier** se fixant sur le skate à la **reconnaissance de figures** en passant par la **visualisation 3D** des figures. Au-delà de la simple reconnaissance de figures qui pourraient s'effectuer avec de l'analyse vidéo, le capteur placé sous le skate permet en plus d'avoir des informations sur la figure **du point de vue du skate**, et donc centré sur son mouvement.



Le movuino placé dans le boîtier et sur le skate

Les enjeux

Sportifs

Prévention de la blessure, étude de la performance, apprentissage de la discipline

Pédagogiques

Etude et compréhension de la discipline avec une approche scientifique

Les objectifs

La reconnaissance de figures

Détection de figures, classification...

Extraction des données pertinentes pour la pratique du skate

Hauteur, évaluation de la progression, étude des chocs...

¹ Simplicité : Action simple voir instinctive pour la personne qui l'exécute mais en réalité très complexe

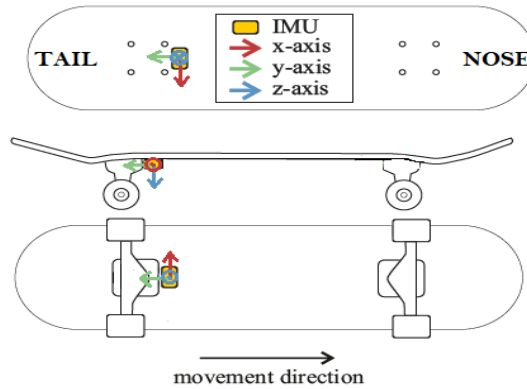


Schéma du movuino et de ses axes sur le skate

L'approche générale

Simple mais efficace

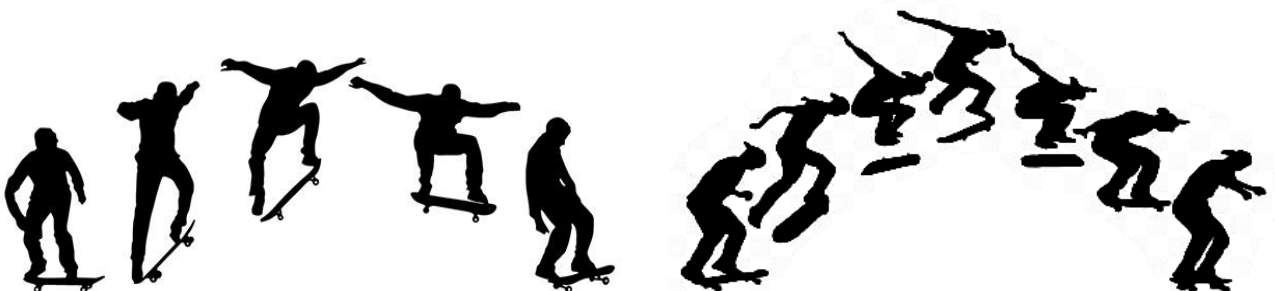
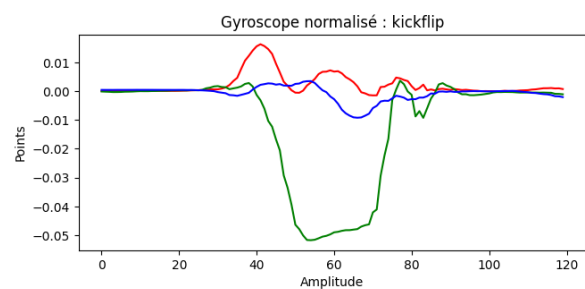
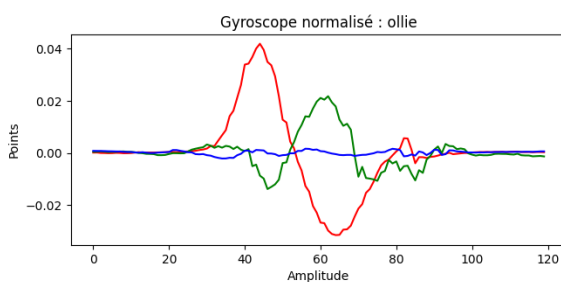
Dans l'état de l'art, nous pouvons trouver beaucoup d'articles scientifiques ayant traité de problématiques similaires. Cependant nous avons remarqué que peu font le lien entre la discipline et leurs données. Les données sont pourtant très **visuelles** et **compréhensibles**. C'est pourquoi nous avons abordé une démarche simple, voire naïve, afin d'exploiter au maximum les données extraites.

Dans un premier temps, nous nous intéressons principalement aux :

- Ollie
- Kickflip
- Heelflip
- Pop shovit
- Frontside shovit
- 360 flip

Car leur **mouvement est bien différentiable** autant dans les données que de manière visuelle.

