



ELIOTT JACOPIN, ANTOINE CORNUÉJOLS, CHRISTINE MARTIN,  
FARZANEH KAZEMIPOUR, CHRISTOPHE SAUSSE

Détection automatique de plantes au sein d'images aériennes de champs par  
apprentissage non supervisé et approche multi-agents

Volume 2, n° 1 (2021), p. 123-156.

[http://roia.centre-mersenne.org/item?id=ROIA\\_2021\\_\\_2\\_1\\_123\\_0](http://roia.centre-mersenne.org/item?id=ROIA_2021__2_1_123_0)

© Association pour la diffusion de la recherche francophone en intelligence artificielle  
et les auteurs, 2021, certains droits réservés.



Cet article est diffusé sous la licence  
CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE.  
<http://creativecommons.org/licenses/by/4.0/>



*La Revue Ouverte d'Intelligence Artificielle est membre du*  
*Centre Mersenne pour l'édition scientifique ouverte*  
[www.centre-mersenne.org](http://www.centre-mersenne.org)

# Détection automatique de plantes au sein d'images aériennes de champs par apprentissage non supervisé et approche multi-agents

Eliott Jacopin<sup>a</sup>, Antoine Cornuéjols<sup>a</sup>, Christine Martin<sup>a</sup>,  
Farzaneh Kazemipour<sup>b</sup>, Christophe Sausse<sup>b</sup>

<sup>a</sup> UMR MIA-Paris, AgroParisTech, INRAE, Université Paris-Saclay, 75005 Paris, France  
*E-mail* : eliott.jacopin@agroparistech.fr, antoine.cornuejols@agroparistech.fr,  
christine.martin@agroparistech.fr

<sup>b</sup> Terres Inovia, Thiverval-Grignon, France.

---

**RÉSUMÉ.** — Cet article a pour objet la détection de plantes à partir d'images aériennes capturées par drones. Une méthode efficace de détection bénéficierait autant aux agriculteurs qui souhaitent connaître l'état de leurs cultures et en prédire le rendement qu'aux agronomes pour faciliter l'acquisition de données expérimentales et en réduire le coût.

Nous proposons une méthode en deux étapes avec, d'abord, un apprentissage non supervisé, puis la mise en œuvre d'un système multi-agent faiblement paramétré. Dans un premier temps, un clustering estime la position des rangs et des plantes en s'appuyant sur des contraintes géométriques connues a priori, à savoir que les rangs de cultures sont globalement équidistants entre eux et qu'il en est de même pour les plantes au sein d'un rang. Dans un second temps, le système multi-agent raffine l'estimation fournie par l'étape de clustering et isole chaque plante.

Nous démontrons les performances de la méthode sur des tâches de comptage de plants de tournesols. Les résultats obtenus sont comparables à ceux de l'état de l'art sur les problèmes faciles et nettement supérieurs sur les problèmes difficiles. Nous présentons ensuite des résultats sur l'identification de gradients de densité dans un champ en fonction de la répartition des plantes qui ont levées.

Une contribution importante de ce travail concerne également la mise au point d'un outil de génération de champs virtuels avec le moteur de jeu *Unity*. Il devient ainsi possible de générer facilement des jeux de données réalistes correspondant à des situations diverses ce qui permet de contourner la difficulté d'obtenir des jeux de données étiquetés.

**MOTS-CLÉS.** — Apprentissage non supervisé, système multi-agents, analyse d'images à partir de drones.

---

## 1. INTRODUCTION

Compter des objets dans une image est une tâche importante en vision par ordinateur, motivée par une grande variété d'applications telles que le comptage de cellules dans des images biologiques, la surveillance du trafic routier, l'estimation du nombre de personnes dans une foule, les enquêtes écologiques ou l'inventaire des produits dans

les magasins et entrepôts. En agriculture, par exemple, les drones permettent d'enregistrer des images à moindre coût, ce qui permet un traitement souple et immédiat des images [5]. Un défi essentiel réside dans le comptage automatique des plantes dans les champs. En effet, le peuplement cultivé, estimé par la densité de plants et son hétérogénéité spatiale, est pour les agronomes un des principaux facteurs explicatifs du rendement. Il peut être utilisé pour réaliser un diagnostic a posteriori ou bien faciliter des décisions techniques en cours de culture.

Cependant, compter des objets est difficile car ils sont souvent variables en termes de forme, de taille, de pose et d'apparence et peuvent être partiellement en occlusion. En agriculture, la présence d'adventices (mauvaises herbes), la variabilité des stades de croissance et les défauts dus aux appareils de prise de vue ou aux conditions météorologiques affectent les performances.

Les méthodes existantes peuvent être catégorisées principalement en deux classes : les méthodes basées sur la détection et les méthodes de régression [30].

Dans les approches *basées sur la détection*, un classifieur, par exemple un réseau de neurones, apprend à reconnaître la présence du ou des objets d'intérêt dans une sous-image ou une fenêtre, puis cette fenêtre parcourt toute l'image afin de compter le nombre total d'objets. Cette approche présente toutefois des difficultés. Tout d'abord, elle nécessite de (très) nombreux exemples d'apprentissage étiquetés, souvent sous la forme de boîtes de délimitation dessinées à la main ou d'annotations de pixels, dont le coût d'acquisition est notoirement élevé. Ensuite, la détection des objets est elle-même une tâche difficile en raison de la variabilité de leur apparence, de la présence de bruit et d'éventuelles occlusions partielles. Outre la sélection de descripteurs pertinents, tels que des ondelettes, des shapelets, des edgelets, etc., ces méthodes demandent également le réglage fin des paramètres de l'algorithme. Enfin, le choix de la taille d'une fenêtre glissante et du processus de défilement peut être délicat.

Par contraste, les méthodes *de régression* tentent d'estimer directement le nombre d'objets d'intérêt à partir d'une caractérisation globale de l'image. Cela permet de surmonter la plupart des difficultés des méthodes basées sur la détection et, ces dernières années, ces méthodes ont défini l'état de l'art en termes de performances, notamment grâce à l'utilisation de réseaux de neurones convolutifs. Cependant, disposer de grandes bases d'images annotées ainsi que d'une expertise avancée pour former les réseaux neuronaux profonds est encore indispensable. En outre, un nouvel entraînement du système est nécessaire lorsque les objets d'intérêt changent, par exemple des plants à un autre stade de développement ou bien un autre type de plante.

Dans cet article, nous introduisons une nouvelle approche, valable lorsque les objets d'intérêt sont organisés selon des structures spatiales régulières, comme les spectateurs dans une salle de spectacle, les marchandises sur les étagères d'un magasin ou les plantes dans les champs. Cette approche fonctionne en deux phases. Tout d'abord, les relations spatiales entre les objets sont estimées. Puis, un système multi-agents (MAS) est employé afin de raffiner l'estimation et de fournir une détection exacte des objets d'intérêt. La structure spatiale estimée lors de la première phase détermine les positions

initiales des agents ainsi que la hiérarchie de contrôle et un ensemble de canaux de communication. Une couche de cette hiérarchie est composée d'agents assimilables à des classificateurs faibles qui tentent de déterminer s'ils sont positionnés sur des objets d'intérêts dans l'image. Ces agents peuvent confirmer ou infirmer leurs suppositions en communiquant entre eux. La deuxième phase est itérative jusqu'à ce que les agents soient stationnaires. Leur nombre final détermine le nombre d'objets détectés.

Les avantages de l'approche sont les suivants :

- (1) Elle *ne nécessite pas de nombreuses images d'entraînement* puisque la détermination de la structure est non supervisée et que les agents eux-mêmes sont des détecteurs simples, faiblement paramétrés.
- (2) Elle *s'adapte facilement à diverses conditions* concernant la structure, la nature des objets, leur taille et leur apparence.
- (3) Elle atteint des *performances élevées* dans les conditions expérimentales variées testées.

Ces bonnes propriétés reposent sur l'hypothèse qu'une structure régulière existe parmi les objets, comme c'est le cas dans les champs. L'approche ne devrait donc pas fonctionner sur le comptage de personnes dans une foule, ou sur le comptage de cellules par exemple. Mais lorsqu'une structure spatiale régulière existe, alors cette connaissance apporte un biais utile et puissant que notre méthode exploite.

La figure 1.1 fournit un exemple d'image aérienne d'un champ de tournesols. On peut y voir des rangées de plantes, ici à un stade de développement assez avancé, présentant un chevauchement entre les plantes, des ombres de tailles diverses et des zones peuplées d'adventices, surtout sur le côté gauche de l'image. Le comptage des plantes est alors un vrai défi.

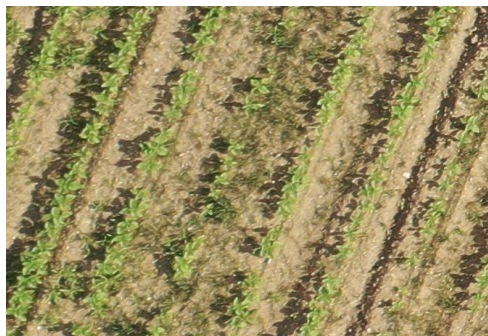


FIGURE 1.1 – Exemple d'une image aérienne d'un champ de tournesols.

L'article est structuré comme suit. La section 2 dresse un état de l'art sur le comptage d'objets dans des images. La section 3 décrit l'approche proposée. La méthode de génération d'ensembles de données synthétiques utilisés dans les expériences est présentée dans la section 4 et les résultats des expériences de comptage sont rapportés



dans la section 5 en sous-section 5.1. Une application de la méthode à la détection de gradients de rendements est pour sa part exposée en sous-section 5.2. Finalement, la section 6 tire un bilan sur la méthode et sur les résultats obtenus et propose des perspectives pour les travaux futurs.

## 2. ÉTAT DE L'ART

On peut prévoir de fortes incitations économiques à savoir compter automatiquement les plantes dans les champs à différents stades de développement. Les images aériennes des drones offrent de nouvelles possibilités pour le faire et la littérature scientifique reflète cet intérêt avec un nombre croissant de publications ces dernières années. Des préoccupations similaires existent pour le comptage automatique d'objets dans d'autres contextes, en particulier le comptage de personnes ou de véhicules.

Le comptage des objets peut se faire en passant par la *détection des objets*, ou bien à partir d'une *estimation de la densité*, généralement directement à partir d'une analyse au niveau des pixels de l'image. Dans le premier cas, la détection des objets repose soit sur une connaissance préalable de la forme des objets à compter, soit sur l'apprentissage par la machine à reconnaître des objets. Cependant, il est généralement difficile de décider quels modèles sont utiles, et l'utilisation de l'apprentissage supervisé, quant à lui, nécessite de (très) nombreuses images étiquetées et des ressources computationnelles importantes, par exemple si l'on a recours à des réseaux neuronaux profonds. Dans le second cas, l'estimation de la densité semble plus simple, mais elle nécessite toujours de grands jeux de données d'apprentissage et fournit des estimations plus grossières du nombre d'objets dans une image. Il est à noter de plus que l'estimation par densité ne permet pas de travailler sur l'hétérogénéité spatiale, alors que la détection permet de cartographier les objets ce qui est un enjeu important. Par ailleurs, les deux approches, basées sur l'objet ou sur la densité, sont sujettes à d'importantes erreurs lorsque les objets sont occlus ou se chevauchent.

Pour l'approche par comptage explicite des plantes, [4] est un exemple représentatif de l'état de l'art pour l'approche par modèle. Dans leurs expériences de comptage de plants de maïs, les auteurs ont sélectionné 4 à 12 modèles et ont utilisé une technique de corrélation croisée normalisée pour estimer le nombre de plantes. La méthode exige que des plantes représentatives dans les images soient choisies, et aucune recette n'est donnée pour cela. Une erreur de 2,2 % est obtenue lorsqu'ils utilisent 12 modèles, mais la performance chute à 25,7 % lorsque les plantes se chevauchent.

Dans l'article [24], les auteurs utilisent des réseaux de neurones profonds pour apprendre à reconnaître les plants de sorgho. Les étapes de prétraitement et de formatage nécessaires avant que l'apprentissage ne puisse avoir lieu sont plutôt complexes. De plus, une technique pour augmenter le nombre d'images d'entraînement étiquetées a du être développée. L'apprentissage lui-même nécessite entre 50 000 et 500 000 itérations, ce qui implique un coût de calcul très lourd. Un pourcentage d'erreur absolu moyen de 6,7 % est obtenu. Cependant, l'article ne précise pas si les ensembles de données utilisés comprenaient des plantes qui se chevauchaient ou non.

L'approche basée sur la densité est illustrée dans [5]. Une méthode de clustering est d'abord employée pour éliminer ce qui est supposé correspondre à des mauvaises herbes et des signaux parasites. Ensuite, des seuils sur différentes longueurs d'onde sont fixés afin de classer les pixels comme appartenant ou non à des plantes. Un réglage fin est nécessaire pour cela. Des taux d'erreur d'environ 5 % avec des écarts-types assez importants sont rapportés dans les expériences. Ici aussi, le chevauchement des plantes entraîne une détérioration significative des performances.

