| Dauce Julien | Georget Thomas |
| --- | --- |
| Hortua Carlos | Jebari Aymane |
| Maillard Rory | Maréchal Luc |



Document de passation

IA Racing

# 

# 

# 

# 

# Objectifs du document

Le présent document vise à préciser les étapes suivies et l’organisation du travail lors du développement de la voiture autonome.

[**Objectifs du document 1**](#_dareoa3rwczp)

[**I - Contexte du projet 3**](#_g911kumu3kol)

[Objectifs du projet 3](#_1d2jswkn1xte)

[Organisation temporelle et échéances 3](#_riss9dwzv41d)

[Répartition des tâches et organisation du groupe 4](#_j3ukt5nvjszw)

[**II - Computer Vision 5**](#_i890qbvkptpn)

[Choix 5](#_dbbqn3ux8zm3)

[Correction de l’effet fish-eye 5](#_u5b6n69n0n3)

[Filtres appliqués à l’image 5](#_z3kzz6mkwmms)

[Calcul de l’écart au centre de piste 5](#_bt8kiuve8rkf)

[Expériences 5](#_qx2ne0xeffh6)

[Observations 6](#_mob5naf2ybc)

[Correction du fish-eye 6](#_j7m0v03ae0l0)

[Filtres appliqués à l’image 7](#_dnluc2ezxfha)

[Calcul de l’écart au centre de piste 7](#_pu5hjjpc4x3p)

[Conclusion et pistes 8](#_s9xv60tq74y7)

[**III - Modèle par renforcement 8**](#_8khov96s9yfd)

# I - Contexte du projet

## Objectifs du projet

Le projet IA Racing a pour objectif global de mettre en place un modèle de voiture autonome s’appuyant sur une conduite par apprentissage. Nous héritons alors du travail réalisé par le groupe précédent sur l’année 2022/2023, qui avait réussi à réaliser un modèle par apprentissage fonctionnel et avait tenté de réaliser un modèle par apprentissage dont le résultat était moins performant.

L’objectif de cette année sera de développer un modèle d’apprentissage par renforcement et un traitement d’image pour récupérer certaines primitives de la route, proposées par M. Claveau.

Le projet s’organise donc autour de ces deux axes principaux, chacun se décomposant en plusieurs composantes secondaires qui se sont précisées dans l’année.

* L’apprentissage par renforcement
  + Détermination de l’algorithme à utiliser
  + Choix de la fonction de reward
  + Choix du modèle
* Le traitement d’image
  + Correction de l’effet fish-eye
  + Applications de filtres
  + Calcul de l’écart au centre de la piste

Les objectifs de résultat sont similaires à ceux de l’année précédente, i.e. les véhicules devront :

1. démarrer
2. Rouler sans erreurs sur plusieurs tours
3. Être robuste à des modifications de l’environnement

Contrairement à l’année précédente, nous ne participerons pas à la course organisée par Sigma, nous n’aurons donc pas la contrainte des objectifs associés.

## Organisation temporelle et échéances

Tout au long du projet, nous avons également dû produire les différents livrables attendus par IMT Atlantique, individuels ou communs afin de permettre l’évaluation en mode projet, estimant entre autres notre capacité de planification des tâches, de suivi des étapes et la communication interne et externe au groupe.

Il est notamment demandé de faire un diagramme de Gantt pour fixer les jalons et prendre du recul sur notre organisation. Les différentes tâches ont donc dû être décomposées, planifiées et ordonnées en amont du projet, puis comparées aux temps effectivement réalisés pour leur complétion. Cela permet un suivi facile des retards, et une vision globale des échéances à venir. Des points méthodologiques étaient notamment prévus afin de nous accompagner dans notre gestion de projet et nous permettant de nous assurer de notre alignement avec les attendus de l’école. Ces échéances ont rythmé notre organisation en nous imposant des tâches précises.

En parallèle, il était nécessaire de tenir des points réguliers avec nos deux tuteurs projet, M. Claveau et M. Hache. Ceux-ci nous ont accompagnés dans notre gestion des aspects techniques, comme la compréhension des réseaux de neurones et la formalisation du problème d’optimisation correspondant à l’apprentissage.