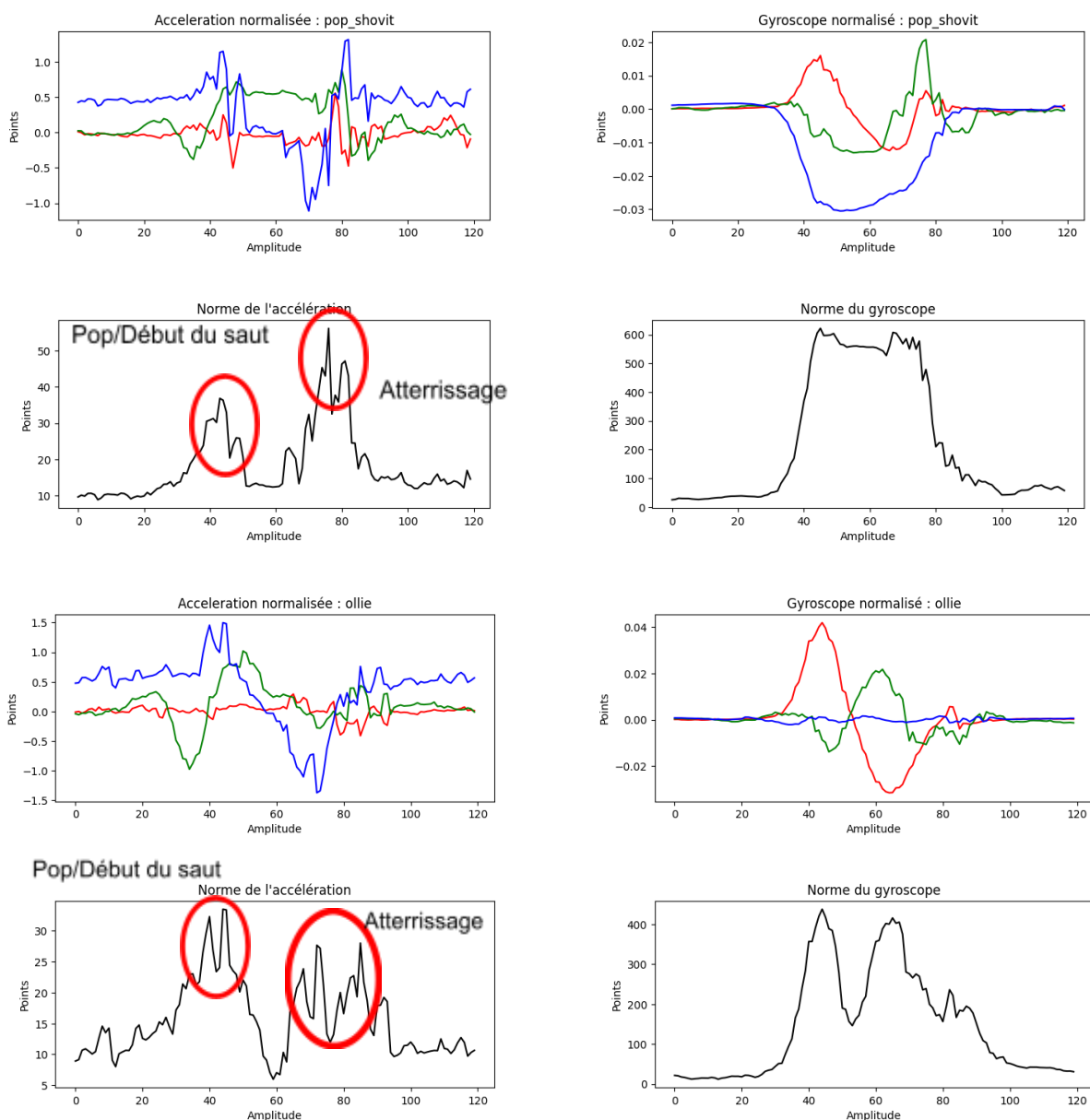
Ci dessous voici deux exemple de données gyroscopiques que nous avons obtenu pour un ollie et un kickflip:



Remarques :

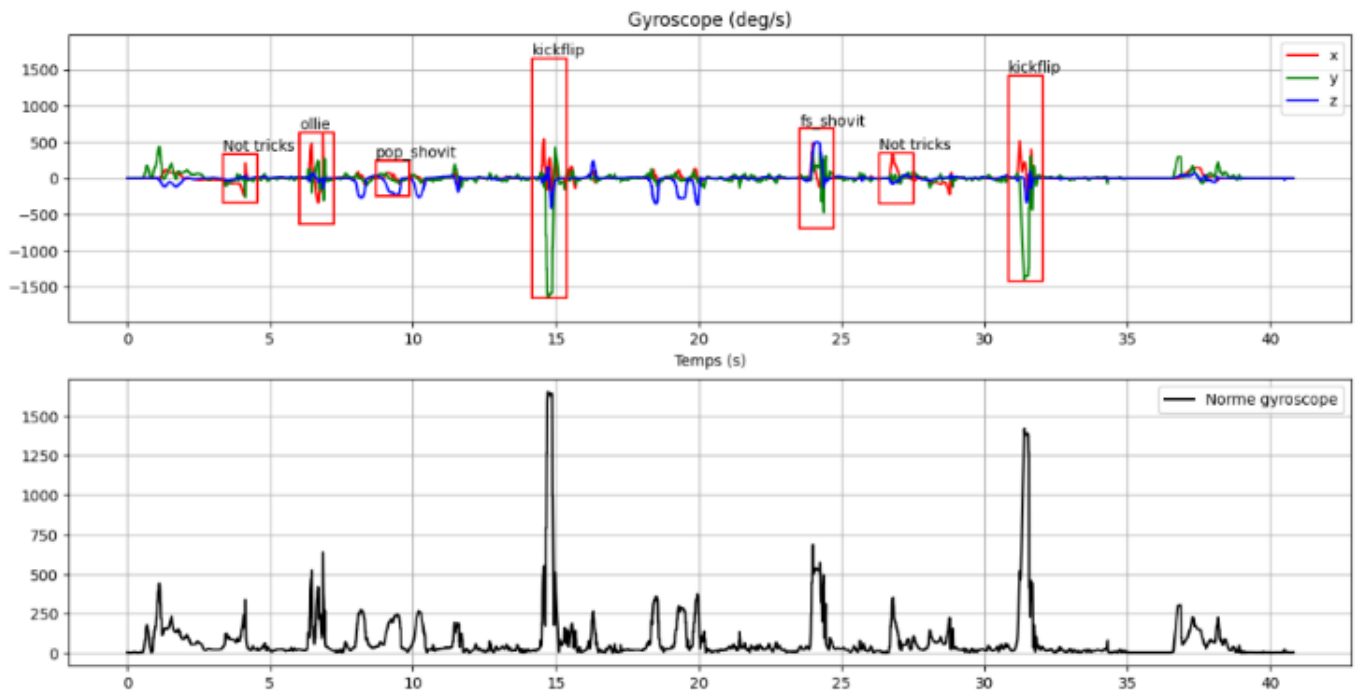
Les courbes rouges, vertes et bleues correspondent respectivement aux axes x, y et z du gyroscope.