Il faut noter qu'il est difficile de comparer équitablement les approches proposées par les différents auteurs en raison de l'absence de jeux de données publiques et donc d'une base commune pour l'évaluation des performances. Une méthode systématique de génération de données réalistes, telle que celle que nous présentons en section 4, serait à cet égard très utile pour des comparaisons directes des approches proposées.

### 3. PRÉSENTATION DE LA MÉTHODE <sup>(1)</sup>

#### 3.1. PHASE 1 : ESTIMER LA STRUCTURE SPATIALE

Les champs de culture présentent en général une structure géométrique régulière avec des rangs parallèles et équidistants sur de vastes portions. De plus, les cultures sont souvent semées sur la base d'une densité contrôlée qui induit une distance égale entre deux plantes consécutives. Les plantes ne sont par conséquent pas aléatoirement positionnées sur les images des champs capturées par des drones aériens : elles sont alignées sur leurs rangées respectives et séparées de leurs voisines de manière à peu près équidistante. Ces observations apportent des informations précieuses pour aider à détecter, et donc à compter, les plantes sur une image capturée par un drone.

C'est pourquoi nous proposons dans cet article une méthode de détection d'objets basée sur l'identification de leur organisation géométrique au sein des images. Pour les images de champs de culture, cette organisation est caractérisée par les distances inter-rangs et inter-plantes. La première phase de notre méthode estime ces distances en commençant par (i) segmenter les zones vertes des images ; puis (ii) en réalisant une rotation des images afin que les lignes soient colinéaires avec l'axe  $Y$  ; et, enfin (iii) en appliquant une analyse par transformée de Fourier (FT) sur le signal produit en projetant les coordonnées des pixels verts sur les axes  $X$  et  $Y$ .

##### 3.1.1. *Segmentation des images*

Pour identifier les zones des images correspondant aux plantes, nous utilisons l'indice de végétation *Excess Green* ( $ExG$ ) en association avec la méthode de segmentation automatique d'Otsu [20]. Chaque pixel d'une image est remplacé par sa valeur verte amplifiée :  $ExG = 2g - r - b$ . L'association de  $ExG$  avec la méthode d'Otsu pour la segmentation des images de champs de culture a été étudiée et validée dans des travaux précédents [20, 6, 7, 21]. À la fin du processus de segmentation, les images des champs

---

<sup>(1)</sup> Le code implémentant la méthode de détection décrite dans cette section est disponible à l'adresse suivante : [https://github.com/LittleCoinCoin/Plant\\_Counting](https://github.com/LittleCoinCoin/Plant_Counting).

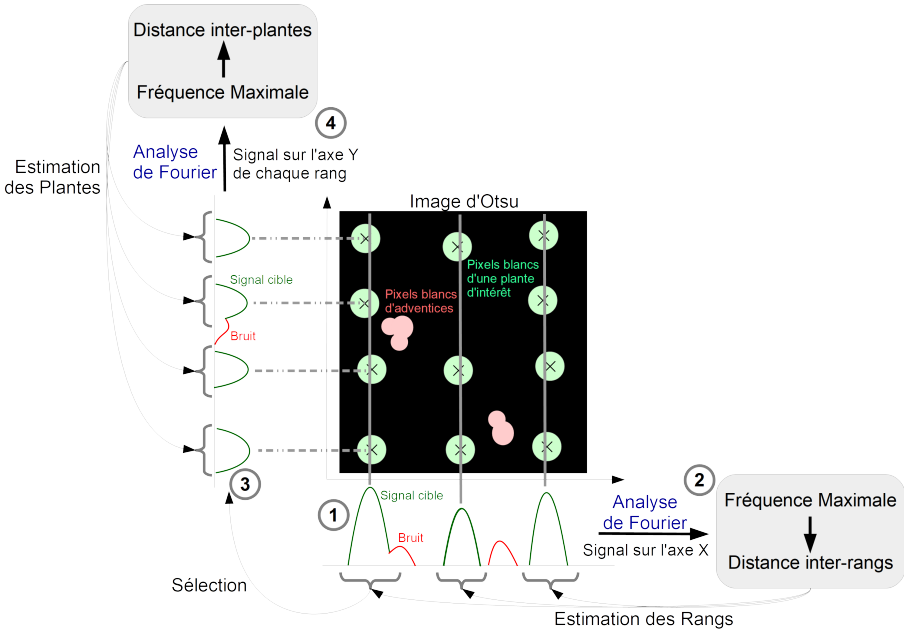


FIGURE 3.1 – Analyse de Fourier selon les axes X et Y. Le signal traité grâce à la transformée de Fourier est issu de la projection des pixels blancs des images Otsu sur les axes X et Y.

de culture RGB sont transformées en images en noir et blanc, appelées *images Otsu*, où les pixels blancs sont censés correspondre à une plante (culture ou adventice).

### 3.1.2. Rotation automatique des images

Pour faciliter l'estimation des distances entre les rangées et entre les plantes, la rotation de toutes les images des ensembles de données a été calculée de manière à ce que les rangées de cultures soient orientées le long de l'axe Y.

Pour ce faire, un score, dénoté  $\rho_\theta$ , est attribué pour chaque rotation  $\theta$  d'une image d'Otsu. Plus précisément,  $\rho_\theta$  est calculé de la manière suivante : Soit  $O_{n,m}$  une image d'Otsu avec  $n$  lignes et  $m$  colonnes ; et,  $\forall i \in \llbracket 1, n \rrbracket, \forall j \in \llbracket 1, m \rrbracket, p_{i,j}$  le pixel de la  $i$ -ème ligne et de la  $j$ -ème colonne de  $O_{n,m}$ . Alors :

$$\forall p_{i,j} \in O_{n,m}, \begin{cases} p_{i,j} = 1 & \text{si le pixel est blanc,} \\ p_{i,j} = 0 & \text{sinon le pixel est noir.} \end{cases}$$

De plus, on définit une fonction  $f$  telle que :

$$f : \mathbb{N} \mapsto \{0, 1\}$$

$$x \mapsto \begin{cases} 0 & \text{si } x = 0 \\ 1 & \text{sinon.} \end{cases}$$

Dans ce cas :

$$\rho_{\theta} = \frac{1}{m} \sum_{j=1}^m f \left( \sum_{i=1}^n p_{i,j} \right) \quad (3.1)$$

Dit autrement, ce score est le rapport entre les colonnes de l'image pivotée qui sont occupées par au moins un pixel blanc et le nombre total de colonnes (nombre de pixels le long de l'axe  $X$ ) de cette image pivotée. Il est compris entre 0 (pour une image noire) et 1 (au moins un pixel blanc par colonne). Comme les rangs des champs de culture sont généralement définis comme des lignes, la rotation  $\theta$  d'une image d'Otsu qui possède le plus de colonnes sans pixel blanc correspond à l'orientation verticale des rangées de culture. Ainsi, on cherche  $\theta$  tel que :

$$\rho_{\theta}^* = \text{ArgMin} \{ \rho_{\theta}, \theta \in [0, 180] \} \quad (3.2)$$

Cette méthode fonctionne tant que deux rangs consécutifs ne se chevauchent pas ou que les adventices ne couvrent pas tout l'espace entre eux. Si tel est le cas, on peut appliquer un filtre sur les images Otsu afin de ne garder que le squelette des rangs de culture en blanc. Cela peut être réalisé, par exemple, avec l'encodage du point milieu suggéré dans [8]. De cette façon, le chevauchement disparaît et la méthode reste utilisable.

### 3.1.3. Analyse par transformée de Fourier : estimation des distances inter-rangs et inter-plantes

Les champs de culture présentent des structures régulières caractérisées par une géométrie périodique entre les rangs ainsi qu'entre les plantes de chaque rang. Ces structures résultent des dispositifs mécaniques utilisés pour semer les graines. Dans une salle de spectacle, la périodicité est imposée par le placement des sièges. Le problème est d'identifier automatiquement les rangs et les plantes à partir d'une image (Otsu) où aucune des deux n'est étiquetée.

L'approche proposée repose sur la connaissance préalable du type de structure à attendre, puis sur la mise au point d'une procédure permettant d'extraire ce type de signal de l'entrée. Ici, des structures linéaires, correspondant à des rangs, sont attendues, avec une distance inter-rangs presque constante<sup>(2)</sup>. La procédure consiste donc à détecter d'abord les lignes, puis, grâce à une analyse de Fourier, à filtrer les lignes qui sont positionnées périodiquement et qui ont donc une forte probabilité de correspondre à des rangs de culture, et non, par exemple, à des adventices. La détection de l'orientation la plus probable des rangs a été décrite dans la section 3.1.2. Les points **1** et **3** de la figure 3.1 illustrent la manière dont un signal périodique est détecté à partir d'une image Otsu ajustée verticalement. Comme les lignes sont supposées avoir été réalignées avec l'axe  $Y$ , la périodicité des positions des lignes apparaît sur l'axe  $X$  : les pics de la distribution de densité des pixels blancs sur l'axe  $X$  reflètent les positions des lignes sur l'image (item **1**).

<sup>(2)</sup> Des travaux sont en cours pour généraliser l'approche à d'autres types de structures géométriques non linéaires.

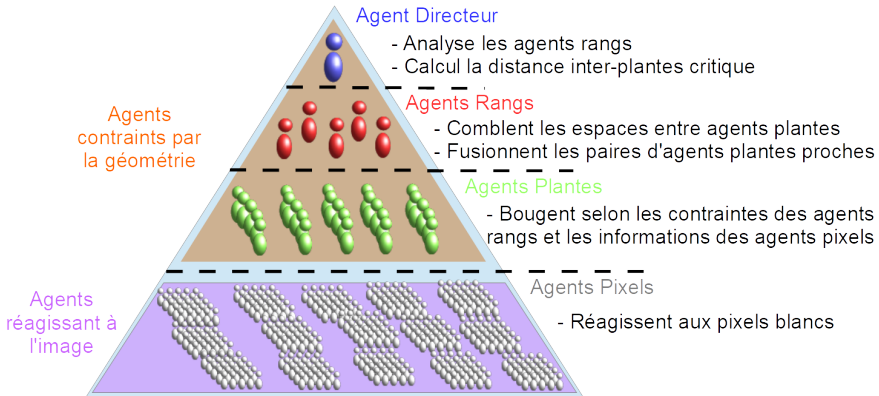


FIGURE 3.2 – Organisation hiérarchique du système multi-agents.

Plus précisément, nous calculons d'abord la distribution de densité des coordonnées  $X$  sous la forme d'un histogramme avec des cases d'une largeur de 1 pixel. Ce premier histogramme est ensuite lissé pour en obtenir un second construit avec une fenêtre coulissante de taille  $F_{\text{length}}$  pixels et se déplaçant par pas de  $F_{\text{step}}$  pixels. Cette étape permet de filtrer le bruit dans le signal et de mieux estimer les positions des images. Le choix des valeurs des paramètres  $F_{\text{length}}$  et  $F_{\text{step}}$  est un compromis afin d'éliminer le bruit présent sur le premier histogramme sans trop amputer l'amplitude du signal original. Dans notre cas, nous devons éviter que des pixels du rang  $i$  soient comptabilisés dans la fenêtre avec les pixels du rang  $i + 1$ . Il est donc recommandé que  $F_{\text{length}}$  soit inférieure au nombre de pixels généralement présent entre deux rangs sur une image. Nous avons donc choisi  $F_{\text{length}} = 20$  pixels car cette valeur satisfaisait largement la contrainte sur toutes nos images. La valeur de  $F_{\text{step}}$  doit être au maximum égale à  $F_{\text{length}}$  pour ne pas perdre d'information. Nous avons utilisé  $F_{\text{step}} = 5$  dans les expériences qui sont rapportées ici. La distance entre les rangs est calculée par une analyse de Fourier sur la distribution de densité et en gardant la fréquence maximale ainsi trouvée. Une fois la période identifiée, une recherche des positions des maxima locaux de la distribution de la densité est entreprise, ce qui donne la position estimée des rangs le long de l'axe  $X$ .

Une deuxième étape vise à estimer la distance entre les plantes. Cette estimation est effectuée comme précédemment, sauf qu'il s'agit maintenant des projections sur l'axe  $Y$  des pixels blancs attribués à chaque rang qui sont pris en compte (éléments 3 et 4). La distance entre les plantes est ensuite estimée à l'aide des projections sur l'axe  $Y$  des pixels blancs attribués à chaque ligne (éléments 3 et 4).

### 3.2. PHASE 2 : COMPTER AVEC UN SYSTÈME MULTI-AGENTS

Tout comme dans [10] pour le cas de la télédétection, nous préconisons l'utilisation d'un système multi-agents (MAS) qui tire parti des connaissances acquises sur la géométrie de l'image. Dans le cadre de la tâche de comptage des plantes, nous avons identifié quatre types d'agents organisés de manière hiérarchique (voir figure 3.2). La

structure pyramidale proposée est basée sur la hiérarchie des structures dans un champ : le champ, les rangées et les plantes, tandis que l'analyse des images s'opère au niveau des pixels. De cette analyse s'ensuit la hiérarchie des agents : un *Agent directeur* (DA) au sommet de la pyramide afin de contrôler l'ensemble du système, des *Agents de rangée* (RAs), les *Agents plantes* (PA) et enfin les *Agents pixels* (PXA).