Le livrable final prend la forme d’un poster pour le forum des projets ProCom afin de présenter le travail effectué aux étudiants des autres années.

## Répartition des tâches et organisation du groupe

Le groupe travaillant sur le projet est composé de 6 étudiants. Nous avons donc décidé de nous répartir en 2 groupes de 3 personnes travaillant respectivement sur la computer vision et sur l’apprentissage par renforcement, ce qui nous a permis de paralléliser les tâches et gagner du temps, les deux objectifs pouvant progresser indépendamment pendant une grande partie du projet.

Nous nous assurions tout de même de maintenir une communication continue et efficace entre les deux parties, notamment grâce à différents moyens selon les besoins :

Gestion de projet

* **Gantt** (avec Gantt Project) : Planification des tâches sur le long terme

Partage des documents :

* **Google Drive** : Réalisation des livrables et partage de fichiers
* **Github** : Partage des codes et des fichiers

Communication :

* **Messenger** : Messagerie instantanée permettant de garder le contact en dehors des rendez-vous journaliers et de communiquer sur nos avancement
* **Discord** : Messagerie qui nous a permis de communiquer rapidement avec les encadrants ainsi que l’autre équipe et de stocker des ressources utiles au projet.
* **Mail** : Communication formelle

# 

# II - Computer Vision

Afin de faire concorder le simulateur et la réalité, réduire l’influence du bruit et simplifier le modèle en réduisant le nombre de paramètres en entrée, nous avons implémenté une partie Computer Vision pour prétraiter l’image d’entrée du modèle.

## Choix

### Correction de l’effet fish-eye

La caméra utilisée sur la voiture crée un effet fish-eye. Il est donc nécessaire de le corriger pour réutiliser les résultats du simulateur.

Cela se fait en calibrant la caméra à l’aide d’une cible de calibration (un échiquier) nous permettant d’obtenir la matrice des déformations dues aux lentilles.

### Filtres appliqués à l’image

Dans les prises de vue réelles, de nombreuses sources de bruit sont présentes, qu’il est nécessaire de corriger. Nous pouvons par exemple citer les variations de luminosité et les reflets sur la piste qu’il est nécessaire d’éliminer afin de détecter correctement les limites de la piste.

Nous avons choisi de combiner plusieurs filtres (inRange, Canny, birdview) afin de donner des informations complémentaires au réseau.

### Calcul de l’écart au centre de piste

Une primitive importante pour connaître notre position sur la piste et la décision de direction et vitesse à prendre est l’écart au centre de piste. Cette valeur est directement connue dans le simulateur, mais doit être calculée lors des roulements sur piste réelle.

Pour cela, nous utilisons un second réseau de neurones de petite taille entraîné sur simulateur, prenant en entrée une image légèrement filtrée (uniquement un inRange).

## Expériences

A chaque amélioration de notre preprocessing, nous faisions des tests des résultats sur piste réelle. Nous pouvions ainsi nous assurer que nos changements donnaient des résultats concrets et cohérents avec ce que nous souhaitions.

A chaque expérience, nous avons suivi le protocole suivant :



Pour optimiser notre conduite, nous connectons une manette de ps4 pour mieux conduire qu’avec la manette fournie par défaut ou le contrôleur web. Lors de la course nous essayons d’être le plus fluide possible et de ne pas trop sortir des lignes. Les images récupérées sont automatiquement enregistrées dans un nouveau répertoire.

## Observations

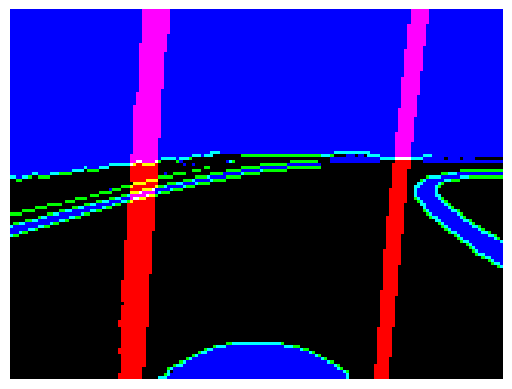
Etant donné que nous avons testé nos corrections individuellement, nous pouvons discuter de leur résultat sans les corréler.

