Apprentissage par renforcement profond de la fixation binoculaire en utilisant de la détection d'anomalies

Learning of Binocular Fixations using Anomaly Detection with Deep Reinforcement Learning

François de La Bourdonnaye[1] Céline Teulière[1] Jochen Triesch[2] Thierry Chateau[1]

[1] Institut Pascal, CNRS UMR 6602, Aubière, France
[2] Frankfurt Institute for Advanced Studies, Germany

Résumé

Par leur capacité à apprendre des comportements visuo-moteurs complexes, les algorithmes d'apprentissage par renforcement profond ont attiré l'attention de la communauté robotique. Pour programmer de façon efficace un tel algorithme, le signal de récompense envisagé doit être informatif dans le sens où il doit discriminer les valeurs des états voisins. Pour cela, des informations à priori sont souvent utilisées. Ce papier propose une méthode pour apprendre à fixer un objet sans ce type d'information. À la place, une récompense informative utilisant très peu d'information supervisée est calculée. Le calcul de la récompense est fondé sur un méchanisme de détection d'anomalies. Celui-ci estime une position d'objet pixellique avec une méthode faiblement supervisée. Cette position estimée est bruitée, ce qui rend le signal de récompense bruité également. Nous proposons une méthode d'apprentissage pour éliminer partiellement ce bruit. La fixation binoculaire est apprise dans un environnement simulé sur un ensemble d'objets aux couleurs et formes variées. La politique apprise est comparée avec une autre entraînée à partir d'une récompense lisse et informative. Nous observons des perfomances similaires, montrant qu'une étape d'encodage de l'environnement peut remplacer des informations à priori.

Mots Clef

apprentissage autonome, apprentissage par renforcement profond, détection d'anomalies semi-supervisée, fixation binoculaire, auto-encodeurs convolutifs

Abstract

Due to its ability to learn complex visuo-motor behaviors, deep reinforcement learning algorithms have attracted much interest in the robotics community. For a practical reinforcement learning implementation on a robot, it has to be provided with an informative reward signal that makes it easy to discriminate the values of nearby states. To address this issue, prior information is often assumed. This paper proposes a method to learn binocular fixations without such prior information. Instead, it uses an informative reward requiring little supervised information. The reward computation is based on an anomaly detection mechanism. This detector estimates in a weakly supervised way an object's pixellic position. This position estimate is affected by noise, which makes the reward signal noisy. We propose a learning method to partially remove the noise. The binocular fixation task is learned in a simulated environment on an object training set with various shapes and colors. The learned policy is compared with another one learned with a highly informative and noiseless reward signal. We observe similar performances, showing that the environment encoding step can replace the prior information.

Keywords

autonomous learning, deep reinforcement learning, semi-supervised anomaly detection, binocular fixations, convolutional autoencoders

1 Introduction

L'apprentissage autonome de compétences sensori-motrices revêt une importance particulière dans des espaces incertains ou complexes. Dans ce contexte, le domaine de la robotique développementale est intéressant car il réplique la capacité des enfants à apprendre des compétences sans utiliser beaucoup de supervision. Pour implémenter de tels principes, les algorithmes d'apprentissage par renforcement sont particulièrement adaptés.

Ces dernières années, les applications de l'apprentissage par renforcement ont bénéficié des avancées de l'apprentissage profond, et sont devenues plus complexes en termes de dimension de l'espace d'état. Par exemple, [1] a appris des politiques visuo-motrices sur différents jeux Atari. Des politiques visuo-motrices ont également été apprises pour plusieurs tâches de robotique de manipulation [2], [3], [4]. Enfin, de nouveaux algorithmes d'apprentissage par renforcement ont été testés sur des tâches simulées à dimensionalité élevée [5], [6], [7].

Une des limites de ces applications est l'utilisation de supervision extérieure pour la fonction de récompense. En effet, pour qu'une application utilisant de l'apprentissage par

renforcement profond soit efficace, les signaux de récompense doivent discriminer les valeurs des états voisins et ne doivent pas être trop bruités. Pour cela, des données supervisées sont souvent utilisées. Dans les jeux Atari, l'environnement donne le score qui est non disponible dans un environnement réel. Pour la robotique de manipulation, une fonction de récompense est souvent définie en se fondant sur la distance entre une position courante d'effecteur et une pose cible [2]. Cela requiert généralement un modèle géométrique et des informations sur la pose d'un objet. De plus, quand la politique voulue doit généraliser sur plusieurs poses d'objets, il est souvent fastidieux de mesurer les différentes positions d'entraînement. Le besoin de s'affranchir d'une supervision humaine trop contraignante est également exprimé dans [8]. Les auteurs ont choisi un moyen simple de définir un but dans des situations variées. Le principe est de choisir des pixels et leur position cible associée. Cette dernière constitue néanmoins une information supervisée.

Le but de ce travail est d'apprendre une tâche de fixation binoculaire en n'utilisant ni information à priori (position d'objets), ni modèle géométrique ou paramètre de caméra. Plus précisément, la tâche consiste à fixer un objet quelconque situé aléatoirement sur une table avec deux caméras. Un intérêt de l'apprentissage d'un tel comportement est de simplifier des tâches de manipulation telles que l'atteinte ou la saisie d'objets, comme l'attestent certains travaux [9], [10], [11]. En effet, leur complexité est réduite car les angles des caméras après fixation donnent une information de profondeur implicite. De plus, cette stratégie s'inspire du comportement humain : les êtres humains regardent un objet avant de s'en saisir. La politique est apprise en utilisant l'algorithme "deep deterministic policy gradient" (DDPG)[5]. Elle lie les pixels bruts et les angles des caméras aux mouvements des caméras. Pour réaliser une fonction de récompense nécessitant le moins possible de supervision, nous proposons d'utiliser un mécanisme de détection d'anomalies faiblement supervisé. Dans notre approche, l'objet est une anomalie par rapport au savoir de l'agent sur son environnement. Le but est de localiser cette anomalie dans l'image et de la guider vers le centre de l'image.