Chaque agent d'une couche agit seul ou reçoit des ordres d'un agent de la couche supérieure : il n'y a pas de communication entre les agents d'une même couche. Les environnements dans lesquels les agents agissent sont les images d'Otsu ajustées verticalement.

### 3.2.1. *L'agent directeur (DA)*

Le DA initialise ou détruit les RAs selon les prédictions faites à l'aide de l'analyse de Fourier (voir 3.1.3). Il décide également de l'arrêt de la simulation. Parce que le DA a accès à toutes les informations fournies par les agents des niveaux inférieurs, c'est lui aussi qui calcule la *distance inter-plantes critique* (IPCD) (voir ci-dessous).

CONTRÔLE DES AGENTS RANGS (RAs). — Au début de la simulation, le DA analyse les lignes détectées par l'analyse de Fourier pour tenter d'exclure les faux positifs : des rangs qui ne sont faits que d'adventices. Pour ce faire, une procédure spéciale est mise au point, basée sur le fait que ces rangées seront placées entre de véritables RAs (rangées composées de plantes). Les rangs faux positifs sont des cas aberrants en regard de la distance inter-rangs et nous les détectons donc grâce à un test statistique d'écart à la moyenne avec un seuil de rejet  $\pi$ .

CALCUL DE LA DISTANCE CRITIQUE INTER-PLANTES (IPCD). — La plupart des décisions des agents dépendent de l'IPCD. Elle est calculée comme la distance inter-plantes la plus représentée dans l'ensemble des distances inter-plantes observées.

### 3.2.2. *Les agents pixel (PXAs)*

Les PXAs travaillent directement sur les pixels des images d'Otsu et sont instanciés par un PA. Ils sont dits *actifs* s'ils sont positionnés sur un pixel blanc. La position d'un PXA est déterminée relativement à la position du PA dont il dépend. Dans nos simulations, les PXAs sont uniformément répartis dans un carré centré sur la position de leur PA et de demi-côté  $\chi$ .

### 3.2.3. *Les agents plantes (PAs)*

Les PAs sont les agents les plus importants pour le comptage des plantes, leur nombre à la fin de la simulation déterminant en effet le nombre de plantes détectées. Chaque PA a sous sa supervision un groupe de PXAs uniformément répartis dans un carré de côté  $\chi$  centré sur la position du PA et dont le rôle est de guider le PA vers les parties les plus blanches d'une image d'Otsu (c'est-à-dire de les guider vers les zones

segmentées qui correspondent aux plantes). Ainsi, à l'étape  $i + 1$  de la simulation, un PA se déplace sur le point moyen défini par tous ses PXAs actifs de l'étape  $i$  :

$$(PA_x^{i+1}, PA_y^{i+1}) = \left( \frac{1}{n} \sum_{PXA \in \mathcal{A}} PXA_x^i, \frac{1}{n} \sum_{PXA \in \mathcal{A}} PXA_y^i \right) \quad (3.3)$$

avec  $\mathcal{A}$  l'ensemble des PXAs actifs. Les  $x$  et  $y$  sont les coordonnées des positions des agents.

Un PA peut décider de diminuer ou d'augmenter sa zone de détection en éliminant ou en initialisant de nouveaux PXAs en bordure de zone. Dans nos simulations, un PA décide de diminuer ou d'accroître sa taille en fonction du nombre de PXAs qu'il supervise qui sont actifs. L'existence d'un nombre important de PXAs actifs suggère indirectement que le PA se trouve sur une plante qui est plus grande que lui. Cela sera d'autant plus vrai que l'on s'approche des 100 % de PXAs actifs. *A contrario*, si peu de PXAs sont actifs, cela suggère que la PA est bien plus grand que la plante potentiellement détectée. Afin d'apporter un peu de souplesse (surtout dans l'étape de croissance), nous avons fixé à 80 % le seuil d'activité des PXAs qui déclenche la croissance d'un PA ; et à 20 % le seuil d'activité pour l'atrophie. Dans notre système, la croissance ou la diminution de la taille du PA revient, respectivement, à incrémenter ou à décrémenter le paramètre  $\chi$  qui représente le demi-côté du carré dans lequel se trouvent les PXAs. Enfin, si deux PAs sont respectivement inclus dans les surfaces de l'images couverte par leurs PXAs alors ces deux PAs sont fusionnés.

### 3.2.4. Les agents rangs

Les RAs sont instanciés par le DA en fonction des lignes détectées par l'analyse de Fourier (Fig. 3.1, item 2). Chaque RA initialise autant de PAs que de plantes détectées par l'analyse de Fourier (Fig. 3.1, item 4). Comme l'analyse de Fourier peut rater des plantes aux extrémités des rangs détectés, les RAs initialisent des PAs supplémentaires uniformément espacés de  $1.1\nu$  fois l'IPCD ;  $\nu$  étant le facteur de fusion des PAs (voir le paragraphe suivant). À chaque étape de la simulation, les RAs éliminent les PAs qui sont situés dans les zones noires de l'image Otsu, c'est-à-dire les PAs ayant une proportion de PXAs actifs inférieure au seuil  $\delta$  choisi par l'utilisateur.

**AJOUTER OU FUSIONNER DES PAs.** — Un RA peut décider que la distance entre deux PAs consécutifs est soit trop grande, soit trop petite. Dans le premier cas, le RA va combler les espaces vides en ajoutant de nouveaux PAs. Dans le second cas, le RA va fusionner les deux PAs concernés selon l'heuristique suivante :

$$Decision = \begin{cases} \text{Ajoute} & \text{si } |PA_y^{i+1} - PA_y^i| > \mu \text{ IPCD} \\ \text{Fusionne} & \text{si } |PA_y^{i+1} - PA_y^i| < \nu \text{ IPCD} \end{cases} \quad (3.4)$$

avec respectivement  $\mu$  et  $\nu$  comme facteurs de remplissage et de fusion.

**RESTREINDRE LES MOUVEMENTS DES PAs.** — Dans un champ, les rangs présentent généralement une forme linéaire, alignée sur l'axe  $Y$  lors du réglage des images (Section

3.1.2). Les plantes qui font partie du même rang sont donc censées être alignées. En conséquence, un RA peut restreindre les déplacements des PAs qu'il supervise afin de les maintenir aussi alignés que possible. Dans notre système, un RA comptabilise le nombre de PAs qui ont bougé vers la gauche ou la droite du graphe dans lequel les rangs sont verticaux. Le groupe minoritaire est arbitrairement ramené sur la position  $X$  moyenne du groupe majoritaire.

---

**Algorithm 1:** Simulation

---

**Input:**  $\tau, \chi, \mu, \nu, \delta, \pi$

- 1 initialize DA, RAs, PAs( $\chi$ ), PXAs
- /\* Sec. 3.2.1 \*/
- 2 AnalyseRangs( $\pi$ )
- 3 CalculIPCD()
- 4 CombleExtrémitésRangs( $\nu$ , IPCD)
- 5 StopSimu  $\leftarrow$  False
- 6 RE\_Eval  $\leftarrow$  False
- 7  $i \leftarrow 1$
- 8 **while**  $i \leq \tau$  & StopSimu = False **do**
- /\* Sec. 3.2.3 \*/
- 9 BougeVersPointMoyen()
- /\* Sec. 3.2.4 \*/
- 10 RestreintMouvementPAsSurX()
- 11 AjouteOuFusionnePAs( $\mu, \nu$ , IPCD)
- /\* Sec. 3.2.3 \*/
- 12 ModifieTaillePAs()
- /\* Sec. 3.2.4 \*/
- 13 DétruitPAsPeuActifs( $\delta$ )
- 14 **if**  $Nb\_PAs_i - Nb\_PAs_{i-1} = 0$  **then**
- if** RE\_Eval = False **then**
- CalculIPCD()
- RE\_Eval  $\leftarrow$  True
- else**
- StopSimu  $\leftarrow$  True
- end**
- 15 **end**
- RE\_Eval  $\leftarrow$  False
- 16 **end**
- 17  $i \leftarrow i+1$
- 18 **end**

---



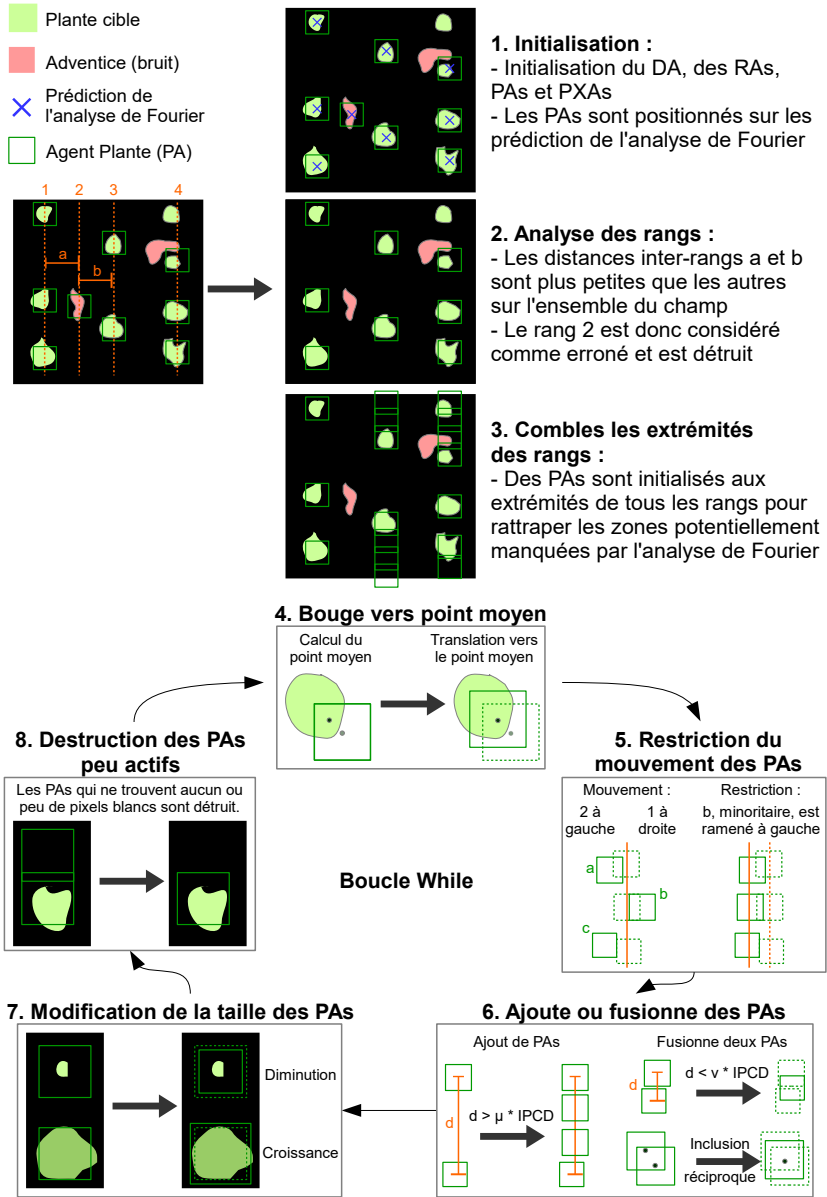


FIGURE 3.3 – Représentation des étapes de l’algorithme 1

### 3.2.5. L'exécution d'une simulation

La simulation consiste en une séquence d'actions que les agents exécutent dans un ordre déterministe répétées au maximum  $\tau$  fois (Algo. 1). Le décompte final des plantes se produit lorsque le nombre de PAS reste constant même après re-calcul de l'IPCD. La figure 3.3 fournit une représentation schématique des étapes de la simulation.

## 4. UTILISATION DE DONNÉES SYNTHÉTIQUES

### 4.1. GÉNÉRER DES IMAGES

Entraîner un algorithme à compter automatiquement des objets d'intérêt nécessite de grands jeux de données avec au moins des centaines d'images, et donc au minimum des milliers d'objets, chacun d'eux devant être étiqueté. Dans le cas du comptage des plantes, même si la capture d'images de champs à partir de drones se démocratise, il n'existe pas encore à notre connaissance de jeux de données publiques et standardisés. En outre, chaque article proposant une nouvelle méthode de comptage mesure l'efficacité de cette dernière sur les images d'un champ différent d'une méthode concurrente. Ainsi, même si les conditions de culture du champs sont souvent fournis dans le corps de l'article, la variabilité annuelle des conditions biotiques et abiotiques entrave la production de jeux de données semblables. Il y a donc un manque de données d'entraînement étiquetées et, plus globalement, un problème de reproductibilité des expériences de détection des plantes d'intérêt.