### Correction du fish-eye



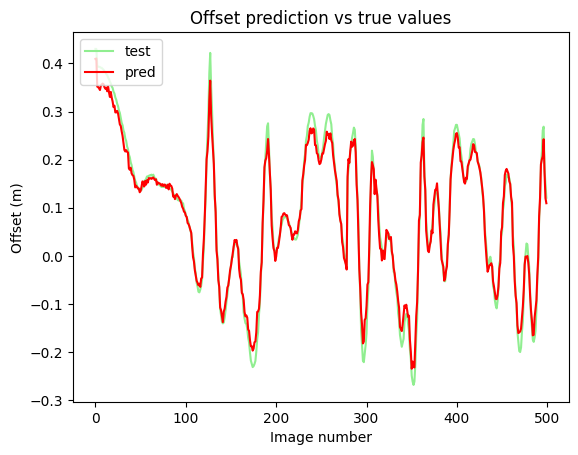
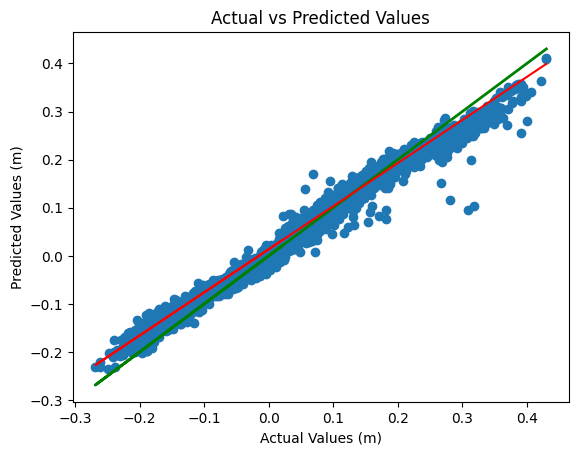
La correction de l’effet fish-eye de la caméra marche correctement. Nous retrouvons une image non déformée de la route, qui permet de transférer de manière plus correcte les modèles entre simulateur et réel.