Notre approche de détection d'objets appartient à la classe des méthodes de détection d'anomalies semi-supervisées. Elle est similaire à [12] où l'on apprend à détecter des intrusions dans des réseaux ou bien des cancers du sein. Pour cela, un perceptron multi-couches est utilisé en auto-encodeur pour reconstruire des entrées renseignées comme "avec anomalies" ou "sans anomalies". Comme pour notre méthode, l'aspect intéressant est la localisation de l'anomalie. Cependant, les applications proposées sont peu complexes en termes de nombre d'états. En effet, pas plus de 50 entrées de réseaux de neurones sont utilisées dans les expériences. [13] utilise des machines à vecteurs de support pour apprendre à détecter des images aberrantes dans une base de données de chiffres. Pour chaque chiffre, une machine à vecteurs de support est entraînée pour modéliser une densité. Les images anormales sont détectées quand elles sont incompatibles avec la densité entraînée. Pareillement à [12], le problème est peu complexe puisqu'il implique 256 dimensions binaires.

Ces approches sont reconnues en tant que méthodes semi-supervisées parce qu'elles utilisent des données étiquetées pour accomplir des tâches non supervisées telles que l'apprentissage de densité et la reconstruction de données d'entrées. Par exemple, [13] utilise des noms de classe pour chaque chiffre et apprend des densités, [12] utilise les descriptifs "avec anomalies" et "sans anomalie" et apprend à reconstruire ses données d'entraînement.

Dans nos travaux, un auto-encodeur est appris au préalable sur un environnement sans objet. Ensuite, si l'on ajoute un objet à l'environnement, l'image d'erreur de reconstruction permet d'estimer la position de l'objet dans l'image. Nous choisissons de classer notre méthode comme étant "faiblement" supervisée pour souligner le fait que les informations supervisées utilisées sont à la fois aisées à produire et peu nombreuses. En effet, l'approche proposée n'utilise pas de supervision humaine à deux exceptions près. Premièrement, le centre de l'image est calculé car requis pour la fonction de récompense. Cependant, ce calcul est universel. Deuxièmement, la base de données d'entraînement de l'auto-encodeur est renseignée comme étant sans objet, et la phase d'apprentissage de la politique comporte toujours un objet dans la scène, ce qui n'est pas particulièrement fastidieux. Après l'étape de détection d'objet, la récompense est calculée en utilisant la distance entre le centre de l'image et la position pixellique estimée de l'objet.

Une autre contribution est la gestion du bruit de la récompense. En effet, du fait du bruit impulsionnel affectant la position estimée de l'objet, l'apprentissage doit s'effectuer avec des signaux de récompense très bruités, ce qui n'est pas le cas de la plupart des applications d'apprentissage par renforcement profond. [14] utilise un filtre à moyenne glissante sur le signal de récompense temporel avec des bons résultats dans la gestion des bruits Gaussiens et impulsionnels sur un problème peu complexe. Cette méthode peut être appliquée pour des problèmes de dimensionnalité élevée, mais elle filtrerait également des variations importantes de récompense dues à des mouvements de grande amplitude. Plus récemment, [15] a étudié l'influence du bruit sur la surestimation de la fonction Q-valeur (définie dans 2.1). Les auteurs ont développé une méthode pour annuler cet effet, en pénalisant dans la mise à jour de la fonction Q-valeur les actions improbables, mais cette détermination requiert des à priori. Dans notre approche, nous choisissons d'appliquer une méthode d'apprentissage pour réduire le bruit impulsionnel de la récompense. Nous montrons tout d'abord que ce bruit réduit la vitesse d'apprentissage. Ensuite, nous proposons une solution qui consiste à apprendre une fonction liant l'amplitude de mouvement à la variation de la position estimée de l'objet dans l'image. Cela peut être effectué par un algorithme de régression gé-

nérique. Cette méthode d'annulation du bruit peut être utilisée par d'autres applications affectées par un bruit impulsionnel.
La dernière caractéristique intéressante de la méthode proposée concerne la capacité de généralisation. Le couple sensori-moteur appris pour la fixation d'objets peut s'appliquer à n'importe quel objet. Pour ce faire, le système apprend la fixation sur une base de données d'objets aux formes et aux couleurs variées.
Le plan de l'article est organisé comme suit : la section 2 reprend des outils théoriques utilisés dans nos travaux et présente la méthode de calcul de la récompense faiblement supervisée. La section 3 décrit les expériences effectuées pour valider le calcul de la récompense ainsi que ses résultats. Ces derniers sont discutés dans la section 4.