Pour les figures choisies, de manière générale on remarque que les données gyroscopiques vont donner plus d'informations sur le **type de figure effectuée**, tandis que les données d'accélération vont permettre d'apporter des **informations supplémentaires** comme la hauteur etc. En effet, en comparant les données gyroscopique d'un shovit et un d'un ollie, nous remarquons qu'il est plus facile de différencier les deux, qu'avec les données d'accélération. Cependant, nous pouvons obtenir des informations supplémentaires grâce à l'accélération comme par exemple, le moment du pop et l'atterrissage de la figure qui sont identifiables en regardant les pics d'accélération.



En se référant au schéma du capteur placé sur le skate on peut voir que, grossièrement, l'axe x du gyroscope va comporter les données correspondant au **pop du skate**, **y au flip** et **z aux shovit** et aux 180. Ce qui s'observe très bien sur nos graphes.

Pour la suite du document, je présente les travaux effectués durant cette année sur la reconnaissance de figures.



Exemple de sortie de l'algorithme de la reconnaissance de figures

La reconnaissance de figures

Pour effectuer la reconnaissance de figures, nous avons dissocié la tâche en deux : **la détection d'événements**, et **la classification des événements détectés**. Cela permet de dissocier la performance de détection et de classification. De plus, cela permet de réduire la quantité de données à traiter par notre algorithme.

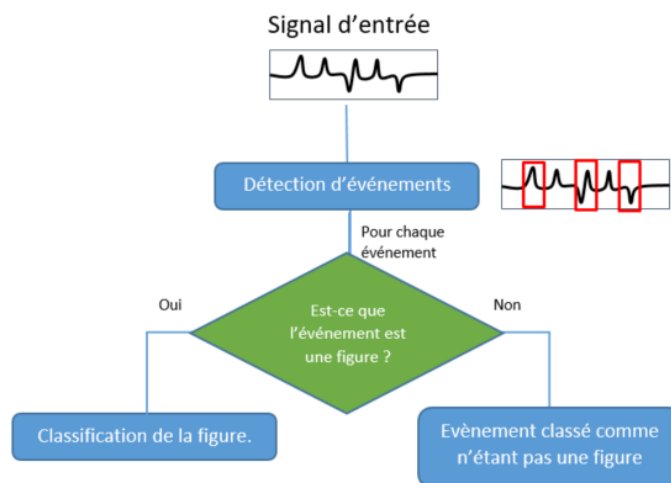


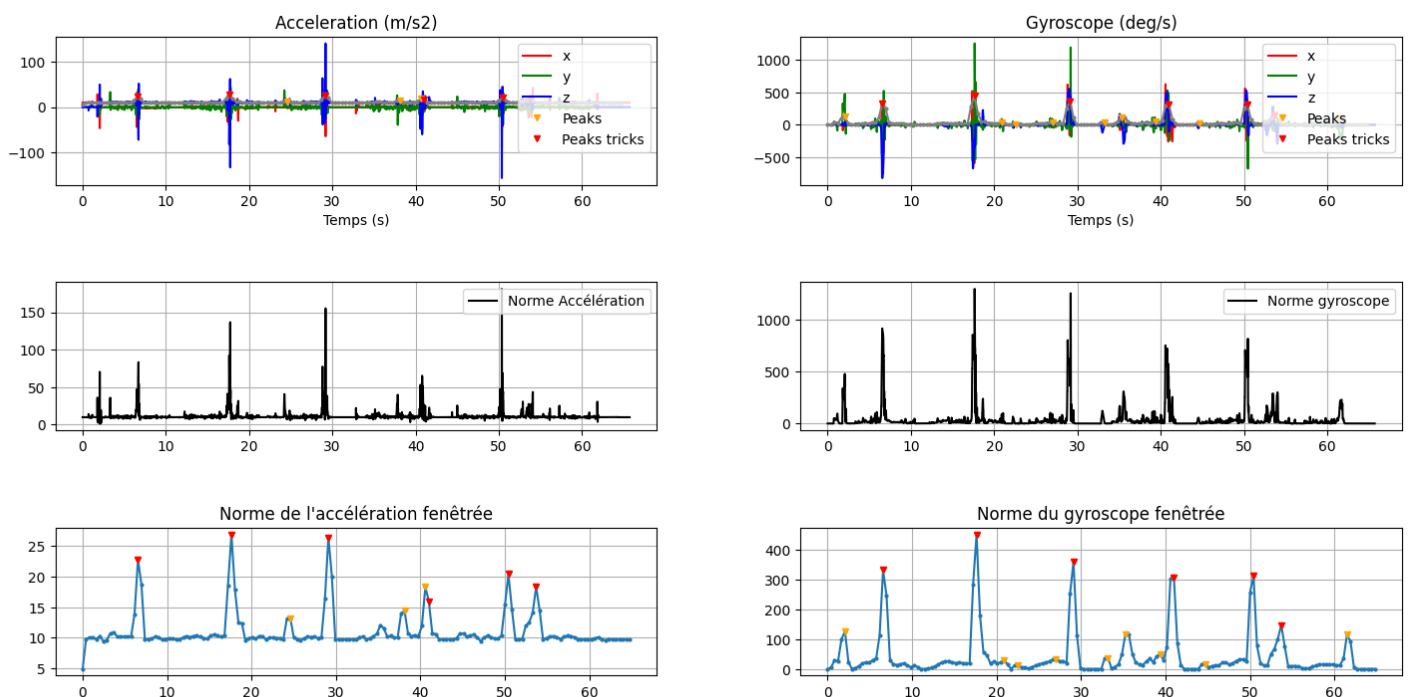
Schéma de l'algorithme

La détection d'évènement

Pour effectuer cette tâche, nous avons utilisé **l'énergie** présente dans les données d'accélérations et gyroscopiques. En effet, des pics sont présents quand le skateur effectue une figure, car il apporte de l'énergie au skate. Cependant, le skateur peut aussi apporter de l'énergie au skate sans pour autant effectuer une figure (virage, pick-up...etc), ces événements parasites dans la reconnaissance de figures étaient donc détectés.

