

AntsTracking : une tentative de manuel

Maxime Woringer

November 22, 2013

Contents

1 Principe	2
2 Aspects techniques	3
3 Installation	3
3.1 Téléchargement	3
3.2 Installation	4
3.2.1 Version installable	4
3.2.2 Version Live CD	4
4 Fonctionnement	4
4.1 Préalable : acquisition des images	4
4.2 Préalable : conversion du film	5
4.3 Création de l'image de référence	6
4.4 Tracking proprement dit	6
5 Problèmes connus/paramètres importants à prendre en compte/solution proposée	7
6 Licence	8
7 Contact	8
8 Références	8

AntsTracking est un logiciel permettant le *tracking* de particules au cours du temps. Pour cela, une série d'images est chargée et les paramètres de l'analyse peuvent soit être ajustés en temps réel par une interface graphique simple soit programmés pour une utilisation en ligne de commande.

AntsTracking se compose de plusieurs modules :

- **AntsTracking** à proprement parler, qui extrait de la série d'images les coordonnées de chaque détection (x, y, t) .
- **AntsAnalyse**, une série de scripts permettant d'une part de reconstituer des traces de fourmis à partir des coordonnées (x, y, t) extraites par **AntsTracking** et d'autre part d'effectuer des statistiques dessus (notamment le MSD (*mean square displacement*, déplacement quadratique moyen, un indicateur basique pour caractériser les régimes diffusifs).
- Quelques scripts épars **AntsPlot** visant à la représentation graphique des traces reconstituées.

Ce manuel présente (pour le moment) uniquement la partie **AntsTracking**, qui constitue le programme principal.

1 Principe

Le principe de fonctionnement de **AntsTracking** est le suivant :

- Le programme est lancé et l'adresse de la séquence d'images est spécifié (une étape de conversion préalable des images est en général nécessaire)
- Une interface graphique simple s'affiche alors, permettant de visualiser en direct les fourmis détectées, de naviguer dans le film et de régler les paramètres de détection
- Les détections sont enregistrées dans un fichier texte, au format **image**, **x**, **y**, qui peut être ouvert par exemple avec **Scilab** pour les analyses et la visualiation.

Le logiciel s'en sort en général assez bien, même dans des situations où de très nombreuses fourmis sont visibles (nous avons réussi à suivre plus d'une centaine de fourmis simultanément). Néanmoins, suivant la résolution de l'acquisition, les fourmis ne sont plus forcément détectées individuellement, surtout si elles ont tendance à se suivre en se touchant ou à former des amas (en contact avec une goutte d'eau sucrée par exemple).

Au niveau des temps de calcul, la détection des fourmis est assez rapide, et le programme tournait sur ma machine (un ordinateur portable d'entrée de gamme de 2011) à environ dix images par secondes (les images étaient enregistrées à 10 fps au format 640×480). En gros, le temps de traitement était équivalent au temps d'acquisition, ce qui permettait en pratique de faire

l'acquisition pendant la journée, le traitement pendant la nuit et de disposer des données le matin du jour suivant.

2 Aspects techniques

Le programme est écrit dans le langage de programmation Python et utilise la bibliothèque de traitement vidéo OpenCV. En pratique, des traitement visant à augmenter le contraste et le rapport signal sur bruit de l'image sont appliqués :

- Soustraction d'une image de référence correspondant au «bruit de fond». Cette étape permet en particulier de s'affranchir en partit des zones de discontinuités ou d'objets présents dans le champ.
- Conversion en noir et blanc (prérequis pour les filtres suivant)
- Application d'un filtre Laplacien. Cette étape permet d'augmenter fortement le contraste des détections/fourmis. En effet, le filtre laplacien calcule la dérivée seconde de l'image, et est donc très sensibles aux variations de couleur.
- Seuillage : l'image noir et blanc en sortie du Laplacien est convertie en image binaire (avec deux valeurs de pixel possible : 0 et 255).

Après ces premiers pré-traitements, un algorithme pré-programmé dans la librairie `OpenCV` est appliqué. Celui-ci vise à extraire les coordonnées de «zones d'intérêt» dans l'image. Comme l'image d'entrée de cet algorithme est une image binaire, celui-ci détecte uniquement les zones que l'on a mises en lumière par le pré-traitement.