2 Méthodes

2.1 Outils théoriques

Apprentissage par renforcement.
Un algorithme d'apprentissage par renforcement est usuellement appliqué sur un processus de décision Markovien $< S, A, R, T >$ où :

- S est un ensemble d'états
- A est un ensemble d'actions
- T est un modèle de transition ($T : S \times A \to S$)
- R est une fonction de récompense ($R : S \times A \times S \to \mathbb{R}$)

L'objectif d'un agent apprenant par renforcement est d'optimiser un critère fondé sur les futures récompenses. Dans ce qui suit, le critère d'optimisation est la somme dépréciée des futures récompenses $J = \sum_{k=0}^{\infty} r_k \gamma^k$ où $\gamma \in [0, 1]$ est le facteur de dépréciation et r_k la récompense à l'étape k.
Pour optimiser le critère, nous entraînons une politique déterministe $\pi : S \to A$. Par la suite, la fonction Q-valeur est utilisée, $Q_\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} r_k \gamma^k \middle| s, a\right]$.

Apprentissage par renforcement profond.
L'apprentissage par renforcement est affecté par le "fléau de la dimension"[16]. Depuis les années 2010, avec l'émergence de l'apprentissage profond et l'arrivée de meilleurs ordinateurs, les fonctions habituelles de l'apprentissage par renforcement telles que les fonctions Q-valeur, avantage, valeur, les politiques et les modèles dynamiques [1] sont de plus en plus fréquemment approximées par des réseaux de neurones profonds. Utiliser de telles structures permet de représenter des fonctions sur un espace d'entrée à dimensionnalité élevée en extrayant automatiquement une hiérarchie de descripteurs. De plus, les dimensions des espaces d'états des tâches d'apprentissage par renforcement sont de plus en élevées. La première application d'apprentissage par renforcement profond appris sur des pixels bruts et sans calculer à la main des descripteurs [18] a utilisé une combinaison d'un algorithme de type "batch Q learning" avec des auto-encodeurs profonds. Le "deep Q-Network"(DQN) [1] a battu des êtres humains experts dans beaucoup de jeux de la console Atari. Cette approche utilise un réseau de neurones convolutif pour approximer la fonction Q.

1. Pour une introduction à ces notions, le lecteur peut se référer à [17]

Deep deterministic policy gradient.
Même si le DQN est très performant dans les jeux Atari, il ne peut être appliqué sur des tâches impliquant un espace d'actions continu. Le DDPG [5] pallie cette difficulté en combinant l'algorithme "deterministic policy gradient algorithm" [19] et le DQN [1]. Il s'agit d'un algorithme acteur-critique mettant à jour le critique Q_ϕ de paramètre ϕ et la politique déterministe (l'acteur) π_θ de paramètre θ. Dans notre approche, le critique et l'acteur sont des réseaux de neurones et les paramètres représentent les poids et les biais de ces derniers. Les mises à jour s'effectuent à l'aide des équations ci-dessous :
Soit T_b un lot de transitions :
$< s_i, a_i, r_i, s_i^{'} >_{i \in [\![1\ ;\ N_\mathrm{b}]\!]} \in S \times A \times \mathbb{R} \times S$, avec N_b la taille du lot.
Les cibles du réseau de neurones Q_ϕ sont calculées en utilisant une mise à jour de type TD(0) (avec un taux d'apprentissage égal à 1) :

$$\forall i \in [\![1\ ;\ N_\mathrm{b}]\!], y_i = r_i + \gamma Q_\mathrm{T}(s_i^{'}, \pi_\theta(s_i^{'})) \qquad (1)$$

avec Q_T un réseau cible qui copie Q_ϕ toutes les K iterations. Le réseau Q_ϕ se met à jour en minimisant la fonction de coût Euclidienne L2 $= \frac{1}{2N_\mathrm{b}} \sum_{i=1}^{N_\mathrm{b}} (y_i - Q_\phi(s_i, a_i))^2$.
En utilisant Q_ϕ et le fait que la politique soit déterministe, le gradient de la politique est calculé comme suit :

$$\frac{\partial Q_\phi}{\partial \theta} \simeq \frac{1}{N_b} \sum_{i=1}^{N_b} \frac{\partial Q_\phi(s_i, \pi_\theta(s_i))}{\partial a} \frac{\partial \pi_\theta(s_i)}{\partial \theta}. \qquad (2)$$

En plus de l'algorithme, nous utilisons une procédure d'inversion de gradients [20] pour borner les actions. Cette méthode diminue le gradient quand l'action calculée par la politique approche une de ses limites (fixées arbitrairement). Si l'action dépasse sa limite, le signe du gradient est changé. Ce mécanisme limite l'amplitude des actions.

2.2 Définition de la tâche de fixation

Tâche.
La tâche de fixation d'un objet est accomplie quand l'objet est au centre des deux caméras (cf Figure 1). Mathématiquement, le problème est modélisé suivant un processus de décision Markovien où :

- $S = (I^{\mathrm{gauche}}, I^{\mathrm{droite}}, q)$, l'ensemble des états se compose de deux images RVB (I^{gauche} et I^{droite}) et des coordonnées des caméras q. Les coordonnées des caméras sont représentées par deux angles de panorama ("pan") pour chaque caméra et un angle d'inclinaison ("tilt") commun (cf Figure 2).

  Avec un détecteur d'objet parfait, la position estimée de l'objet dans l'image pourrait remplacer les images en tant qu'états. Cependant, le bruit impulsionnel affectant la détection d'objet rend cette astuce inefficace.

Apprendre un lien direct entre les images et les actions résout ce problème et évite également le besoin d'utiliser un détecteur d'objet durant l'exploitation.

- $A = \Delta q$, l'ensemble des actions comprend les trois variations de coordonnées angulaires de caméras.
- T, le modèle de transition n'a pas besoin d'être défini pour notre approche et la récompense R est décrite ci-dessous.