Nous proposons ici une solution à ces problèmes grâce à l'utilisation d'un moteur de création d'environnement virtuel pour générer des champs de culture artificiels. Ces environnements sont en effet aujourd'hui capables de générer des images photoréalistes très convaincantes. De plus, puisqu'il s'agit d'un environnement virtuel, l'utilisateur exerce un contrôle complet sur son contenu. L'étiquetage des objets est ainsi automatique et il est facile d'accéder à toutes les informations des objets d'intérêt (nombre, position, métadonnées, etc.). Ces données issues d'environnements virtuels sont appelées « *synthetic data* » en anglais que nous traduirons directement par « *données synthétiques* » dans le reste de cette étude. Les données synthétiques ont notamment été utilisées avec succès pour combler le manque de données réelles nécessaires à l'entraînement d'algorithmes d'apprentissage supervisé pour résoudre des problèmes de vision par ordinateur [25, 27, 9, 18].

Un certain nombre de questions se posent lorsque l'on a recours à des données synthétiques pour entraîner un système adaptatif, et encore plus lorsqu'il s'agit, en les utilisant, d'opérer des choix importants sur la méthode. Savoir générer des données synthétiques représentatives des données réelles, c'est-à-dire partageant les caractéristiques sur lesquelles s'appuie le système adaptatif reste un sujet d'étude qui, dans de nombreux domaines, n'est pas trivial. Dans notre cas, celui de la reconnaissance de plants de tournesols à partir d'images de drones, il faut savoir générer des images de tournesols, donc leurs caractéristiques de forme et de signature RGB à tous les stades de développement et de même pour les adventices. En revanche, il n'est pas nécessaire de chercher à reproduire leur texture puisque celle-ci n'intervient pas dans la méthode

développée. Il faut s'assurer que la granularité des images correspond bien à celle d'images prises à partir de drones. On peut également chercher à varier les caractéristiques du sol et, pour une étude en cours, faire varier aussi la géométrie des rangs de tournesols. Les questions sur le degré de réalisme nécessaire et sur le type de caractéristiques utilisées par la méthode ont guidé nos choix pour la génération d'images synthétiques. À la fin, la méthode a été de toute façon validée sur des images réelles.

#### 4.2. DESCRIPTION DES DONNÉES GÉNÉRÉES

Nous avons choisi d'utiliser le moteur de jeu Unity [28]. Unity s'est en effet largement imposé dans l'industrie du jeu vidéo comme un moteur complet et simple d'utilisation. D'autres industries ou domaines de recherches comme l'intelligence artificielle peuvent également désormais profiter de ce moteur grâce à des modules dédiés, par exemple pour l'apprentissage automatique dans des agents autonomes [14] ou pour la vision par ordinateur [29, 12]. Nous avons utilisé le module *Unity Perception* [29] qui a été spécialement développé pour la production de données synthétiques.

Des études antérieures sur le comptage automatique des plantes à partir d'images de drones ont montré que la difficulté de la tâche augmente avec la proportion de chevauchement des cultures [4]. Or, le chevauchement entre plantes voisines au sein d'un même rang s'accroît au fur et à mesure que les plantes grandissent. Les méthodes de l'état de l'art ont donc davantage de difficultés à compter précisément le nombre de plantes sur une image lorsqu'elles sont à un stade de développement avancé.

Afin de tester la méthode proposée, nous avons généré trois jeux de données représentant des champs de tournesols à trois stades de développement. Nous nommons ces stades selon la nomenclature utilisée par *Terres Innovia* : le stade A2 correspond aux cotylédons étalés, B3/B4 à la seconde paire de feuilles et E1 au stade bouton floral.

Dans le stade A2 (premier ensemble de données), les plantes sont séparées les unes des autres ce qui est représentatif d'un stade de développement précoce. Dans le stade B3/B4 (deuxième ensemble de données), elles se chevauchent pour certaines feuilles mais pas pour toutes. Enfin, dans le stade E1 (troisième ensemble de données), il y a chevauchement entre les plantes. L'ensemble de données (A2) est considéré comme facile, (B3/B4) comme intermédiaire et (E1) comme difficile. Mise à part la nature des modèles 3D des plantes, les paramètres utilisés pour générer les champs sont similaires pour les trois jeux de données. Chaque champ de culture a été généré avec une distance entre les rangées de 70 cm et une distance entre les plantes de 20 cm avec une variabilité de 5 %. Cela donne une moyenne cible de 7 plantes/m<sup>2</sup> qui correspond aux champs de tournesols typiques. La probabilité de croissance des plantes a été fixée à 0,9. Le seuil de bruit Perlin utilisé pour générer les surfaces où poussent les mauvaises herbes a été fixé à 0,73, tandis que la probabilité de croissance des mauvaises herbes a été fixée à 0,5. Dans chacun de ces ensembles de données, 100 champs de culture ont été générés, et quatre images ont été prises pour chacun d'entre eux. Ainsi, chaque ensemble de données contient 400 images, soit 1200 images au total. Pour prendre des photos des champs virtuels, nous avons simulé un court plan de vol de drone qui couvre le coin inférieur gauche du champ en se déplaçant une fois le long de la hauteur

et de la largeur du champ (voir les chiffres bleus sur la Fig. A.3). Nous avons configuré le mouvement du drone simulé pour qu'il recouvre l'image de 50 % sur la hauteur et la largeur, comme c'est habituellement le cas avec les images des drones.

## 5. EXPÉRIENCES ET RÉSULTATS

### 5.1. RÉSULTATS SUR LE RATIO DE COMPTAGE

La méthode proposée consiste en deux étapes : d'abord la détection et l'estimation de la structure spatiale, et ensuite, à partir de cette structure, l'identification des objets.

L'objectif des expériences menées est triple. *D'abord*, évaluer les performances de la première phase seule dans le comptage des plantes, *ensuite*, mesurer la valeur ajoutée de la deuxième phase sur la base d'une approche multi-agents, et, *enfin*, examiner le gain de performance, le cas échéant, lorsque des parties d'un champ sont couvertes par des passages multiples du drone et qu'une redondance des informations s'ensuit (voir la zone Z4 dans la figure A.3 pour un exemple).

Nous calculons donc le ratio de comptage selon :

$$\text{Ratio de Comptage} = \frac{\text{compte total des plantes prédites}}{\text{compte total des plantes réelles}} \quad (5.1)$$

Pour le SMA, un PA est compté comme une plante prédite.

Dans ce qui suit, nous comparons les performances de l'analyse de Fourier seule (Section 5.1.1), avec l'approche multi-agent à partir d'une seule image (Section 5.1.2), et, finalement, avec une technique qui prend en compte le fait que plusieurs images (jusqu'à quatre) peuvent couvrir une zone donnée (Section 5.1.3).

#### 5.1.1. Détection de la structure spatiale et comptage des plantes

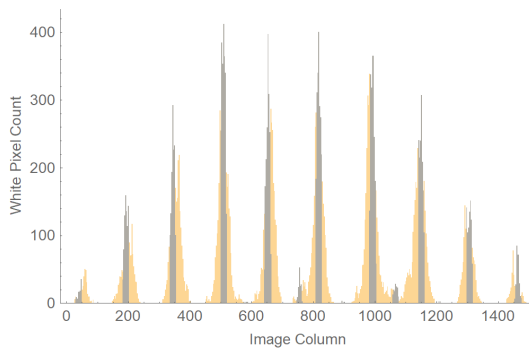


FIGURE 5.1 – Exemple de détection de lignes grâce à l'analyse de Fourier. L'histogramme en jaune résulte de la projection des pixels blancs d'une image Otsu sur l'axe X. Les parties bleues de l'histogramme sont les lignes détectées.

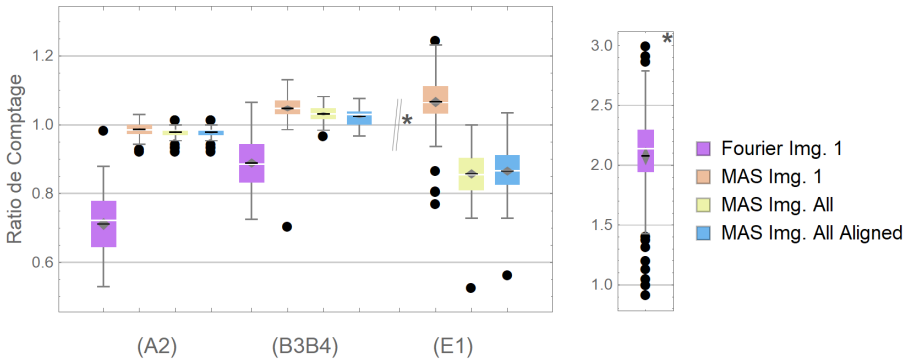


FIGURE 5.2 – Résultats sur le ratio de comptage (RC). Les couleurs des boîtes à moustaches indiquent la méthode utilisée pour compter le nombre de plantes. Avec *Fourier Img. 1* nous avons compté les plantes avec l’analyse de Fourier sur une image pour chacun des 100 champs de l’ensemble de données. Les mêmes images ont été utilisées avec *MAS Img. 1* qui compte les plantes en utilisant le système multi-agents à partir de l’initialisation avec les résultats de l’analyse de Fourier. *MAS Img. All* et *MAS Img. All Aligned* sont des méthodes qui exploitent la redondance lorsque plusieurs images couvrent la même zone dans un champ. Les points noirs représentent les valeurs aberrantes. Les limites inférieures et supérieures des boîtes indiquent respectivement le centile 0,25<sup>e</sup> et 0,75<sup>e</sup>. La médiane est représentée sur chaque boîte par un trait blanc, tandis que la moyenne est représentée par un trait noir. Le losange gris représente l’intervalle de confiance. Quand les losanges ne se chevauchent pas entre paires de boîtes, l’hypothèse nulle que les moyennes sont égales peut être rejetée en utilisant un t-test à deux échantillons.

Comme expliqué dans la section 3.1, l’analyse de Fourier est utilisée pour approcher la structure spatiale d’une image. L’algorithme cherche d’abord à découvrir les rangs et ensuite à localiser les plantes en leur sein. Cela repose sur l’analyse de la distribution de densité de la projection des pixels blancs d’une image Otsu sur l’axe  $X$  ou  $Y$  (la figure 5.1 montre une telle distribution de densité (en jaune) ainsi que les pics détectés (en bleu)).

Les plus grands pics correspondent effectivement aux rangées, mais les adventices peuvent également produire des pics, quoique plus petits.

Les résultats obtenus par l’analyse de Fourier sont résumés dans le tableau 5.2 à la ligne *Fourier* tandis que la figure 5.2 fournit des détails sur la distribution du ratio de comptage (les boîtes violettes indiquent les résultats de l’analyse de Fourier).

Le ratio de comptage obtenu avec l’analyse de Fourier seule est plutôt mauvaise avec une sous-estimation conséquente sur (A2) et (B3B4) (−29 % et −11 % en moyenne

respectivement) et une sur-estimation très importante sur (E1) (+108 % en moyenne). Pourquoi en est-il ainsi ?

- (1) Pour le jeu de données (A2), les plantes sont bien séparées, mais cela implique également que les pics de l'histogramme utilisé par l'analyse de Fourier sont relativement étroits. C'est pourquoi si un pic est légèrement en dehors d'une position prédite par l'analyse, il peut être entièrement manqué par celle-ci. Cela peut avoir pour conséquence d'ignorer des rangs existants ou des plantes à l'intérieur d'un rangs.
- (2) Pour les ensembles de données (E1), le chevauchement des feuilles entre les plantes induit un très grand bruit. Ce dernier conduit l'analyse de Fourier dédiée à l'identification des plantes, sur l'axe des  $Y$  à trouver une fréquence bien supérieure à la cible réelle : il en résulte une surestimation du nombre de plantes.
- (3) Le jeu de données (B3/B4) correspond à un problème de difficulté plus faible que les deux autres, et on observe effectivement un ratio de comptage meilleur, de l'ordre de 11 %.

L'analyse de Fourier n'est cependant utilisée que pour estimer les relations spatiales entre les plantes dans les champs de culture et son ratio de comptage n'est pas un facteur clé.

### *5.1.2. Effets de l'analyse multi-agents*

L'étape multi-agent initialise les PAs en utilisant les prédictions faites par l'analyse de Fourier, puis laisse les PAs évoluer et converger vers les plantes présumées. La question est de savoir si l'étape reposant sur le SMA améliore les performances de comptage.

Dans nos expériences sur le comptage des plantes, nous avons effectué les simulations avec les valeurs des paramètres suivantes :  $\tau = 50$ ,  $\chi = 20$ ,  $\mu = 1,5$ ,  $\nu = 0,5$ ,  $\delta = 0,01$  et  $\pi = 0,0001$ .