Ci-dessous voici un exemple de résultat obtenu suite à la détection d'évènement.

De bas en haut nous affichons : les canaux x,y,z bruts, la norme de ces canaux et tout en bas le résultat de la détection d'événements, **les triangles rouges indiquent que les pics sont des événements à observer de plus près**. A droite et à gauche nous affichons respectivement les courbes du gyroscope et les courbes de l'accélération.



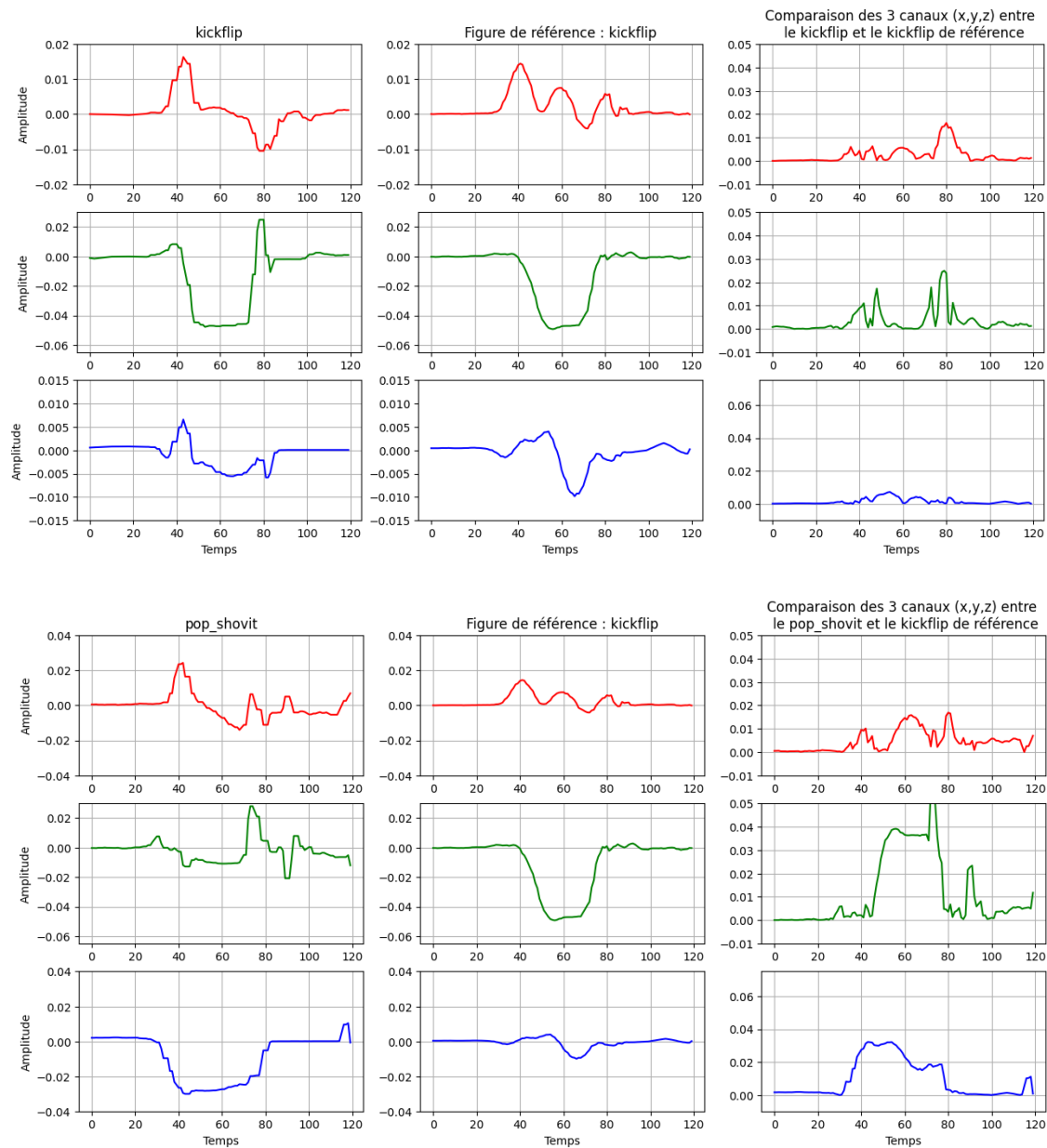
Une fois les événements détectés, nous avons effectué la reconnaissance de figures sur ces derniers.

Tri des événements et classifications

D'une pierre 2 coups

Comme évoqués précédemment, **beaucoup d'événements détectés n'étaient pas des figures**. Cependant, notre méthode de classification permettait d'effectuer un tri. En effet, afin de savoir quelle figure correspondait à l'événement détecté, nous avons **comparé ce dernier à des figures de référence**. La figure correspondant à l'événement était celle qui « **ressemblait le plus à l'événement** ». Dans notre cas, nous avons choisi la figure de référence qui minimise la **distance euclidienne** avec l'événement. La figure était attribuée si la distance était en dessous d'un certain **seuil**, sinon elle était considérée comme un **mouvement de transition** (virage, pick-up, etc.).