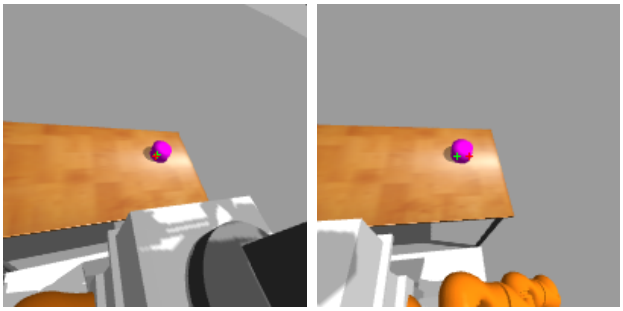

FIGURE 1 – Fixation binoculaire réalisée sur un cylindre violet, la croix rouge représente le centre de l'image, la croix verte est la position estimée de l'objet dans l'image

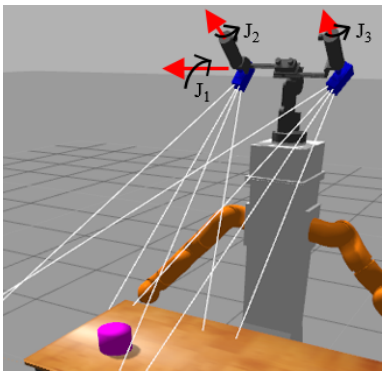

FIGURE 2 – Représentation du système de deux caméras avec ses trois degrés de liberté

Structures des réseaux de neurones.
Les structures de Q et de la politique sont représentées sur la figure 3. Elles ont été choisies pour optimiser les performances d'apprentissage (amélioration de la politique, fonction de coût de Q). Pour ces figures, les acronymes anglo-saxons FC et ReLU désignent respectivement une couche "entièrement connectée" et une fonction d'activation f définie sur $\mathbb{R}$ par $f(x) = \max(0, x)$.

2.3 Calcul de la récompense

Méthode générale.
Le calcul de la récompense implique trois étapes : une étape de pré-entraînement dans laquelle l'agent apprend à reconstruire son environnement sans objet, une étape de détection d'objets et le calcul de la récompense.

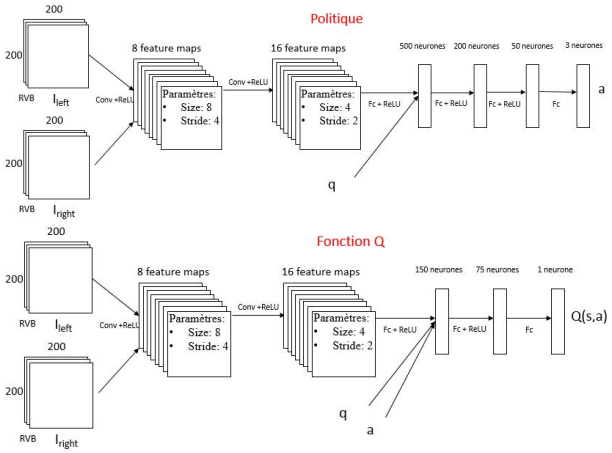

FIGURE 3 – Structure de la politique et de la fonction Q-valeur

Dans les prochains paragraphes, l'exposant c représente la caméra gauche ou droite.

Pré-entraînement
Pour chaque caméra $c = \text{gauche}$ ou droite, 10 000 configurations de caméras sont générées, ce qui produit deux bases de données de 10 000 images D^{gauche} et D^{droite}. L'ensemble des configurations de caméras se répartit en une grille régulièrement espacée entre les butées (fixées arbitraitement pour garder la table dans le champ de vision). Les images RVB 200×200 sont converties en niveaux de gris et sous-échantillonnées (50×50).
Ensuite, l'auto-encodeur A_{ψ^c} est entraîné sur D^c.

Détection de l'objet
L'étape de détection d'objets prend en compte l'image d'erreur de reconstruction. Cette méthode suppose que les objets sont mal reconstruits car les auto-encodeurs ne sont pas entraînés dessus. Ainsi, l'erreur de reconstruction est localisée à la position de l'objet dans l'image. Cela nous permet d'estimer la position pixellique de l'objet en utilisant un estimateur de densité par noyau.

FIGURE 4 – Schéma du calcul de la position de l'objet

La figure 4 montre les étapes du calcul de la récompense :

- Sous-échantillonnage et conversion en niveaux de gris $I^c_{\text{origine}} \rightarrow I^c_{\text{gr}}$
- Reconstruction en utilisant l'auto-encodeur appris $I^c_{\text{gr}} \rightarrow \hat{I^c_{\text{gr}}} = A_{\psi^c}(I^c_{\text{gr}})$
- Calcul de l'image d'erreur $|I^c - \hat{I}^c|$

- Extraction des N points $\{P(i)\}_{i\in[\![1\ ;\ N]\!]} = \{(x(i), y(i))\}_{i\in[\![1\ ;\ N]\!]}$ avec la plus grande intensité. $\{I(i)\}_{i\in[\![1\ ;\ N]\!]}$ est l'ensemble des luminances correspondantes.