### Filtres appliqués à l’image

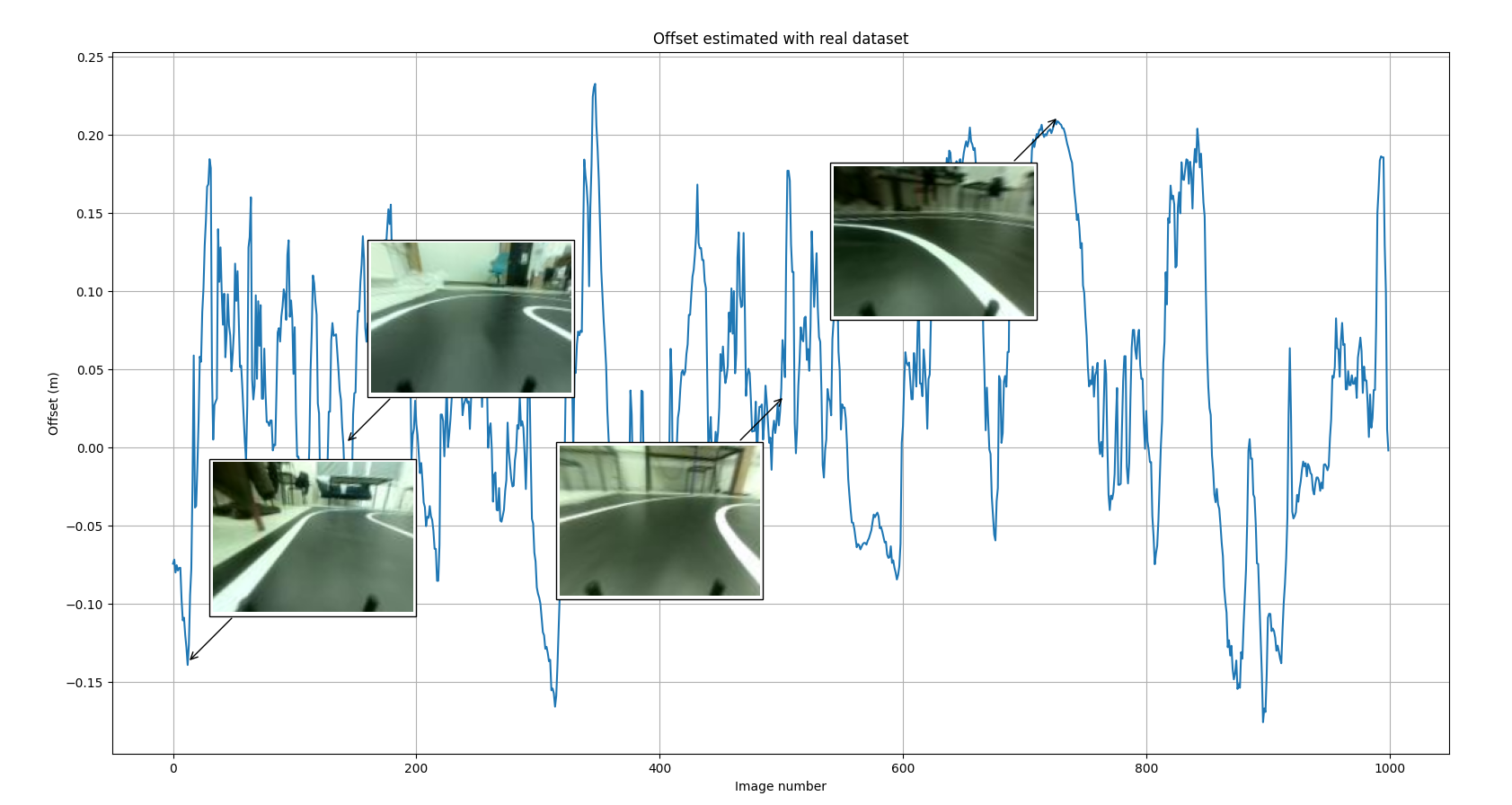


La combinaison des trois filtres sur les canaux RGB de l’image nous permet de distinguer correctement les lignes de la piste. Cependant, lors de notre présentation au forum, nous avons pu constater que les reflets sur la piste et sur le sol pouvaient parfois ne pas être filtrés. Après une adaptation des seuils des différents filtres, nous avons pu améliorer le résultat au forum. Nous en retenons cependant que les valeurs utilisées doivent être testées dans différentes conditions pour ne pas se spécialiser dans une unique situation (dans notre cas fenêtres fermées et lumières allumées).

### Calcul de l’écart au centre de piste



Nous avons d’abord utilisé le réseau de neurones en simulateur. On retrouve ici deux représentations des résultats prédits et réels, à gauche une représentation temporelle, et à droite une représentation comparative. On constate que les valeurs prédites sont cohérentes avec les valeurs cherchées, à l’exception des valeurs extrêmes, où il a tendance à légèrement réduire les valeurs. Le réseau de neurones fournit donc un résultat que nous considérons comme satisfaisant en simulateur.



Nous avons ensuite testé le réseau sur un dataset d’images réelles corrigées par l’effet fish-eye. Le résultat ci-dessus nous montre une très bonne corrélation entre les valeurs obtenues (la piste ayant une largeur de 40cm) et la position réelle de la voiture. Il faut cependant être attentif, puisque certaines valeurs sont erronées (voir 3e image ci-dessus, ou la voiture est clairement à droite de la piste mais le CTE pratiquement nul).

## Conclusion et pistes

Finalement, les objectifs de la partie CV du projet ont été partiellement remplis. Nous avons réussi à gommer les différences entre le simulateur et la réalité à l’aide des différents filtres appliqués, et nous avons réussi à déterminer l’une des primitives de la piste : l’écart au centre de voie. Cependant, nous n’avons pas réussi à reconstruire le rayon de courbure de la piste ainsi que les angles proches et lointains que nous souhaitions obtenir au départ. Une idée pour obtenir ces différentes valeurs serait d’entraîner des réseaux de neurones, mais nous n’avions plus le temps nécessaire pour labelliser l’ensemble des images requises pour l'entraînement de tels réseaux.

# III - Modèle par renforcement