Pour illustrer ceci, voici un exemple de comparaison entre un *kickflip* et le *kickflip* de référence ainsi qu'entre un *pop shovit* et le *kickflip* de référence :



De haut en bas : comparaison *kickflip*/*kickflip* de référence et *popshovit*/*kickflip* de référence

De gauche à droite : Figure à comparer - *kickflip* de référence - soustraction de chacun des channels x,y,z

On peut voir dans les deux comparaisons que le *kickflip* ressemble plus au *kickflip* de référence que le *popshovit* sur les axes y et z car la différence entre les deux courbes est plus stable. En effet le *pop shovit* a un mouvement très porté par z (bleu), tandis que le *kickflip* est très porté par y (vert), d'où une différence importante sur ces 2 canaux. Malgré des axes x (rouge), plus similaires, on montre bien que l'on peut différencier des figures entre elles avec cette méthode qui est assez simple de compréhension et d'implémentation.

Résultats

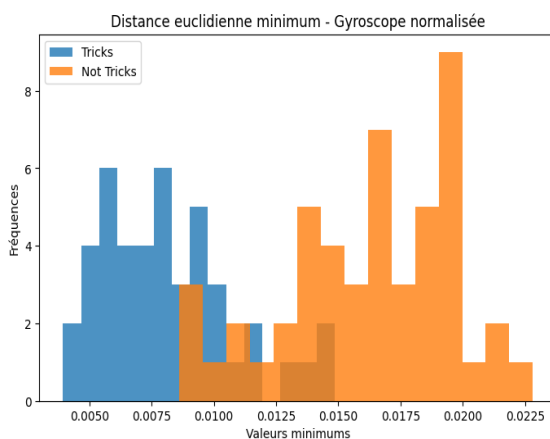
Détection d'événements

Pour nos différents enregistrements, nous avons détecté **162 événements**. Sur ces événements, **113 correspondaient à des figures** et **49 à des mouvements de transitions parasites**.

N = 162	Positive	Négative
Positive	113 (TP)	0 (FN)
Négative	49 (FP)	N/A

Pour le réglage de la partie détection d'événements, nous avons cherché à détecter toutes les figures (**sensibilité de 1**), cependant sur ces événements détectés nous avons **précision de seulement 70%**.

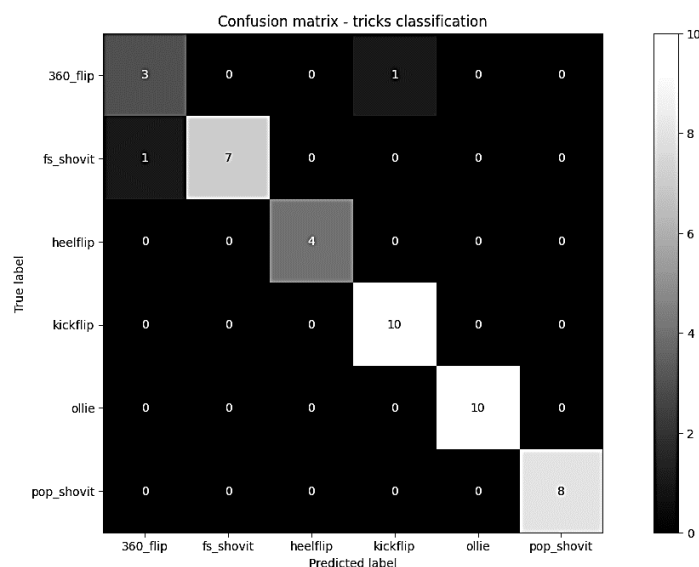
Avec le tri effectué durant la classification, nous avons réussi à améliorer les résultats de la détection en appliquant un **seuil sur la distance minimale**. En effet, le seuil que nous avons choisi maximise le f1 score (arbitraire) et permet d'obtenir en plus **une précision de 90.6%** et un **rappel de 87.7% (représente le taux d'événements oubliés)**.



Histogramme des distances minimales des figures et des non-figures

Classification

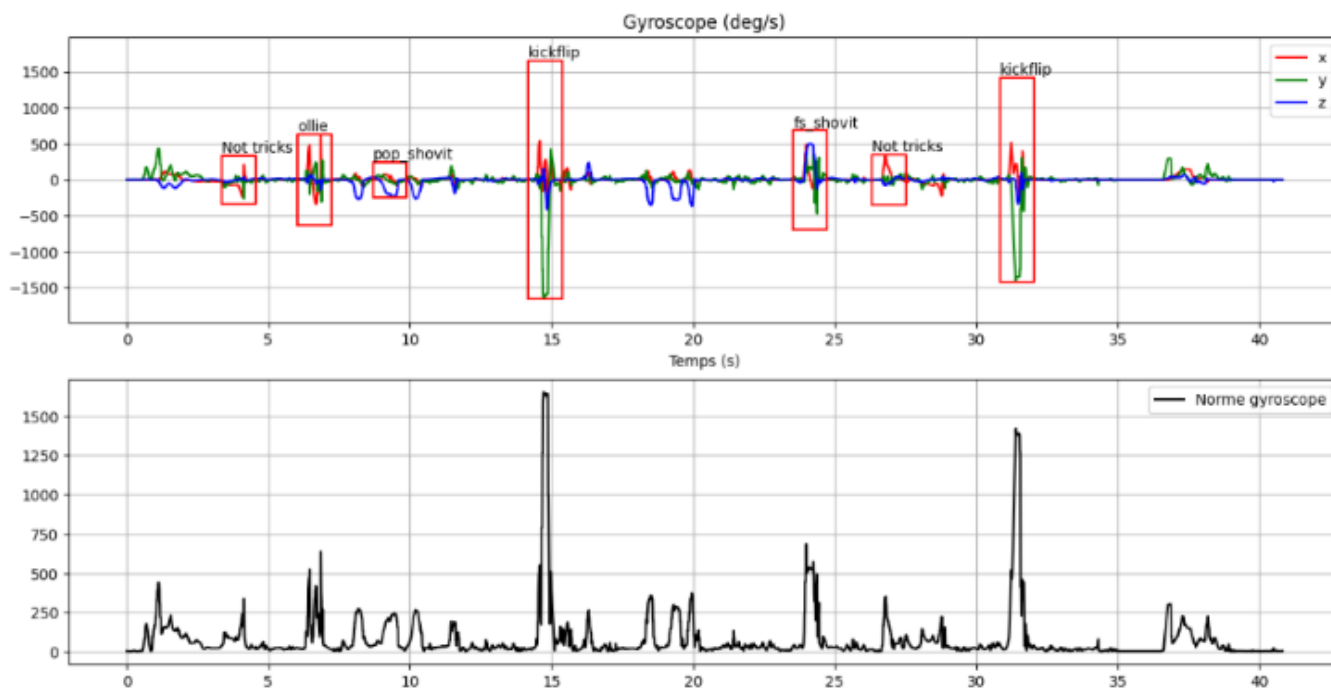
Sur les **6 figures choisies**, notre algorithme de classification par minimisation d'une distance était très performant. Sur **44 figures testées**, seulement 2 ont donné le mauvais résultat :