  De ces points, une distribution de probabilité $\{p(i)\}_{i\in[\![1\ ;\ N]\!]}$ est calculée en utilisant un estimateur de densité par noyau. Nous utilisons un noyau Gaussien de variance unitaire : $\forall i \in [\![1\ ;\ N]\!], p(i) = \frac{1}{N}\sum_{j=1}^{N} I(j).K(P_i - P_j)$,
  avec $K(P_i - P_j) = \frac{1}{2\pi}\exp^{-0.5||P_i - P_j||_2^2}$.
  La position pixellique estimée de l'objet P_O^c se trouve là où la probabilité est maximale : $P_O^c = P_k$
  où $k = \arg\max_i(p(i))$. N est égal à 150 dans nos expériences.

Récompense
La récompense est une fonction décroissante de la distance Euclidienne $||P_{\mathrm{O}}^c - P_{\mathrm{Ic}}^c||_2$. Nous proposons d'utiliser une fonction affine à valeurs dans $[-1, 1]$ pour chaque caméra :

$$r^c = 2 \times \left(\frac{d_{\max} - ||P_{\mathrm{O}}^c - P_{\mathrm{Ic}}^c||_2}{d_{\max}} - \frac{1}{2} \right) \quad (3)$$

où d_{max} est la distance maximale entre la position pixellique de l'objet et le centre de l'image.

$$r = r^{\mathrm{gauche}} + r^{\mathrm{droite}} \quad (4)$$

Cette méthode permet d'extraire un signal de récompense dans un mode "faiblement" supervisé. De surcroît, la fonction de récompense est informative car elle permet de discriminer les valeurs de deux états voisins.

Structure de l'auto-encodeur.
La structure de l'auto-encodeur est présentée figure 5.

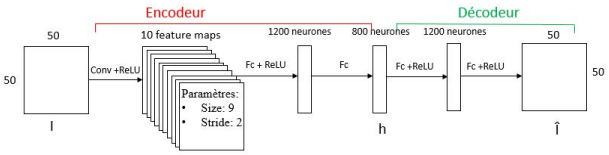

FIGURE 5 – Structure de l'auto-encodeur

Le choix de l'architecture (nombre de couches, type de couche, nombre d'unités dans la troisième couche cachée) est réalisé empiriquement en essayant d'obtenir des bonnes performances et d'assurer un apprentissage rapide.

Méthode d'annulation du bruit.
Les auto-encodeurs entraînés ne reconstruisent pas parfaitement les images sans objet dans la scène. En effet, des zones de haute fréquence telles que des pieds de table sont difficiles à reconstruire. Ainsi, celles-ci sont parfois détectées à la place des objets. Cela produit un bruit impulsionnel dans la fonction de récompense (cf Figure 6). Nous choisissons d'apprendre la fonction $\Delta \mathrm{d} = f(||\Delta q||_2)$ pour détecter une variation de détection d'objets anormale par rapport à l'amplitude du mouvement des caméras. Δd est la distance Euclidienne dans l'image entre deux positions d'objets successives. Cette fonction peut être approximée par n'importe quelle méthode de régression incluant les réseaux de neurones, les processus Gaussiens ou les machines à vecteurs de support. Ici, un processus Gaussien avec une fonction d'interaction exponentielle est utilisé et entraîné sur 1 000 transitions. Cela nous permet de supprimer les transitions dont la différence (en pixels) entre la variation réelle de détection et celle estimée est au-delà d'un certain seuil.

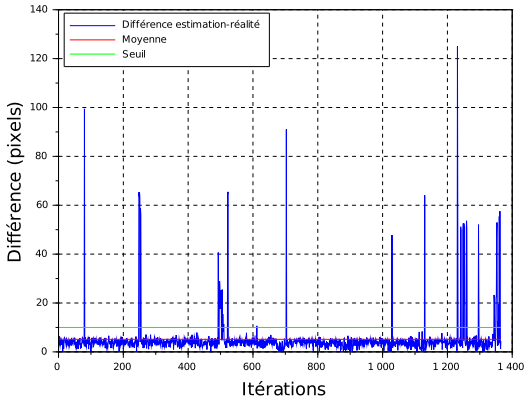

FIGURE 6 – Évolution de l'erreur pour la variation de l'estimation de la position de l'objet

La figure 6 illustre la stratégie de suppression de bruit et révèle la nature impulsionnelle du bruit. Le seuil a été fixé à 10, et supprime 3% des échantillons.

3 Résultats

3.1 Environnement expérimental

Les expériences se déroulent dans un environnement synthétique en utilisant le simulateur Gazebo et le "middleware" ROS. Un système de caméras est monté sur une plate-forme robotique (cf figure 2). Une table est placée sous les caméras et un objet y est déposé. Les expériences comprennent deux ensembles d'objets. Les ensembles d'entraînement et de test sont représentés figure 7. Afin de généraliser la compétence de fixation à de nouveaux objets, les objets de l'ensemble d'entraînement ont des formes (cylindre, sphère, parallélépipède) et couleurs variées. Chaque objet de l'ensemble de test a une nouvelle forme par rapport aux objets d'entraînement. Le but est de vérifier la capacité de généralisation du comportement appris.

3.2 Entraînement

L'algorithme 1 présente la procédure d'apprentissage de la fixation. Pour éviter de traiter des données d'entraînement corrélées, les objets et leurs positions changent régulièrement selon une loi uniforme. N. B. : Au lieu d'utiliser le processus Orstein-Uhlenbeck pour l'exploration comme

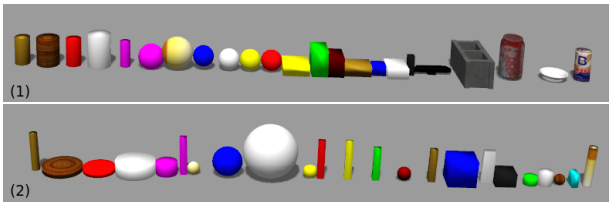

FIGURE 7 – Ensembles d'entraînement (1) et de test (2)

proposé dans [5], un bruit Gaussien d'espérance nulle et de variance constante est utilisé. Avec une fonction de récompense informative, ce type d'exploration est suffisant pour apprendre la tâche de fixation. Le nombre d'itérations de l'algorithme est réglé à 150 000 parce que la politique ne s'améliore plus après cela.