La valeur de  $\tau$  a été fixée comme limite supérieure du nombre de pas de la simulation qui n'a jamais été atteinte dans nos expériences.  $\chi$  est le demi-côté du carré dans lequel les PXAs d'un PA scannent l'image à la recherche des plantes d'intérêt. Cette valeur est donc dépendante de la taille des plantes et doit être adapté selon les cas. C'est effectivement le seul paramètre qui nécessite une rapide calibration de la part de l'utilisateur. Nous avons choisi une valeur de 20 pixels en estimant visuellement si les résultats sur une image paraissent bon. Les valeurs  $\mu$  et  $\nu$  ont été choisies pour des raisons géométriques.  $\nu$  est le facteur de fusion des PAs ; une valeur de 0,5 signifie que deux PAs parfaitement positionnés sur des plantes consécutives absorberont un PAs mal positionné entre eux, ce qui est souhaitable.  $\mu$  est le facteur de remplissage ; si deux PAs sont parfaitement positionnés sur des plantes mais qu'une autre plante a été oubliée entre les deux, une valeur de 2 devrait permettre sa détection. Cependant, une valeur de 1,5 s'est avérée meilleure lors des tests. Abaisser les valeurs de  $\delta$  et

Étape	Paramètres	Valeur	Section	Description
Pré-Traitements	$\theta$	1°	3.1.2	Pas de l'angle de rotation pour le redressement automatique des images
Analyse de Fourier	$F_{\text{length}}$	20 pixels	3.1.3	Taille de la fenêtre pour lisser le signal
	$F_{\text{step}}$	5 pixels	3.1.3	Pas de la fenêtre lors du lissage du signal
Système Multi-Agents	$\tau$	50	3.2.5	Nombre de répétitions maximal de la simulation
	$\chi$	20 pixels	3.2.3	Taille du demi-côté du carré de PXAs d'un PA
	$\mu$	0.5	3.2.4	Facteur de remplissage
	$\nu$	1.5	3.2.4	Facteur de fusion
	$\delta$	1 %	3.2.4	Seuil de d'activité pour la destruction d'un PA
	$\pi$	0.0001	3.2.1	Seuil de rejet statistique pour la détection des faux rangs
Post-Traitements	$\gamma$	3 PAs	5.1.3	Nombre de votes nécessaire pour accepter un groupe de PAs provenant de 4 analyses

TABLE 5.1 – Récapitulatifs des paramètres et de leurs valeurs.

$\pi$  conduira la simulation à surestimer le nombre de plantes alors que les augmenter conduira à une sous-estimation. Ces valeurs ont été optimisées en testant à plusieurs reprises le système sur des ensembles de données synthétiques d'entraînement. Les résultats rapportés ont été obtenus sur des ensembles de données de test différents de ceux de l'entraînement.

Comme le montrent la figure 5.2 et le tableau 5.2, les résultats indiquent que la phase multi-agents améliore considérablement les performances de comptage. Effectivement, pour les trois niveaux de difficulté, la performance de comptage est plus proche de 1 que pour l'analyse de Fourier seule. En particulier, le SMA ne réalise que 1 % de sous estimation en moyenne sur (A2) contre 29 % pour l'analyse de Fourier. Et les résultats sur (B3/B4) et (E1) sont aussi bien meilleurs pour le SMA avec 5 % d'erreurs contre 11 % pour Fourier sur (B3/B4); et 7 % avec le SMA contre 108 % pour Fourier sur (E1). Dans les trois cas, les intervalles de confiances sont aussi plus étroits en passant de 0,09 à 0,02, 0,07 à 0,05 et 0,39 à 0,08 respectivement. De tels gains en performance valident l'approche en deux étapes : d'abord l'approximation d'une structure, puis l'utilisation d'un SMA pour affiner le comptage. Ces gains indiquent également que l'approximation de la structure géométrique sous-jacente du champ par l'analyse de Fourier n'a pas besoin d'être très bonne pour que le SMA produisent

de bonnes estimations de comptage. Cette approximation se basant sur la structure spatiale des images n'en reste pas moins essentielle.

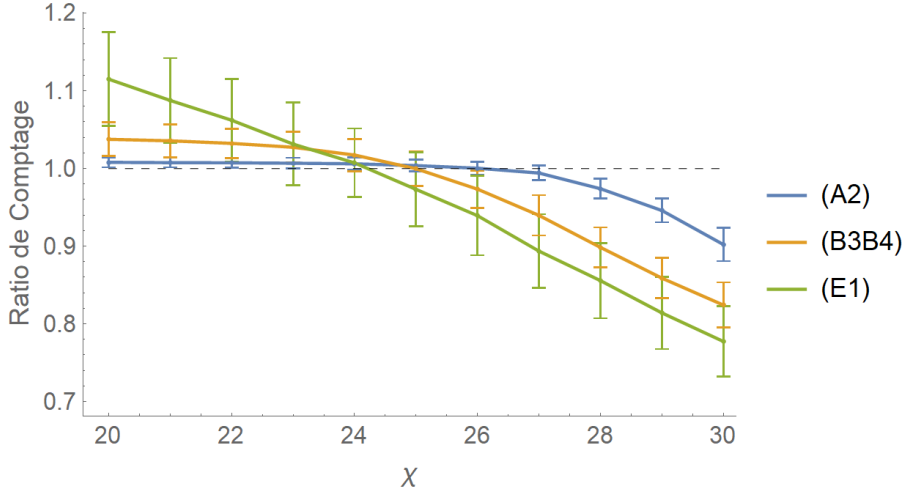


FIGURE 5.3 – Sensibilité du ratio de comptage à la taille initiale des agents plantes (paramètre  $\chi$ )

Le paramètre  $\chi$ , la taille en pixels du demi-côté du carré initial d'un agent plante (PA), est un facteur important du système multi-agents. Nous avons donc réalisé une étude de sensibilité sur les résultats en fonction de sa valeur. La figure 5.3 en récapitule les résultats pour les trois stades de développement des plantes : A2 (plantes séparées les unes des autres), B3B4 (chevauchement de certaines feuilles entre plantes) et E1 (chevauchement entre les plantes). Chaque courbe décrit le ratio de comptage par la méthode sur le nombre réel de plantes en fonction de la taille  $\chi$ . Deux conclusions principales peuvent en être tirées. D'abord, *les trois courbes sont décroissantes*, passant d'une situation de surévaluation du nombre de plantes à une situation de sous-évaluation. Cette tendance commune aux trois courbes, mais d'autant plus nette que le chevauchement entre plantes est important, peut s'expliquer. Le système multi-agents détruit en effet des PAs si ses PXAs sont globalement inactifs (voir section 3.2.3); et le système fusionne deux PAs s'ils sont respectivement inclus dans leur zone de détection (voir section 3.2.3). Un RA peut aussi fusionner deux PAs mais cela n'a pas d'influence ici. Ainsi, plus le stade de croissance des plantes est avancé et plus les PXAs seront globalement actifs, et donc, moins les PAs seront détruits. Dans ce cas, lorsque  $\chi$  est faible (*e.g.* 20), il n'y aura pas assez de fusions par proximité des PAs ce qui mène à une sur-estimation finale du nombre de plantes. *A contrario*, lorsque  $\chi$  est plus élevé (*e.g.* 30), les PAs seront trop fusionnés ce qui mène à une sous-estimation globale. Cela explique que les tendances soient similaires pour les trois courbes mais qu'elles n'aient pas les mêmes amplitudes de variation. Ensuite, *il existe des valeurs de  $\chi$ , ici  $\chi = 24$  et  $\chi = 25$  pour lesquelles les trois courbes sont très proches de 1*. Cela signifie que pour ces valeurs de  $\chi$ , quelque soit le stade de développement des



plantes, la performance de comptage est très bonne. On remarquera que les résultats rapportés dans cet article, obtenues pour  $\chi = 20$  et très bons, sont donc sous-optimaux par rapport à ceux qui pourraient être obtenus pour  $\chi = 24$  ou  $\chi = 25$ , en particulier pour le stade de développement E1.

Toutefois, on peut imaginer aller plus loin. En effet, en raison du chevauchement de 50 % sur la hauteur et la largeur des images, la plupart des zones d'un champ de culture sont couvertes par plusieurs images différentes (jusqu'à quatre fois dans l'exemple de la figure A.3). Est-il possible que même ces bons résultats, meilleurs que ceux de l'état de l'art, puissent encore être améliorés en recourant à la redondance ainsi offerte ?

### 5.1.3. *Ce qu'apporte le chevauchement entre images*

Une pratique courante lors de l'acquisition d'images de champs de culture consiste à laisser des images consécutives se chevaucher. L'une des principales motivations de cette pratique est d'éviter que les plantes situées sur les bords d'une image ne soient que partiellement visibles, et donc ignorées. Une autre motivation est l'espoir que les erreurs commises sur une image puissent être compensées sur une autre image qui couvre partiellement la même zone. Dans notre cas, les ensembles de données synthétiques ont été construits avec un chevauchement de 50 % sur la hauteur et la largeur des images. À titre d'illustration, dans notre exemple, il existe une zone (par exemple,  $Z4$ ) qui est couverte par les quatre images.

Afin de combiner les informations provenant de l'analyse de  $N$  images (par exemple  $N = 4$ ) pour une zone donnée, nous avons utilisé une méthode itérative. Son but est de former des groupes de PAs qui peuvent être de taille  $n \in [1, N]$ . Le regroupement des PAs est initialisé à partir de l'une des  $N$  analyses puis de nouveaux PAs sont ajoutés aux groupes en itérant sur les  $N - 1$  analyses restantes. Lors de l'itération, un PA est ajouté à un groupe si sa distance à ce groupe est inférieure à  $v \frac{1}{N} \sum_{j=1}^N \text{IPCD}_j$  (la moyenne des IPCD pondérée par le facteur de fusion); sinon, il crée lui-même un groupe. Si plusieurs PAs provenant d'une même analyse sont assez proches pour être ajoutés à un même groupe, alors seul le plus proche y est effectivement rattaché. Par conséquent, il ne peut y avoir qu'un PA de chaque analyse au maximum par groupe. Dans notre cas avec quatre images, nous formons donc des groupes d'au plus quatre PAs. On considère qu'un groupe correspond à une plante s'il est constitué d'au moins  $\gamma = 3$  PAs (c'est un vote majoritaire strict).

Les résultats de la combinaison des informations provenant des quatre images sont présentés sous le nom de *MAS Img. All* dans le tableau 5.2 et la figure 5.2.

Une autre variante de cet algorithme (appelée *MAS Img. All Aligned*) a été introduite. *MAS Img. All Aligned* se différencie de *MAS Img. All* par l'ajout d'un pré-traitement visant à aligner les  $N$  analyses. Pour cela, une analyse  $\mathcal{A}_i$  avec  $i \in [1, N]$  est arbitrairement choisie comme référence. Puis, les  $N - 1$  analyses restantes (dénotées  $\mathcal{A}_j$  avec  $j \in [1, N]$  et  $j \neq i$ ) sont individuellement alignées sur  $\mathcal{A}_i$ .

Les résultats présentés dans le tableau 5.2 et dans la figure 5.2 montrent que la combinaison des informations provenant de l'analyse de plusieurs images permet

Jeux de Données	(A2)	(B3/B4)	(E1)
Fourier Img. 1	0.71 (0.09)	0.89 (0.07)	2.08 (0.39)
MAS Img. 1	0.99 (0.02)	1.05 (0.05)	1.07 (0.08)
MAS Img. All	0.98 (0.02)	1.03 (0.02)	0.86 (0.07)
MAS Img. All Aligned	0.98 (0.02)	1.02 (0.02)	0.87 (0.07)

TABLE 5.2 – Résultats des scores moyens sur les trois jeux de données. L'écart-type est entre parenthèses. Les valeurs ont été arrondies au deuxième chiffre.

d'améliorer la ratio du comptage pour le seul jeu de données (B3/B4). La sur estimation est réduite de 2 % en moyenne et l'écart-type diminue également en passant de 0.05 à 0.02. Pour (A2), la stratégie de regroupement commet 1 % d'erreur supplémentaire et pour (E1) on passe de 7 % de sur estimation à 13-14 % de sous-estimation en moyenne. La baisse de performance est peut-être négligeable pour (A2) mais elle est remarquable pour (E1) : le taux d'erreur augmente et la nature de l'erreur change (d'une sur-estimation vers une sous-estimation). L'explication est que lorsque les plantes occupent une surface importante, comme c'est le cas dans le stade (E1), les PAs qui les détectent dans les différentes images peuvent avoir leurs centres trop éloignés pour qu'ils soient considérés comme faisant partie d'un même groupe, et si un groupe n'a pas trois PAs au moins, il n'est pas compté comme une plante.

Ces résultats sur (E1) contrastent avec ceux présentés dans une étude précédente [11]. Nous avons généré trois jeux de données similaires à (A2), (B3/B4) et (E1) (dénommés (S), (B) et (O) respectivement) mais pour lesquels un modèle 3D de maïs sans modèle de croissance était utilisé. Dans ce cas, le regroupement des PAs provenant de différentes images avait donné de bons résultats. La méthode de regroupement reste donc à améliorer pour s'appliquer à un plus large spectre de plantes.

#### 5.1.4. Application à des images réelles

Nous avons également appliqué la méthode à un sous-ensemble d'un jeu de données d'images réelles de champs de culture fourni par *Terres Inovia*.