Matrice de confusion de la classification

La sortie de l'algorithme

Les performances de nos sous-algorithmes étant satisfaisantes, nous avons choisi de les tester sur une séquence de figures complètes, et nous avons obtenu en sortie :



Sortie de l'algorithme de classification

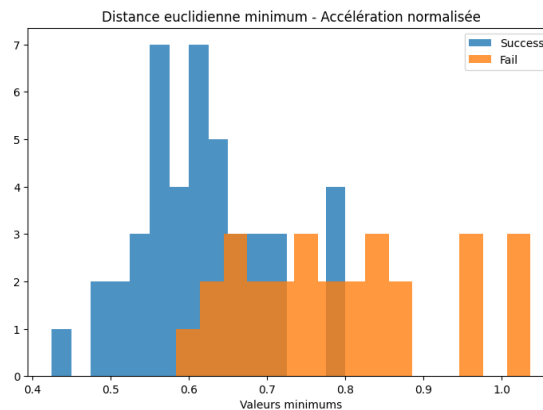
On peut voir sur cette courbe les rectangles qui correspondent aux événements détectés. Nous avons mis la mention *not_tricks* sur les figures situées au-dessus du seuil. Nous pouvons remarquer la présence d'un pop-shovit, qui n'en est pas un. En effet, l'événement correspond à un virage, et possède une courbe

similaire à celui du pop-shovit, ce qui fait que l'on a détecté comme tel (mouvement très porté par z). C'est une des limites de notre algorithme.

Succès d'une figure

Avec la même méthode du seuil, nous avons essayé de différencier les **figures ratées** et **non ratées**.

Voici l'histogramme représentant la distance minimale avec l'accélération pour les figures ratées et non ratées :



Histogramme des distances minimales des figures ratées et non ratée

On remarque que les deux groupes se séparent un peu. Un seuil peut donc être appliqué pour estimer si une figure est ratée ou non. En maximisant le f1 score, nous obtenons une précision de 80% et un rappel de 80% environ aussi. Ces résultats sont bons, mais à creuser, car nous ne prenons pas en compte le fait que la figure soit ratée connaissant la figure.

Limites de notre approche

Méthode trop « simple »

Perte d'informations en « fabricant » les figures de références, on évalue seulement la globalité de la figure.

Beaucoup d'événements parasites

Malgré les améliorations dues au seuil, nous avons encore la présence d'événements parasites. Cependant, nous connaissons et pouvons classifier la plupart des événements,

Penser à classifier les événements parasites seraient une première idée ?

Idées pour la suite

Tester l'algorithme avec d'autres figures

Tester des algorithmes plus performants

Etudier des figures impliquant la rotation du corps

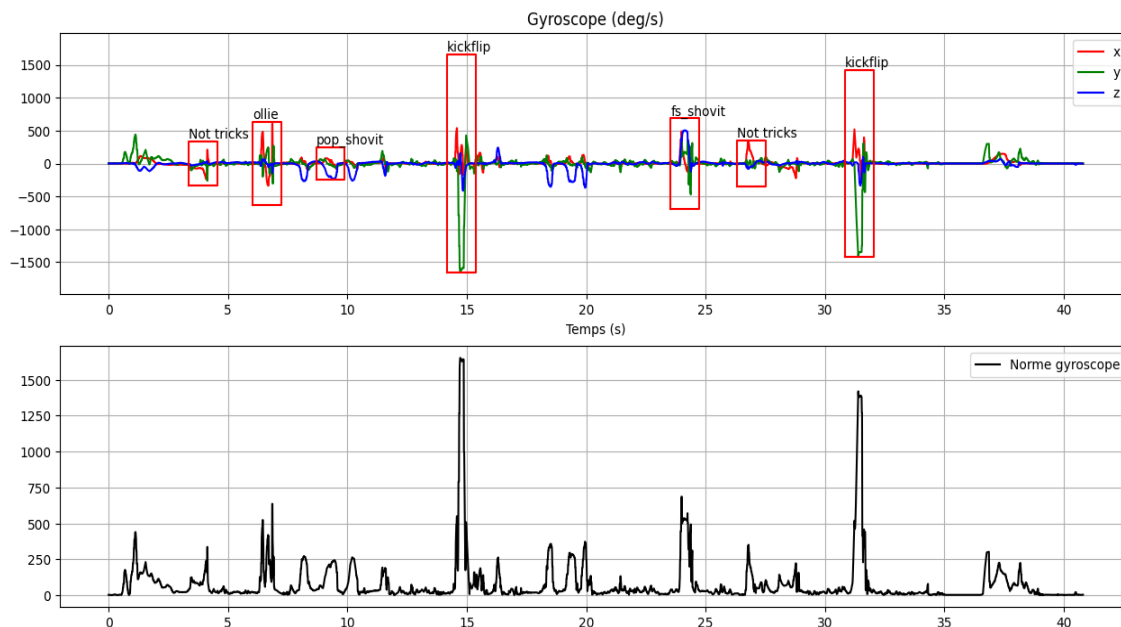
Etudier plus en profondeur le lien skate/données

Trouver des critères permettant d'évaluer des figures (timing, hauteur...)