Algorithme 1 Procédure d'entraînement

Paramètres :

position initiale s_0
variance du bruit Gaussien $\epsilon = 0.02$
nombre total d'itérations $N_{\text{tot}} = 150\,000$
nombre d'itérations par épisode$N_{\text{eps}} = 35$
nombre de transitions par lot $N_{\text{b}} = 16$
seuil de supression du bruit $th = 10$
taille de la base d'entraînement du processus Gaussien $N_{\text{gp}} = 1\,000$

Entrées :

ensemble d'entraînement D_{train}

Sorties : Q_ϕ, π_θ

Etapes :

$t \leftarrow 0$
Initialiser le buffer de transitions T_{buf}
$cond \leftarrow (t < N_{\text{gp}})$ or $((|\Delta \text{d} - f(||\Delta q||_2)| < th)$ and $(t > N_{\text{gp}}))$
while $t < N_{\text{tot}}$ **do**
    Choisir un objet aléatoirement dans D_{train} et le placer au hasard
    Aller à la position initiale s_0
    $t_{\text{eps}} \leftarrow 0$
    while $t_{\text{eps}} < N_{\text{eps}}$ **do**
        Appliquer $a_t = \pi_\theta(s_t) + \mathcal{N}(0, \epsilon)$
        Observer s_{t+1}
        Calculer r_t et Δd_t
        if $cond$ **then**
            Ajouter $< s_t, a_t, r_t, s_{t+1} >$ to T_{buf}
        Tirer au hasard N_{b} transitions dans T_{buf}
        Mettre à jour Q_ϕ et π_θ avec le DDPG [5]
        if $t = N_{\text{gp}}$ **then** Entraîner $\Delta \text{d} = f(||\Delta q||_2)$
        $t \leftarrow t + 1$
        $t_{\text{eps}} \leftarrow t_{\text{eps}} + 1$
    Retirer l'objet

Pour optimiser la vitesse d'apprentissage, deux astuces sont implémentées dans la procédure d'entraînement. D'abord, nous ajoutons un terme à la récompense qui pénalise la divergence et une convergence trop importante. Ensuite, à chaque fin d'épisode, le système de caméras est placé en position initiale.

L'objectif des expériences sur l'entraînement est double. D'une part, nous voulons montrer que le bruit impulsionnel affecte l'apprentissage. D'autre part, nous voulons démontrer que l'apprentissage avec la récompense faiblement supervisée et filtrée produit des performances similaires aux entraînements avec une récompense supervisée. Celle-ci est obtenue en calculant la distance entre la projection du centre de gravité de l'objet et le centre de l'image (voir les équations (5) and (6) avec les notations des sections précédentes) :

$$r^C_{\text{sup}} = 2 \times \left(\frac{d_{\max} - ||P^C_{\text{pr}} - P^C_{\text{Ic}}||_2}{d_{\max}} - \frac{1}{2} \right), \quad (5)$$

$$r_{\text{sup}} = r^{\text{gauche}}_{\text{sup}} + r^{\text{droite}}_{\text{sup}}. \quad (6)$$

Cette fonction est affine et requiert le calcul des paramètres extrinsèques à chaque itération et une calibration avant l'apprentissage.

Pour réaliser les objectifs, les apprentissages avec les récompenses supervisées, faiblement supervisées avec ou sans suppresion de bruit sont comparés. Nous moyennons trois expériences pour chaque cas. L'erreur de fixation est relevée à chaque itération :

$$e_{\text{p}}(t) = \frac{||P^{\text{left}}_{\text{p}}(t) - P^{\text{left}}_{\text{Ic}}||_2 + ||P^{\text{right}}_{\text{p}}(t) - P^{\text{right}}_{\text{Ic}}||_2}{2}. \quad (7)$$

Elle représente la moyenne sur les images gauches et droites de la distance Euclidienne entre le centre de l'image et la projection du centre de gravité. Les résultats sont présentés figure 8 (Pour une meilleure lisibilité, les courbes sont lissées par un filtre exponentiel). L'axe vertical représente $e_{\text{p}}(t)$ et l'axe horizontal le temps. La courbe $e^{\text{s}}_{\text{p}}(t)$ représente $e_{\text{p}}(t)$ avec la récompense supervisée, $e^{\text{w}}_{\text{p}}(t)$ avec la récompense faiblement supervisée et bruitée et enfin, $e^{\text{wf}}_{\text{p}}(t)$ avec la récompense faiblement supervisée et filtrée. La courbe $e^{\text{w}}_{\text{p}}(t)$ présente les pires performances d'apprentissage. Cela indique que le bruit affecte l'apprentissage. Les courbes $e^{\text{s}}_{\text{p}}(t)$ et $e^{\text{wf}}_{\text{p}}(t)$ ont des formes similaires. Nous observons que $e^{\text{s}}_{\text{p}}(t)$ decroît un petit peu plus rapidement. Cet écart est dû au bruit résiduel présent dans les échantillons.