Il est important de noter qu'il existe une infinité de pré-traitements et d'algorithmes de *tracking* possible. Ici, l'algorithme implémenté est le plus simple possible (bien que l'on puisse déjà dissenter éternellement sur les choix qui ont été faits).

3 Installation

3.1 Téléchargement

Il existe deux versions :

- Version *live CD*, "batteries included". Disponible sur CD

- Version "installable", disponible à l'adresse <http://www.eleves.ens.fr/home/woringer/>

3.2 Installation

3.2.1 Version installable

Je ne vais pas détailler ce cas-là, parce que c'est une grosse galère sur la majorité des systèmes (en particulier Windows, avec Mac on y arrive, et avec GNU/Linux ça ne pose pas de souci en général). Il faut :

- Installer Python,
- Installer la bibliothèque `OpenCV`, avec les *binding* Python (en gros pour que la bibliothèque puisse être utilisée à partir de Python),
- Décompresser `AntsTracking`.

3.2.2 Version Live CD

Comme j'arrivais à l'installer nulle part, j'ai fait une copie de mon ordinateur sur un DVD. On peut le démarrer sur un PC (ou un mac Intel) et tout est déjà installé.

En gros, il faut mettre le CD dans le lecteur, redémarrer l'ordinateur, au moment où il démarre il faut appuyer sur **Suppr**, **F1**, **F2**, ..., ou **Échap** (ça dépend de la marque de l'ordinateur et du modèle) et lui dire de démarrer sur le CD.

À ce moment là, si tout se passe bien, le CD démarre, un écran «Ubuntu» s'affiche et vous pouvez passer à la suite.

4 Fonctionnement

4.1 Préalable : acquisition des images

Dans cette partie, quelques conseils en vrac pour avoir une acquisition de pas trop mauvaise qualité :

Champ d'acquisition L'algorithme est très sensible aux irrégularités. En théorie, celles-ci n'influencent pas les détections puisqu'une différence avec une image de référence est réalisée. Néanmoins, en pratique, le moindre décalage entre l'image de référence et l'acquisition (par exemple parce que la caméra a légèrement bougé) est très fortement amplifié

par la détection de contraste et crée de nombreuses fausses détections. De plus, il est préférable que les images présente le plus fort contraste, et donc que le fond soit clair. Nous réalisons nos expériences soit sur des feuilles de papier A3, soit sur des vitres posées sur des feuilles en papier blanc.

Appareil d'acquisition nous disposions d'une webcam bas de gamme pour faire les acquisitions. C'était une mauvaise idée, car la résolution est bien trop faible. Si vous disposez d'un appareil photo pour filmer ou d'un caméscope, ça facilitera grandement les analyses par la suite. La caméra doit être fixée et à la verticale par rapport au plan qu'on filme (parce que sinon on n'observe pas les bonnes distances). On avait un champ de 30×30 cm si je me souviens bien, et on doit pouvoir moduler en fonction de la résolution de la caméra et de ce que vous voulez faire. À noter qu'il n'est pas forcé de prendre les films dans la qualité maximale. En particulier parce que l'algorithme peut devenir très lent avec de grandes images (là j'ai pas de chiffres en tête). Ensuite parce qu'on se retrouve vite avec énormément de films.

Éclairage l'éclairage est un paramètre qui nous avait posé problème. En effet, si celui-ci varie trop au court de l'expérience, la qualité des détections varie elle aussi énormément. De même, s'il y a des ombres sur le plateau, il est possible d'avoir des biais de détection parce que l'algorithme est meilleur avec un certain contraste... D'une manière générale, nous avons constaté qu'éclairer plus n'améliorait pas les expériences, car la caméra avait tendance à saturer et à rendre beaucoup moins bien les contrastes.

Image de référence Le traitement du film commence par le calcul d'une différence avec une image de référence. Il est donc nécessaire que vous ayez au moins une image dans votre film qui présente tout le dispositif expérimental visible (gouttes de nourriture, passerelle ou autre) mais aucune fourmi. Si jamais par erreur ou contrainte technique vous avez commencé l'acquisition *in media res*, le plus simple est d'effacer sur une image de l'acquisition les fourmis visibles avec un logiciel de retouche photo.