Au total, le jeu de données contient 2111 images non étiquetées dont nous en avons extrait au hasard 50 qui ont été étiquetées manuellement et utilisées pour tester notre méthode. La figure 1.1 est une image extraite de ce jeu de données. En comparaison avec les jeux de données synthétiques, les tournesols se trouvent entre les stades (A2) et (B3/B4). De plus, les images mélangent des zones où les plantes sont bien séparées et des zones où les feuilles d'une plante se chevauchent avec celles de ses voisines dans la même rangée. Les images ont également été capturées par un drone à une hauteur de 30 m (contre 10 m pour les données synthétiques) et de nombreuses adventices sont présentes ce qui rend parfois difficile, même pour un humain, l'identification visuelle des tournesols. On peut donc supposer que les 50 images sont globalement comparables à celles des jeux de données synthétiques.

Notre méthode a donné un ratio de comptage moyen de 1,03 pour un écart-type de 0,12 sur le sous-ensemble de 50 images. Ces résultats sont au moins aussi bons que ceux rapportés dans l'état de l'art (voir section 2). De plus, ils sont dans l'intervalle des résultats obtenus sur les ensemble de données synthétiques (A2) et (B3/B4), même si l'écart-type est plus important.

Cela confirme que l'utilisation de données synthétiques pour régler la méthode que nous proposons est une procédure prometteuse qui conduit effectivement à de bons résultats sur des données réelles.

## 5.2. RÉSULTATS POUR LA DÉTECTION DE VARIATIONS DE RENDEMENTS

Une estimation précise de la densité de plants est utile dans deux domaines. En premier lieu, elle est importante pour le diagnostic agronomique, en particulier pour les cultures dont le rendement dépend en grande partie du nombre de plants (maïs, tournesol, betteraves...). Elle facilite l'interprétation de cartes de biomasse et de rendement et elle peut aussi intéresser les agronomes et les expérimentateurs qui cherchent à expliciter le lien entre peuplement et rendement et dépendent de méthodes de comptage au sol coûteuses et limitées. En second lieu, une utilisation est aussi envisageable pour l'aide à la décision pendant la culture pour assurer l'efficacité des intrants tels que l'eau et les engrais, et pour maximiser la productivité, la qualité et le rendement des cultures.

### 5.2.1. *La méthode proposée*

Les techniques classiques reposent sur des drones équipés de capteurs hyperspectraux, multispectraux ou thermiques et sur le calcul subséquent de l'indice de végétation qui décrit la densité relative et la santé de la culture. Cependant, ces techniques requièrent un calibrage précis pour établir le lien entre la réflectance mesurée et l'état réel des plants. Ce calibrage à son tour nécessite que des mesures au sol aient été effectuées de manière fiable, précise et représentative de l'état de la parcelle étudiée. De nombreux travaux montrent que ces conditions ne sont pas évidentes à satisfaire [3, 13, 26, 17].

L'approche proposée dans notre travail, basée sur le comptage des plants, est beaucoup plus précise que des techniques indirectes utilisant l'indice de réflectance et une estimation de densité. De plus, l'entraînement du système s'opérant à partir de données de synthèse, le calibrage est beaucoup plus aisé et beaucoup plus fin et il est possible quelque soit la condition des cultures que le praticien souhaite simuler.

### 5.2.2. *Le protocole expérimental*

En raison de l'absence de bases de données publiques bien étalonnées portant sur les variations de densité dans les champs, les expériences réalisées l'ont été sur des images virtuelles de champs. Cela permet de contrôler les conditions de variations de densité et de comparer les résultats obtenus avec la « vérité » terrain qui est contrôlée. Nous avons

ainsi généré des jeux de données avec une probabilité de levée des plantes qui dépend de la hauteur et de la largeur d'un champ ainsi que de leurs positions dans le champs.

Dans nos expériences, un gradient linéaire artificiel est défini de sorte que le couvert végétal de la parcelle soit le plus faible en bas à gauche (aux coordonnées (0,0) du champ) et maximal en haut à droite (aux coordonnées (14,14)). Nous avons échantillonné quatre intensités de gradient avec des variations allant de 0 % à 100 % (0-100), 25 % à 100 % (25-100), 50 % à 100 % (50-100) et 75 % à 100 % (75-100). En théorie, moins la variation est importante, plus la tâche de détection du gradient est difficile car il est nécessaire d'être très précis durant la détection pour bien rendre compte des variations de densité de faible ampleur. Toutefois, pour notre méthode, la variation la plus difficile à mesurer est la plus large (0-100). En effet, notre méthode se base sur la détection de la géométrie sous-jacente du champ (distances inter-rangs et inter-plantes régulière) pour détecter les plantes. Or, moins il y a de plantes, et plus cette géométrie est difficile à détecter.

Les trois stades de développement (A2, B3/B4 et E1) ont été utilisés. Pour ces douze modalités ( $4 \times 3$ ), 100 champs ont été générés et, pour limiter le nombre d'images à acquérir, des parcelles de 14 m de côté ont été définies, chacune pouvant être couverte par 6 images contiguës. Il y a donc 1 800 images par intensité de gradient et  $4 \times 1\,800 = 7\,200$  images au total.

Nous analysons par la suite deux types de résultats : (i) des profils de densité et (ii) la densité à l'échelle du champ complet. Nous appelons « profil » une section de la densité du champ complet réalisée sur l'axe de variation de densité la plus importante du champ. Dans notre cas, la variation de densité est la plus forte entre le coin en bas à gauche et le coin en haut à droite de l'image des champs. Les profils sont donc établis le long de la diagonale correspondante. Chaque profil est constitué de 101 points (mesure 0 en bas à gauche et mesure 100 en haut à droite).

### 5.2.3. Résultats expérimentaux sur la détection de variation de densité

Les densités sont obtenues en comptant les plantes dans des mailles carrées contiguës sur les parcelles puis ce décompte est interpolé par une spline cubique pour obtenir une mesure continue du décompte. Cette approche de reconstruction est intéressante car elle permet aux praticiens d'analyser des variations à différentes échelles. Plus la taille de maille est fine et plus l'hétérogénéité de la couverture végétale du champ peut être mesurée. Toutefois, si la maille est trop petite, le comptage va être très dépendant de la position de la maille ce qui va introduire un bruit fort dans les estimations. *Contrairement*, si la maille est trop grande alors on calcule la densité moyenne sur le champ, sans plus pouvoir détecter des variations. Il existe donc des bornes minimale et maximale au-delà desquelles le résultat ne serait plus pertinent. La borne minimale peut être déterminée en mesurant la taille de la maille en-deçà de laquelle un changement de positionnement modifie de manière significative le comptage dans chaque maille individuellement. Pour nos expériences, nous avons utilisé une maille de  $1\text{ m}^2$  afin d'être certain de compter des plantes aux endroits normalement peuplés puisque nous avons une distance inter-rangs de 70 cm en moyenne.

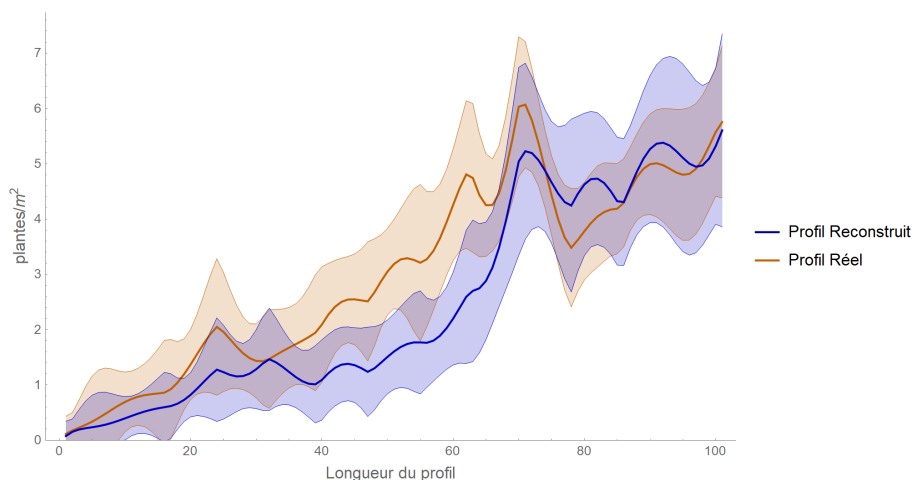


FIGURE 5.4 – Profils de densité moyen d’un champ virtuel créé dans le logiciel Unity avec un gradient 0-100 au stade de développement E1. En orange, le profil de la densité réelle de plantes. En bleu, le profil de la densité reconstruite par notre méthode de comptage de plantes. La combinaison d’un gradient 0-100 et d’un stade de développement E1 est le cas le plus difficile présent dans les jeux de données que nous avons généré. Les lignes continues correspondent au moyennes sur les 100 champs et les surfaces transparentes à leurs écarts-types respectifs. La figure A.5 en annexe compare les profils des autres couples (gradient - stade de développement).

La figure 5.4 permet de comparer le profil obtenu à partir de la disposition réelle moyenne des plantes dans 100 champs (en orange) avec le profil obtenu à partir de la disposition détectée moyenne dans ces même 100 champs (en bleu). La figure présente les résultats pour le couple de conditions (gradient / stade de développement) le plus difficile que nous avons généré. La détection des variations de densité serait parfaite si la courbe bleue était superposée à la courbe orange. Dans le cas le plus difficile présenté, la courbe bleue, quoique non directement superposée à la courbe orange, en suit tout de même assez fidèlement les variations. Même dans ce cas difficile, la méthode reproduit correctement les variations locales de la densité de plantes. Et les résultats sont bien meilleurs, avec des cas de superposition presque parfaits, pour les autres couples de conditions (gradient / stade de développement) (figure A.5 en Annexe).

Pour compléter cette étude impliquant des gradients linéaires, nous avons aussi testé notre méthode lorsque les variations de densité sont non linéaires. La figure 5.5 montre les résultats pour deux champs. À gauche, les densités obtenues à partir des positions réelles de plantes et à droite les densités obtenues à partir des positions des plantes détectées. La similarité est évidente, ce qui montre la puissance de notre

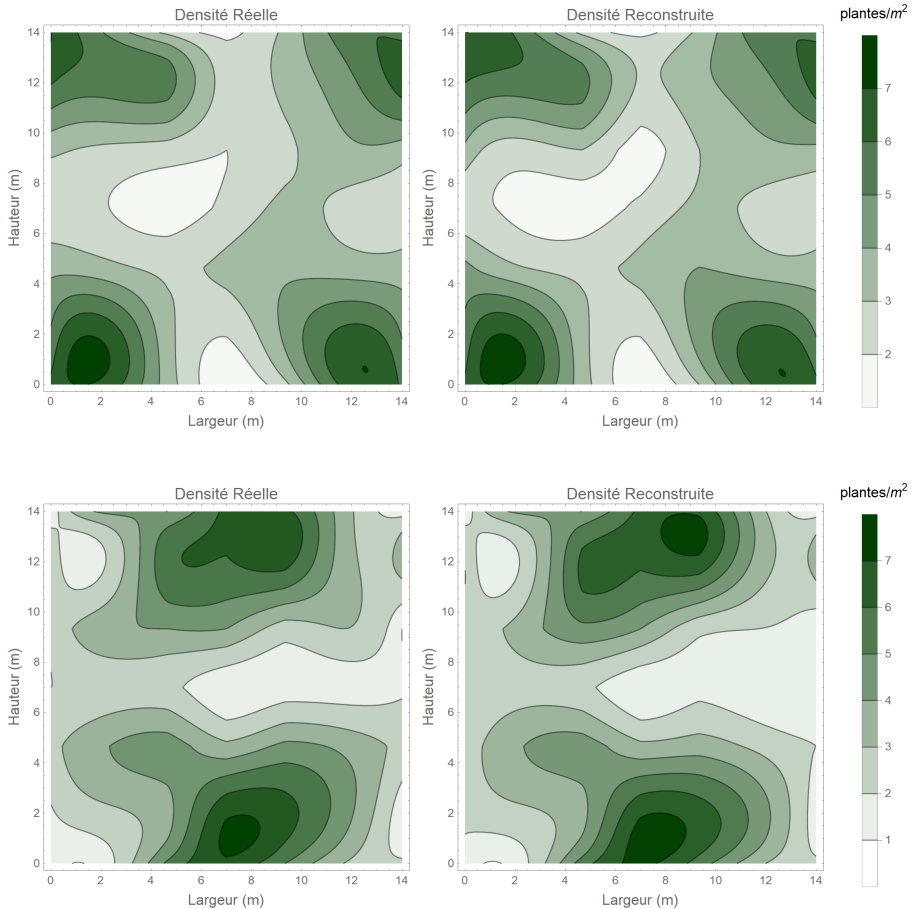


FIGURE 5.5 – Densité de plantes pour deux champs virtuels différents. À gauche, la densité réelle de plantes sur un champ au stade B3/B4 créé dans le logiciel Unity. À droite, la densité reconstruite par notre méthode de comptage de plantes.

méthode fondée sur le comptage des plantes. Les variations de densité sont bien reproduites et, ici, il en est de même des valeurs exactes des densités reconstruites car les niveaux de verts sont définis sur la même échelle.