3.3 Test

L'objectif des tests de politique est divisé en trois sous-objectifs. Premièrement, on doit évaluer les performances des politiques apprises sur l'ensemble d'entraînement comme sur celui de test pour vérifier qu'il n'y a pas de sur-apprentissage. Deuxièmement, nous voulons comparer les politiques apprises avec la récompense bruitée et filtrée. L'idée est de montrer qu'en plus d'affecter la vitesse d'apprentissage, le bruit altère les performances de la politique

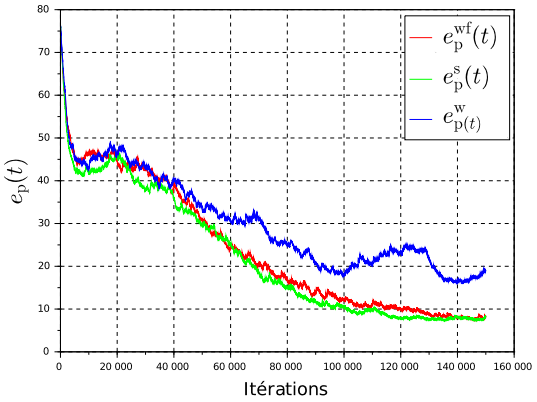

FIGURE 8 – Erreur de fixation en fonction du temps

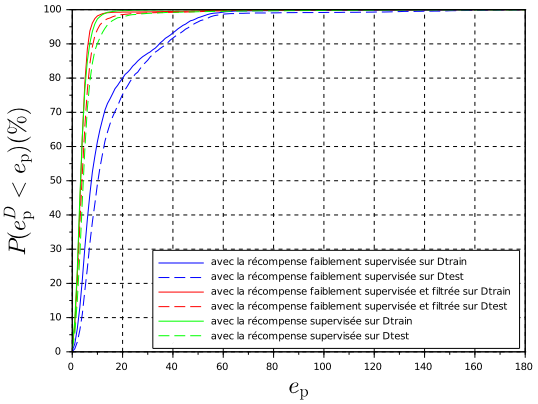

FIGURE 9 – Diagramme des fréquences cumulées de l'erreur de fixation (en %)

apprise. Enfin, nous voulons comparer la politique apprise à partir de notre récompense avec celle apprise avec un signal informatif et sans bruit. Le but est de vérifier que la récompense faiblement supervisée peut apprendre le même type de comportement qu'une récompense supervisée.

Algorithme 2 Procédure de test

Paramètres :

Le nombre total d'épisodes $N_{\mathrm{tot}} = 2\,000$
Le nombre d'itérations par épisode $N_{\mathrm{eps}} = 35$

Entrées :

La politique π_θ
L'ensemble d'objets $D = D_{\mathrm{test}}$ ou D_{train}

Sorties : L'ensemble des erreurs de fixation finales D_{p}

Etapes :

$t \leftarrow 0$
while $t < N_{\mathrm{tot}}$ **do**
    Choisir un objet aléatoirement dans D et le placer au hasard
    Aller à la position initiale s_0
    $t_{\mathrm{eps}} \leftarrow 0$
    while $t_{\mathrm{eps}} < N_{\mathrm{eps}}$ **do**
        Appliquer $a_t = \pi_\theta(s_t)$
        Observer s_{t+1}
        $t_{\mathrm{eps}} \leftarrow t_{\mathrm{eps}} + 1$
    $t \leftarrow t + 1$
    Calculer l'erreur de fixation $e_{\mathrm{p}}(t_{\mathrm{eps}})$ et l'ajouter à D_{p}
    Retirer l'objet

L'algorithme 2 décrit la procédure de test. Nous choisissons d'évaluer à chaque fin d'épisode la variable aléatoire $e_{\mathrm{p}}(t_{\mathrm{eps}})$. Les statistiques (moyenne, médiane et écart type) de cette variable aléatoire sont calculées pour 2 000 épisodes. La valeur 2 000 a été choisie pour que les statistiques soient significatives. De plus, pour chaque cas de récompense, les résultats sont moyennés sur trois politiques et représentés table 1. Les colonnes r_{w}, r_{wf} et r_{s} représentent respectivement les cas de la récompense bruitée, de la récompense filtrée et de la récompense supervisée. Enfin, un histogramme des fréquences cumulées est dressé pour caractériser plus précisément la distribution de $e_{\mathrm{p}}(t_{\mathrm{eps}})$ (cf figure 9). L'ordonnée $P(e_{\mathrm{p}}^{D} < e_{\mathrm{p}})$ représente le pourcentage des échantillons dont l'erreur de fixation est inférieure à l'abcisse. Nous observons trois faits importants. D'abord, les résultats des politiques apprises avec la récompense bruitée sont les pires. Ensuite, les résultats des politiques apprises avec les récompenses supervisée et faiblement supervisée et filtrée sont proches. Les différences observées sont dues à la variance inter-expérience et au fait que nous moyennons les résultats sur 3 politiques pour chaque cas. Enfin, les erreurs sont plus élevées sur les objets de test. Cependant, comme la différence est petite, il n'y a pas d'important sur-apprentissage.