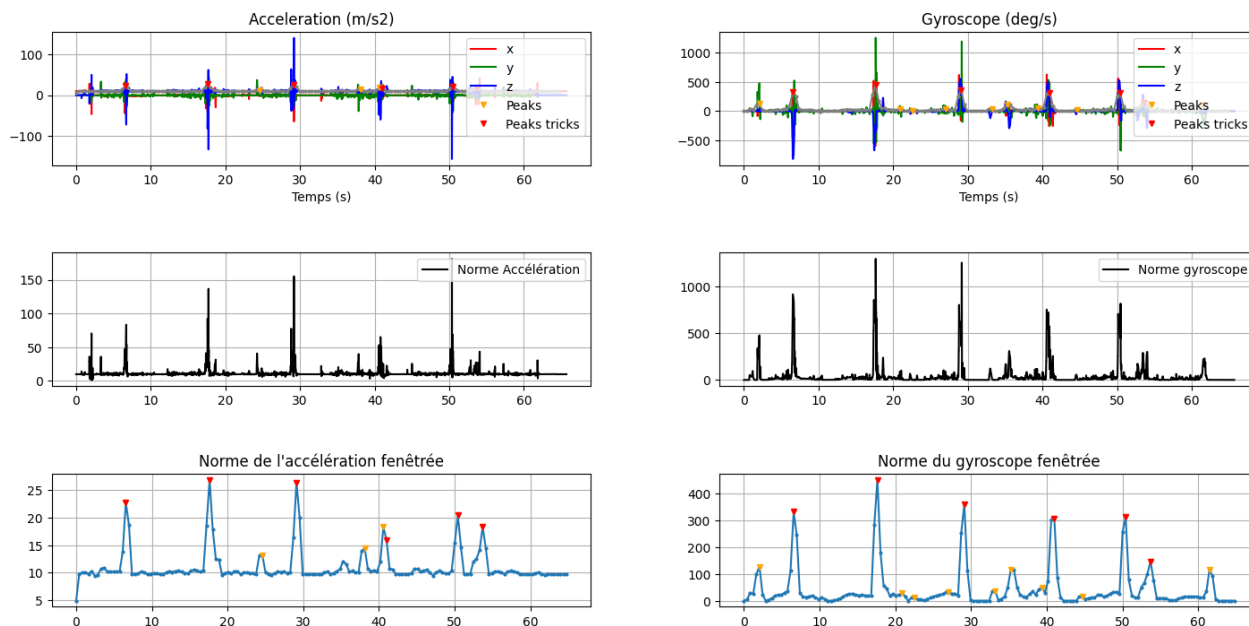
Remerciements

Pour ce projet je tiens à remercier Joël Chevrier et Adrien Husson qui m'ont aiguillé et m'ont laissé du temps pour démarrer ce projet. Je remercie également Alexandre Singier pour m'avoir aidé dans la confection des boîtiers contenant le capteur et Paul Montecot pour avoir appuyé les réflexions sur la partie machine learning. Je remercie Julien Douxami pour m'avoir accompagné durant le projet de fin d'étude ainsi que notre enseignant encadrant Eric LECARPENTIER. Ainsi que toutes les autres personnes qui ont participé de près ou de loin au projet.

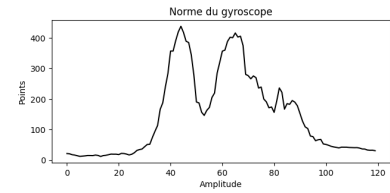
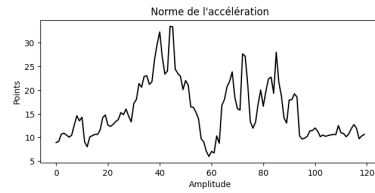
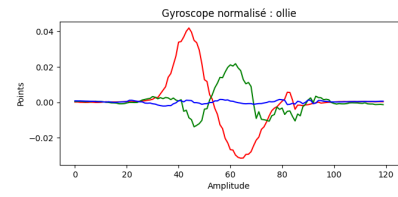
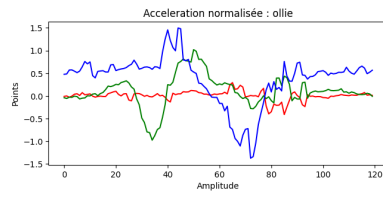
Annexes



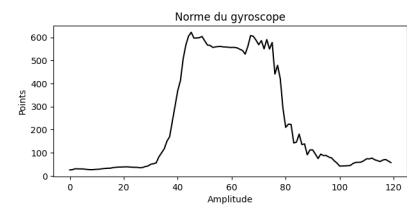
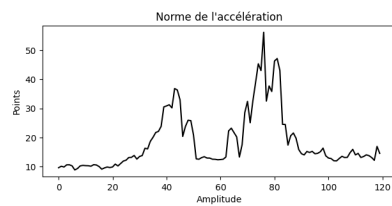
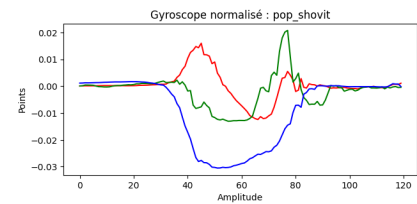
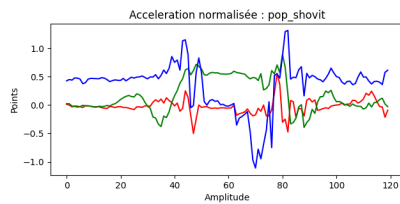
Sortie de l'algorithme - Zoom sur le gyroscope