## 6. CONCLUSIONS

Avec la généralisation des appareils de prise de vue, il est de plus en plus essentiel de développer des systèmes d'analyse d'image fiables et interprétables [19]. Ce document a introduit une nouvelle méthode pour compter les objets tout en satisfaisant à ces contraintes. Elle est applicable lorsque les objets sont organisés dans l'espace selon

un schéma régulier. La méthode détecte d'abord le motif spatial et l'utilise ensuite pour initialiser des agents dans un système multi-agents. La méthode est simple et ne nécessite pas de réglage fin des paramètres, ni de définition délicate des modèles ou d'apprentissage coûteux. En fait, elle ne requiert que des ressources informatiques très modestes. De plus, le déroulement des calculs par la méthode : détection de la structure spatiale, puis les étapes successives de calcul du système multi-agents peuvent facilement être visualisées et ce qui se passe analysé, ce qui permet de modifier facilement des paramètres du système, et modifier ou ajouter de nouvelles fonctions. C'est un avantage notable par rapport à l'emploi de méthodes opaques comme les réseaux de neurones.

Au cours d'une série d'expériences approfondies sur des ensembles de données contrôlées et des images aériennes réelles de champs de culture, la méthode a permis d'obtenir des performances comparables aux meilleures performances reportées dans la littérature scientifique lorsque les objets sont bien séparés et a dépassé les meilleures performances connues lorsque les objets se chevauchent.

Les travaux futurs porteront sur l'incorporation de modèles de croissance des plantes dans le générateur de données synthétiques afin de pouvoir mener des expériences plus approfondies sur le comptage des plantes, de la levée à la récolte. D'une part, nous pensons que la méthode présentée dans cet article se prêtera naturellement à des comptages successifs de la croissance des plantes dans le même champ et à l'estimation de la croissance individuelle des plantes. En effet, les agents existant à la fin d'un processus de comptage à un moment donné peuvent être utilisés comme graines pour le processus de comptage à l'étape suivante (par exemple sur des images du même champ prises un mois plus tard). D'autre part, cela contribuerait à la mise en place d'un référentiel d'images réalistes des champs de culture, permettant ainsi de faire des comparaisons systématiques entre les différentes méthodes de comptage des plantes.

Une autre extension des travaux rapportés ici concerne le comptage d'objets liés par des structures géométriques non linéaires, comme dans des champs en montagne ou bien en raison de mouvements spécifiques des engins de semences. En dehors du problème du comptage des plantes, nous pensons que la méthodologie présentée dans cet article peut être appliquée à d'autres problèmes de détection d'objets sur les images, à condition qu'une structure spatiale régisse les positions de l'objet cible. Nous prévoyons de vérifier cette affirmation en utilisant la méthode sur d'autres environnements spatialement structurés tels que le comptage de personnes dans les stades ou les salles de spectacle.

## ANNEXES

### DÉTAILS SUR LA GÉNÉRATION DES IMAGES DE CHAMPS

Unity est un moteur de jeu basé sur la notion de *gameobjects* (GO) auxquelles sont attachés des *components* (CP). Les GOs se différencient alors en fonction des combinaisons de CPs qu'on leur attache. Il existe de nombreux CPs pré-intégrés dans Unity qui permettent de paramétrer la physique (comme la gravité), les éclairages, les textures, l'intelligence artificielle, etc. Il est également possible d'attacher aux GOs des scripts codés en C# pour définir des fonctionnalités « sur mesure » lorsque l'on souhaite des comportements plus spécifiques que ceux offerts par les CPs pré-intégrés.

Dans notre cas, le générateur de champ de culture est un script C# attaché à un seul GO dans l'environnement virtuel (appelé une *scène*). Le script prend en entrée un ensemble de paramètres qui agissent sur des caractéristiques de la scène comme le soleil, le drone, la taille du champ, les distances inter-rangs et inter-plantes, etc. Certains de ces paramètres sont tirés aléatoirement dans des intervalles de valeurs définis par l'utilisateur afin de pouvoir générer un large éventail de champs de culture. Cette manière de générer des données synthétiques est appelée *domain randomization* [27]. La scène contient également un GO avec un CP appelé *scenario* issue du module *Unity Perception* qui contrôle l'acquisition des images et l'étiquetage automatique. La génération des champs est détaillée ci-dessous en commençant par le choix d'un modèle pour les plantes d'intérêt.

#### Générer des plantes

Plusieurs solutions dites de « *génération procédurale* » [23] existent pour générer des modèles réalistes 3D de plantes automatiquement et à moindre coût. Nous avons porté notre choix sur une implémentation du *Lindenmayer System* (L-System) [16] initialement développée dans [2] en C++ avec une interface en python appelée L-Py. Plus exactement, nous avons utilisé le portage de L-Py produit par [15] pour le logiciel Blender [1]. Ce portage nécessitait la compilation de L-Py sous forme d'une librairie dynamique pour qu'il soit utilisable par Blender et seule une librairie pour la version UNIX de Blender existant [15], nous avons réalisé le support pour les machines Windows. Grâce à cela nous sommes en mesure, dans Blender, d'écrire un script qui simule la phylloxaxie des plantes d'intérêts.

La figure A.1 présente des exemples de modèles 3D de plantes simulant la phylloxaxie du tournesol (voir Listing A.4 en Annexe pour le script). Les stades de croissances présentés sont les stades A2, B3/B4 et E1. De profil, on remarque que les espaces inter-nœuds sont en théorie trop grands si l'on compare à une plante de tournesol réelle. Mais, puisque nous allons simuler l'acquisition d'images par un drone aérien, cela ne posera finalement pas de problèmes car la ressemblance en vue de dessus est suffisante. Il aurait aussi été possible de fournir un travail artistique plus important afin d'améliorer le réalisme des textures des plantes 3D afin de ne pas biaiser l'étape de segmentation d'Otsu. En pratique, les images segmentées de champs virtuels avec ces modèles 3D de plantes se sont révélées très proches des images segmentées de champs



réels. Nous n'avons donc pas jugé nécessaire d'améliorer davantage l'aspect visuel des modèles 3D.

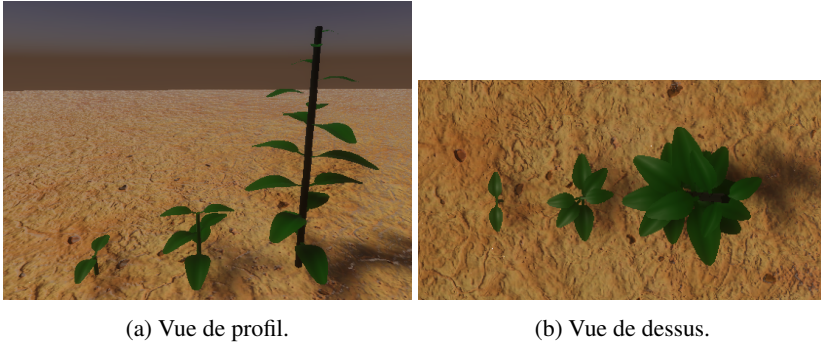


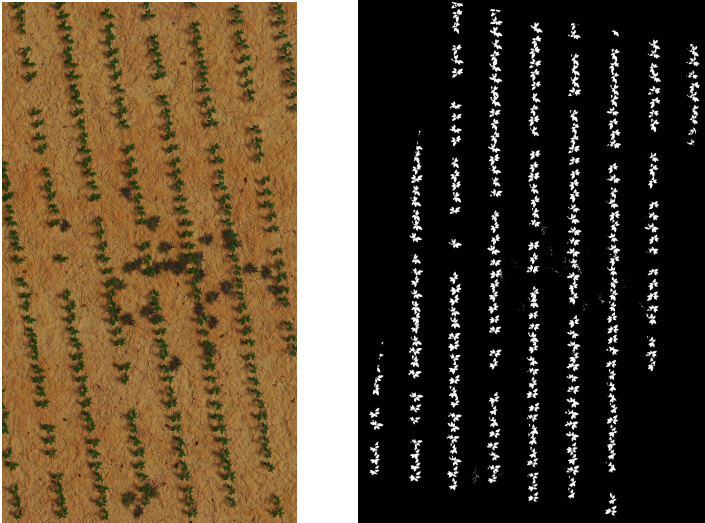
FIGURE A.1 – Modèles 3D de tournesols générés avec L-Py sous Blender puis importés dans Unity. Trois stades de développement sont montrés avec, de gauche à droite, le modèle simulant les stades A2, B3/B4 et E1.

### *Générer les champs*

Les champs sont générés en supposant une machine agricole équipée d'un semoir dont les caractéristiques sont spécifiées dans les paramètres du CP : nombre de rangs, distance inter-rangs et densité de graines en semis (simulé par la distance inter-plantes). Il est également possible de simuler la pousse ou la mort de la plante via une probabilité de croissance qui peut-être uniforme sur le champs ou dépendante de sa hauteur et de sa largeur. Cela permet de créer des champs avec des couvertures végétales variables et de simuler, par exemple, une croissance faible des plantes d'un côté du champ et une croissance maximale de l'autre.

Afin d'introduire du bruit dans les images pour notre problème de détection des plantes, nous avons également laissé la possibilité d'ajouter des adventices dans le champ. Nous n'avons pas envisagé de positions particulières pour les adventices. En revanche, nous avons observé qu'elles avaient tendance à être regroupées plutôt qu'isolées dans les champs réels. C'est pourquoi nous avons utilisé le bruit de Perlin [22] pour générer aléatoirement des espaces sur le champ de culture où les adventices seraient présentes.

La figure A.2a donne un exemple d'une image RGB d'un champ virtuel avec des tournesols au stade B3B4. Sur la même image, on peut observer en dehors des rangs d'autres tâches vertes qui correspondent à des adventices. La figure A.2b est la même image après qu'un filtre Otsu ait été appliqué et que l'image ait été réorientée de sorte que les rangs soient alignés avec l'axe  $Y$ . (Voir les sections 3.1.1 et 3.1.2). On remarque que les adventices ont été très bien éliminées, ce qui est représentatif de ce qui est obtenu tant sur des images virtuelles que sur des images réelles.



(a) Une image d'un champ virtuel. (b) Image Otsu alignée verticalement.

FIGURE A.2 – Exemple d'une image synthétique d'un champ de tournesols au stade B3/B4 et de l'image d'Otsu associée verticalement ajustée. Le champs a été généré avec une distance inter-rangs de 70 cm et une distance inter-plantes de 20 cm avec une variabilité de 5% pour les deux. Cela donne pour une densité initiale d'environ 7 plantes/ $m^2$ . Les tournesols avaient une probabilité uniforme de lever de 0,9 sur la surface du champ. L'image a été capturée avec un drone simulé à une hauteur de 10 m avec une caméra d'une distance focale de 50 mm.

### *Placer le soleil*

Dans Unity, le soleil est un type de source lumineuse parmi d'autres et son élévation et son azimut peuvent être simulés en utilisant les paramètres de rotation disponibles via le *CP Transform*. Par conséquent, afin de générer une variété de champs de culture différemment ombragés dans nos ensembles de données synthétiques, nous avons ajouté quelques paramètres pour contrôler à la fois l'élévation et l'azimut.

### *Simuler le vol de drone*

Pour notre étude, la seule caractéristique d'un drone qu'il est obligatoire de simuler est la caméra par laquelle les images seront prises. L'altitude du drone est la hauteur à laquelle la caméra prend les images. La distance focale de la caméra et la taille du capteur peuvent également être configurées à l'aide de deux paramètres. En outre, le plan de vol d'un drone tient généralement compte des proportions de chevauchement sur les axes  $Y$  et  $X$  entre les images qu'il capture pour éviter d'avoir des cultures coupées en deux au bord des images. Deux autres paramètres contrôlent ce chevauchement. La figure A.3 décrit un plan de vol du drone.

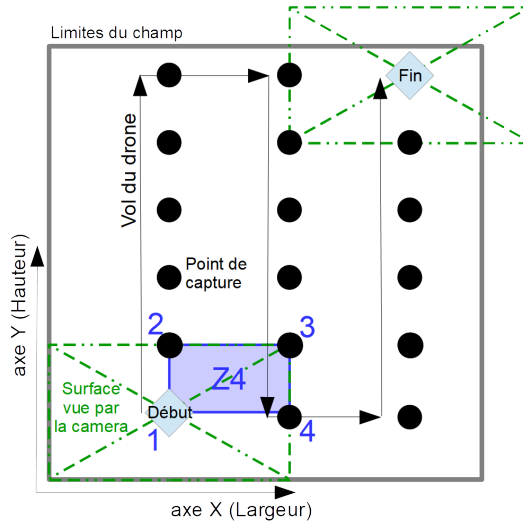


FIGURE A.3 – Schéma d'un plan de vol d'un drone au-dessus du champ de culture virtuel. La position de départ est calibrée pour capturer le coin inférieur gauche du champ. Les autres points de capture sont calculés en fonction du chevauchement des images configuré sur les axes  $X$  et  $Y$  (ici, 50 % sur les deux). Par conséquent, les images de la limite supérieure et droite du champ peuvent dépasser ces points de chevauchement. La zone nommée  $Z4$  est capturée quatre fois, une par chacun des quatre points de capture numérotés en bleu.