	Entraînement			Test		
	r_{w}	r_{wf}	r_{s}	r_{w}	r_{wf}	r_{s}
$moy(e_{\mathrm{p}})$	14.11	4.95	4.80	17.10	5.92	6.81
$med(e_{\mathrm{p}})$	8.57	4.35	4.28	11.20	4.85	5.39
$ec(e_{\mathrm{p}})$	14.66	5.31	3.02	17.77	6.63	8.42

TABLE 1 – Performances des politiques apprises

4 Discussion

Nous avons présenté une approche d'apprentissage de fixation binoculaire fondée sur une combinaison d'apprentissage par renforcement profond et de détection d'anomalies qui requiert très peu de supervision. Nos expériences montrent que la politique apprise s'applique bien à de nouveaux objets. Nous avons montré que le bruit impulsionnel affecte et la vitesse d'apprentissage et la qualité de la politique résultante. Pour y faire face, nous avons proposé une méthode d'annulation de bruit qui améliore la qualité de l'apprentissage. En outre, nos expériences montrent que

l'apprentissage avec la récompense faiblement supervisée et filtrée et avec celle supervisée aboutissent à des performances similaires sur les objets d'entraînement comme sur ceux de test. Cependant, l'apprentissage est un peu plus lent pour la récompense faiblement supervisée filtrée en raison du bruit résiduel. D'un point de vue global, notre étude montre que la fonction de récompense proposée qui n'utilise vraiment que très peu de supervision permet un apprentissage efficace de la fixation binoculaire. Cela indique que les informations "à priori" habituellement utilisées par d'autres systèmes [11] pourraient être remplacées par une étape de pré-entraînement d'auto-encodeur couplée avec un mécanisme de détection d'anomalies.

La méthode de calcul de récompense comporte quelques limitations. D'une part, l'environnement ne peut varier durant l'apprentissage. Autrement, l'auto-encodeur doit être adapté durant l'apprentissage. D'autre part, la méthode est appliquée dans un cadre simple et n'a pas encore été adaptée dans un environnement encombré.

Comme futurs travaux, nous voudrions rendre les auto-encodeurs adaptatifs pour améliorer la robustesse du système. De plus, nous voudrions valider la méthode sur un robot réel. Finalement, nous souhaitons adapter notre méthode à d'autres types de tâches dans la robotique de manipulation.

Remerciements

Ce travail a bénéficié d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'avenir dans le cadre du projet LabEx IMobS3 (ANR-10-LABX-16-01) et de l'Equipement d'excellence RobotEx (ANR-10-EQPX-44), d'une aide de l'Union Européenne au titre du Programme Compétitivité Régionale et Emploi 2014-2020 (FEDER-Région Auvergne), et d'une aide de la Région Auvergne. JT reçoit le soutien de la fondation Quandt.

Références

[1] V. Mnih, K. Kavukcuoglu , D. Silver, A. Rusu Andrei, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski , S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg , and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540) :529–533, 2015.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end Training of Deep Visuomotor Policies. *J. Mach. Learn. Res.*, 17(1) :1334–1373, 2016.

[3] V. Kumar, E. Todorov, and S. Levine. Optimal control with learned local models : Application to dexterous manipulation. In *ICRA*, pages 378–383, 2016.

[4] C. Finn, X.Y.Tan, Y.Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *ICRA*, pages 512–519, 2016.

[5] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015.

[6] N. Heess, G. Wayne, D. Silver, T.P. Lillicrap, T. Erez, and Y. Tassa. Learning Continuous Control Policies by Stochastic Value Gradients. In *NIPS*, pages 2944–2952, 2015.

[7] S. Gu, T.P. Lillicrap, I. Sutskever, and S. Levine. Continuous Deep Q-Learning with Model-based Acceleration. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2829–2838, 2016.

[8] C. Finn and S. Levine. Deep Visual Foresight for Planning Robot Motion. *ArXiv*, 2016.

[9] A. Ghadirzadeh, A. Maki, and M. Björkman. A sensorimotor approach for self-learning of hand-eye coordination. In *IROS*, pages 4969–4975, 2015.

[10] L. Natale, F. Nori, G. Sandini, and G. Metta. Learning precise 3D reaching in a humanoid robot. In *ICDL*, pages 324–329, 2007.

[11] H. Hoffmann, W. Schenck, and R. Moller. Learning Visuomotor Transformations for Gaze-control and Grasping. *Biol. Cybern.*, 93(2) :119–130, 2005.

[12] S. Hawkins, H. He, G.J. Williams, and R.A. Baxter. Outlier Detection Using Replicator Neural Networks. In *DaWaK*, volume 2454 of *Lecture Notes in Computer Science*, pages 170–180, 2002.

[13] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J.C. Platt. Support Vector Method for Novelty Detection. In *NIPS*, pages 582–588, 1999.

[14] A.Moreno, J.D.Martin, E.Soria, R.Magdalena, and M.Martinez. Noisy Reinforcements in reinforcement learning : some case studies based on gridworlds. In *WSEAS*, pages 296–300, 2006.

[15] R. Fox, A. Pakman, and N. Tishby. Taming the Noise in Reinforcement Learning via Soft Updates. In *UAI*, 2016.

[16] R. E. Bellman. *Adaptive Control Processes : A Guided Tour*. MIT Press, 1961.

[17] R.S. Sutton and A.G. Barto. *Introduction to Reinforcement Learning*. MIT Press, 1998.

[18] S. Lange and M. Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *IJCNN*, pages 1–8, 2010.

[19] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic Policy Gradient Algorithms. In *ICML*, Beijing, China, 2014.

[20] M.J. Hausknecht and P. Stone. Deep Reinforcement Learning in Parameterized Action Space. *CoRR*, abs/1511.04143, 2015.