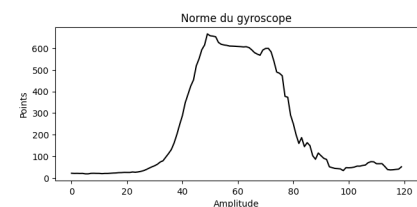
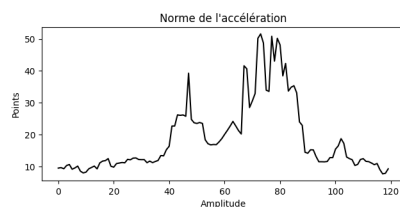
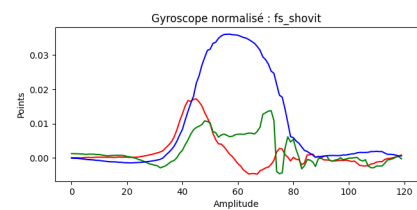
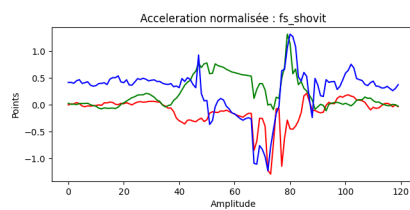
Sortie de l'algorithme de détection d'événements



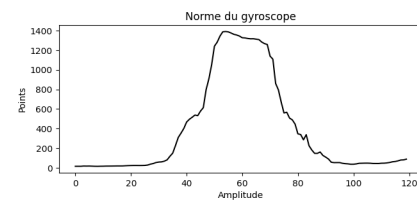
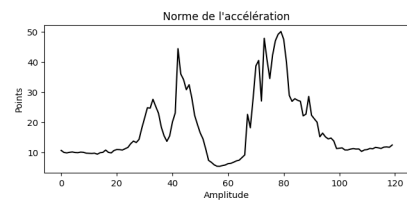
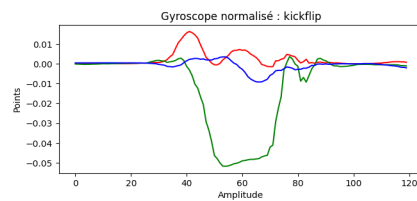
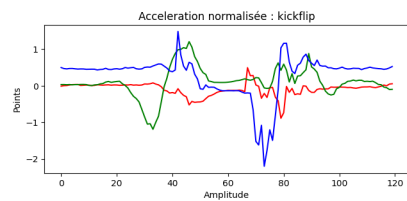
Ollie – Figure de référence



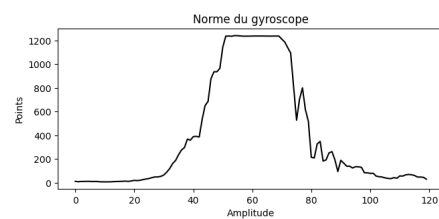
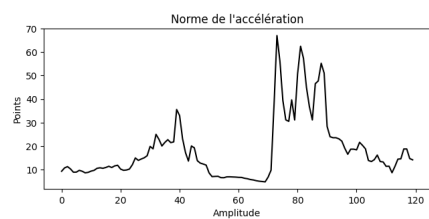
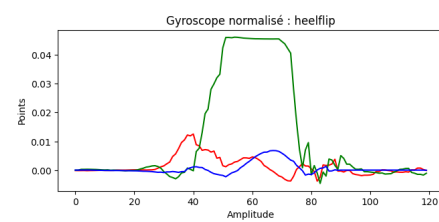
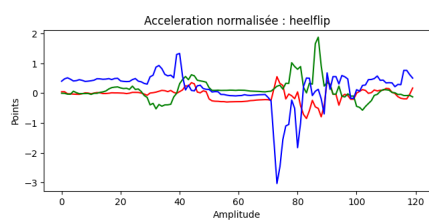
Pop shovit – Figure de référence



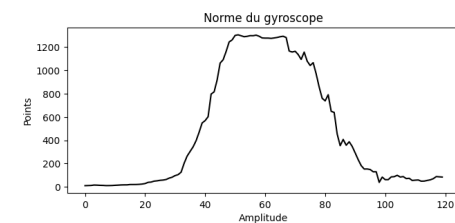
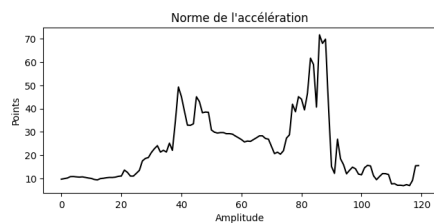
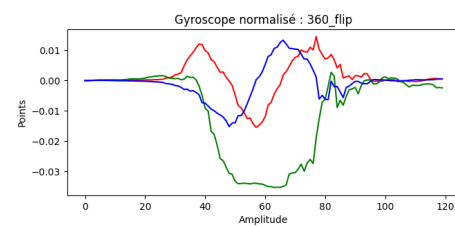
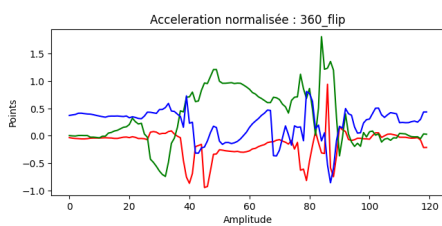
Fs shovit – Figure de référence



Kickflip- Figure de référence



Heelflip – Figure de référence



360 flip – Figure de référence