*Détection automatique de plants*

```
# constants
MAX_AGE = 10 #Plant's maximum age
Age_Sep = 0.5 #Plant's max age growth behaviour separation
MAX_INTERNODE_RADIUS = 1.2 #internode growth factor radius
Ir = 0.5 #Internode starting radius
Idr = 1.05 #Internode radius growth factor
MAX_INTERNODE_LENGTH = 2 #internode growth factor length
Il = 1 #Internode starting length
Idl = 1.05 #Internode length growth factor
MAX_LEAF_SIZE = 2 #max leaf growth factor
Lr = 0.3 #Leaf starting Size
Ldr = 1.2 #Leaf Growth Factor
Ldf = 0.9 #Leaf dying age factor
Axiom: A(0)
derivation length: 10

production:
I(l,r): #Internode
    if l < MAX_INTERNODE_LENGTH * Il:
        if r < MAX_INTERNODE_RADIUS * Ir:
            produce I(Idl*l, Idr*r)
        else:
            produce I(Idl*l, r)
    else:
        if r < MAX_INTERNODE_RADIUS * Ir:
            produce I(l, Idr*r)
        else:
            produce I(l, r)

L(age,s): #Leaf
    if s < MAX_LEAF_SIZE * Lr:
        produce L(age+1,Ldr*s)
    else:
        produce L(age+1,s)

A(age): # Apex
    if age == MAX_AGE: #TOP
        produce I(0.25*Il,Ir)L(1,0.5*Lr)I(0.25*Il,Ir)T
    elif age < Age_Sep*MAX_AGE: #LOWER PART, OPPOSITE PHYLLOTAXY
        produce I(Il,Ir)L(1,Lr)/(180)L(1,Lr)/(137.5)A(age+1)
    else: #UPPER PART, ALTERNATE PHYLLOTAXY
        produce [f(0.5*Il)L(1,Lr)]/(137.5)I(Il,Ir)L(1,Lr)/(137.5)A(age+1)

homomorphism:
I(l,r) --> F(l,r)
L(age,s):
    if age < Ldf * MAX_AGE:
        produce ~("Leaf", s)
T --> ~("Flower")
```

Listing A.4 – Fichier L-py utilisé pour générer les tournesols sous Blender

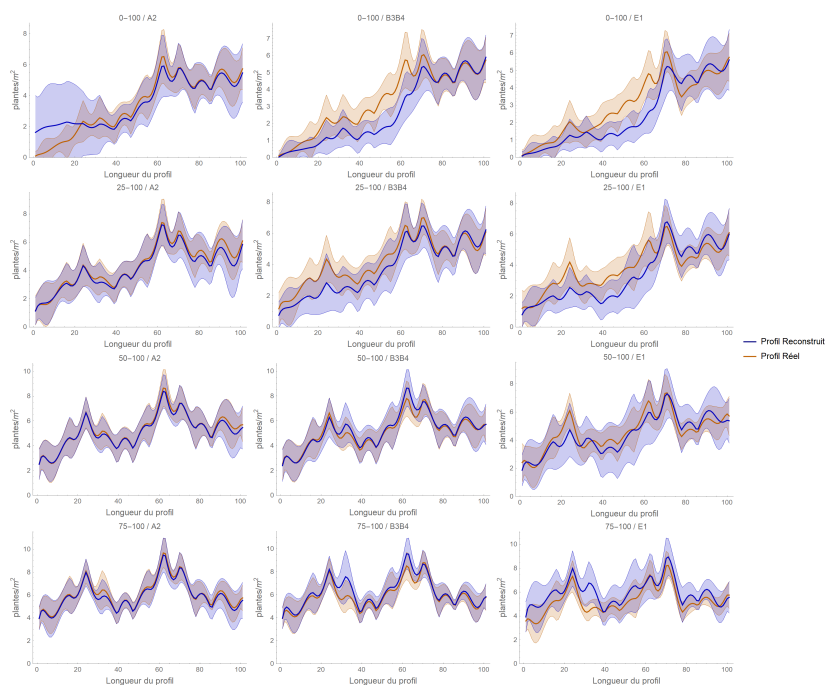


FIGURE A.5 – Profils de densité moyen de champs virtuels créés dans le logiciel Unity pour les 12 couples de conditions (gradient / stade de développement). Le couple de conditions le plus simple est en bas à gauche ; le couple de condition le plus difficile est en haut à droite (il correspond à la figure 5.4).

## BIBLIOGRAPHIE

- [1] BLENDER ONLINE COMMUNITY, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [2] F. BOUDON, C. PRADAL, T. COKELAER, P. PRUSINKIEWICZ & C. GODIN, « L-Py : An L-System Simulation Framework for Modeling Plant Architecture Development Based on a Dynamic Language », *Frontiers in Plant Science* **3** (2012) (English), article no. 76 (20 pages).
- [3] M. BURKE & D. B. LOBELL, « Satellite-based assessment of yield variation and its determinants in smallholder African systems », *Proceedings of the National Academy of Sciences* **114** (2017), n° 9, p. 2189-2194.
- [4] H. GARCÍA-MARTÍNEZ, H. FLORES-MAGDALENO, A. KHALIL-GARDEZI, R. ASCENCIO-HERNÁNDEZ, L. TIJERINA-CHÁVEZ, M. A. VÁZQUEZ-PEÑA & O. R. MANCILLA-VILLA, « Digital Count of Corn Plants Using Images Taken by Unmanned Aerial Vehicles and Cross Correlation of Templates », *Agronomy* **10** (2020), n° 4, article no. 469 (16 pages).
- [5] F. GNÄDINGER & U. SCHMIDHALTER, « Digital counts of maize plants by unmanned aerial vehicles (UAVs) », *Remote sensing* **9** (2017), n° 6, article no. 544 (15 pages).
- [6] J. M. GUERRERO, G. PAJARES, M. MONTALVO, J. ROMEO & M. GUIJARRO, « Support vector machines for crop/weeds identification in maize fields », *Expert Systems with Applications* **39** (2012), n° 12, p. 11149-11155.

- [7] M. GUIJARRO, G. PAJARES, I. RIOMOROS, P. HERRERA, X. BURGOS-ARTIZU & A. RIBEIRO, « Automatic segmentation of relevant textures in agricultural images », *Computers and Electronics in Agriculture* **75** (2011), n° 1, p. 75-83.
- [8] S. HAN, Q. ZHANG, B. NI & J. F. REID, « A guidance directrix approach to vision-based vehicle guidance systems », *Computers and Electronics in Agriculture* **43** (2004), n° 3 (en), p. 179-195.
- [9] S. HINTERSTOISSER, O. PAULY, H. HEIBEL, M. MAREK & M. BOKELOH, « An Annotation Saved is an Annotation Earned : Using Fully Synthetic Training for Object Instance Detection », <http://arxiv.org/abs/1902.09967>, 2019.
- [10] P. HOFMANN, « Multi-agent Systems in Remote Sensing Image Analysis », in *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - Volume 1 : ICAART 2019*, INSTICC, SciTePress, 2019, p. 178-185.
- [11] E. JACOPIN, N. BERDA, L. COURTEILLE, W. GRISON, L. MATHIEU, A. CORNUÉJOLS & C. MARTIN, « Using Agents and Unsupervised Learning for Counting Objects in Images with Spatial Organization : », in *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*, SCITEPRESS - Science and Technology Publications, 2021, p. 688-697.
- [12] Y.-C. JHANG, A. PALMAR, B. LI, S. DHAKAD, S. K. VISHWAKARMA, J. HOGINS, A. CRESPI, C. KERR, S. CHOCKALINGAM, C. ROMERO, A. THAMAN & S. GANGULY, « Training a performant object detection ML model on synthetic data using Unity Perception tools », <https://blogs.unity3d.com/2020/09/17/training-a-performant-object-detection-ml-model-on-synthetic-data-using-unity-computer-vision-tools/>, 2020.
- [13] Z. JIN, G. AZZARI, M. BURKE, S. ASTON & D. B. LOBELL, « Mapping smallholder yield heterogeneity at multiple scales in Eastern Africa », *Remote Sensing* **9** (2017), n° 9, article no. 931 (15 pages).
- [14] A. JULIANI, V.-P. BERGES, E. TENG, A. COHEN, J. HARPER, C. ELION, C. GOY, Y. GAO, H. HENRY, M. MATTAR & D. LANGE, « Unity : A General Platform for Intelligent Agents », <http://arxiv.org/abs/1809.02627>, 2020.
- [15] N. LEOPOLD, « Algorithmic Botany via Lindenmayer Systems in Blender », Bachelor Thesis, TU Wien, Faculty of Computer Science, 2017.
- [16] A. LINDENMAYER & P. PRUSINKIEWICZ, *Algorithmic Beauty of Plants*, Springer New York, New York, 1990 (English), OCLC : 958528770.
- [17] D. B. LOBELL, G. AZZARI, M. BURKE, S. GOURLAY, Z. JIN, T. KILIC & S. MURRAY, « Eyes in the Sky, Boots on the Ground : Assessing Satellite-and Ground-Based Approaches to Crop Yield Measurement and Analysis », *American Journal of Agricultural Economics* **102** (2020), n° 1, p. 202-219.
- [18] S. I. NIKOLENKO, « Synthetic Data for Deep Learning », <http://arxiv.org/abs/1909.11512>, 2019.
- [19] J. OLSZEWSKA, « Designing Transparent and Autonomous Intelligent Vision Systems », in *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - Volume 2 : ICAART 2019*, INSTICC, SciTePress, 2019, p. 850-856.
- [20] N. OTSU, « A Threshold Selection Method from Gray-Level Histograms », *IEEE Transactions on Systems, Man, and Cybernetics* **9** (1979), n° 1, p. 62-66, Conference Name : IEEE Transactions on Systems, Man, and Cybernetics.
- [21] M. PÉREZ-ORTIZ, J. M. PEÑA, P. A. GUTIÉRREZ, J. TORRES-SÁNCHEZ, C. HERVÁS-MARTÍNEZ & F. LÓPEZ-GRANADOS, « Selecting patterns and features for between-and within-crop-row weed mapping using UAV-imagery », *Expert Systems with Applications* **47** (2016), p. 85-94.
- [22] K. PERLIN, « An image synthesizer », *ACM SIGGRAPH Computer Graphics* **19** (1985), n° 3, p. 287-296.
- [23] A. DE LA RE, F. ABAD, E. CAMAHORT & M. C. JUAN, « Tools for Procedural Generation of Plants in Virtual Scenes », in *Computational Science - ICCS 2009* (Berlin, Heidelberg) (G. Allen, J. Nabrzyski, E. Seidel, G. D. van Albada, J. Dongarra & P. M. A. Sloot, éd.), Lecture Notes in Computer Science, Springer, 2009, p. 801-810 (en).
- [24] J. RIBERA, Y. CHEN, C. BOOMSMA & E. J. DELP, « Counting plants using deep learning », in *2017 IEEE global conference on signal and information processing (GlobalSIP)*, IEEE, 2017, p. 1344-1348.
- [25] F. SADEGHI & S. LEVINE, « CAD2RL : Real Single-Image Flight without a Single Real Image », <http://arxiv.org/abs/1611.04201>, 2017.

- [26] A. G. SCHUT, P. C. S. TRAORE, X. BLAES & A. ROLF, « Assessing yield and fertilizer response in heterogeneous smallholder fields with UAVs and satellites », *Field Crops Research* **221** (2018), p. 98-107.
- [27] J. TOBIN, R. FONG, A. RAY, J. SCHNEIDER, W. ZAREMBA & P. ABBEEL, « Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World », <http://arxiv.org/abs/1703.06907>, 2017.
- [28] UNITY TECHNOLOGIES, « Unity 2019.4.15 », 2020, <https://unity.com/>.
- [29] ———, « Unity Perception Package », <https://github.com/Unity-Technologies/com.unity.perception>, 2020.
- [30] Z. ZOU, Z. SHI, Y. GUO & J. YE, « Object Detection in 20 Years : A Survey », <http://arxiv.org/abs/1905.05055>, 2019.

---

ABSTRACT. — This paper focuses on the detection of plants from aerial images captured by drones. An efficient detection method would benefit both farmers who wish to know the state of their crops and predict their yield and agronomists to facilitate the acquisition of experimental data and reduce the cost.

We propose a two-step method with, first, an unsupervised learning, then the implementation of a weakly parameterized multi-agent system. In the first step, a clustering system estimates the position of rows and plants based on a priori known geometric constraints, namely that crop rows are globally equidistant from each other and that the same is true for plants within a row. In a second step, the multi-agent system refines the estimation provided by the clustering step and isolates each plant.

We demonstrate the performance of the method on sunflower plant counting tasks. The results obtained are comparable to those of the state of the art on easy problems and significantly better on hard problems. We then present results on the identification of density gradients in a field based on the distribution of emerged plants.

An important contribution of this work also concerns the development of a tool for generating virtual fields with the game engine *Unity*. It is thus possible to easily generate realistic datasets corresponding to various situations, thus overcoming the difficulty of obtaining labelled datasets.

KEYWORDS. — Unsupervised learning, multi-agent system, analysis of images from UAVs.

---

*Manuscrit reçu le 30 mars 2021, révisé le 15 juillet 2021, accepté le 1<sup>er</sup> septembre 2021.*