4.2 Préalable : conversion du film

Une fois que vous aurez réalisé les acquisitions, il est nécessaire de le convertir en une série d'images nommées de la manière suivante : `image0.jpg`,

image1.jpg, image2.jpg, ... Pour cela, le plus simple est d'utiliser l'utilitaire `ffmpeg`.

En gros, après avoir lancé le *Live CD*, il faut :

- Connecter la clé USB/le disque dur/la carte mémoire à l'ordinateur. Normalement, celui-ci s'ouvre automatiquement, et un dossier apparaît dans le répertoire `/media/`.
- Créer un dossier pour votre vidéo dans le dossier `AntsTracking/Videos/`, en évitant les espaces et les caractères accentués.
- Ouvrir un terminal à partir du menu *Applications -> Terminal* et taper `cd Bureau/AntsTracking/` pour se placer dans le répertoire de `Antsracking`
- Lancer la conversion en tapant dans le terminal la commande `ffmpeg -i /media/{nom clé usb}/{dossier}/{film} -o Videos/{nom du dossier que vous venez de créer}/image%d.jpg` (si vous appuyez plusieurs fois sur la touche `Tab` en tapant l'adresse de la vidéo, la console vous propose plusieurs choix à chaque fois, ça permet d'éviter les erreurs de frappe).

4.3 Création de l'image de référence

Il est possible de choisir de manière interactive l'image de référence en appuyant sur la touche `b` pendant que le film joue (voir plus bas). Lors de la première utilisation, il est nécessaire d'initialiser cette image en copiant la première image du film (`image0.jpg`) dans le dossier `/reference/` et en la renommant en `reference.JPG` (attention aux majuscules).

4.4 Tracking proprement dit

Une fois la conversion achevée, vous pouvez lancer `AntsTracking` en tapant (toujours dans la même console) `python AntsTracking.py`. Le programme vous demande alors le dossier d'entrée (celui que vous avez créé pour accueillir le film lors de la conversion), et le nom du fichier de sortie (qui sera créé dans le dossier `résultat`). Valider par `Entrée` à chaque fois.

Le programme affiche alors une interface graphique. Il se pilote presque entièrement au clavier et la liste des raccourcis s'affiche dans la console :

Hot keys:

ESC - quit the program

s - show/hide points
r - auto-initialize tracking
c - delete all the points
n - switch the "night" mode on/off
SPACE - play/pause
m - Play video faster
l - Play video slower
e - save the frame
b - next reference frame
i - switch view

Les raccourcis importants sont :

- i : pour changer entre l'image originale et l'image après tous les pré-traitements
- s : pour afficher/cacher les détections
- ESPACE : play/pause
- m/l : plus vite/plus lentement. l permet de jouer le film à l'envers.
- b : remplacer l'image de référence par l'image actuelle

À l'écran :

- Des curseurs permettant de régler les paramètres du pré-traitement. Leur effet est visible en appuyant sur i et en visualisant l'image binaire. Il est toujours intéressant de faire quelques essais et de bidouiller un peu les paramètres pour avoir de meilleures détections. D'une manière générale, l'algorithme est assez sensible à ceux-ci. Attention, une trop forte valeur d'érosion peut ralentir énormément les calculs (plus d'une centaine de fois).
- L'image, avec son numéro et la vitesse de déplacement.

Une fois le *tracking* terminé, appuyer sur la touche Echap, le fichier texte est normalement enregistré.

5 Problèmes connus/paramètres importants à prendre en compte/solution proposée

Le tracking est de mauvaise qualité De nombreuses causes sont possibles (mauvais éclairage, mauvaise image de référence, mouvement de

la caméra, mauvais contraste mais aussi mauvais paramétrage du logiciel).

Est-ce qu'on *tracke* toutes les fourmis ? Tout dépend de la qualité d'acquisition. Nous avons constaté sur nos acquisitions que le logiciel mettait en général un point sur chaque fourmi lorsque la densité n'est pas trop élevée. Dès que celle-ci devenait importante, l'algorithme avait tendance à saturer et le nombre de points ne reflétait plus le nombre de fourmis. Par conséquent, il faut être conscient si l'on veut réaliser un comptage que celui-ci n'est valable que si les fourmis ne se touchent pas (ce qui n'est pas le cas lorsqu'elles boivent à une goutte d'eau par exemple).

Comment reconstituer les traces ? `AntsTracking` produit en sortie une liste de coordonnées (x, y, t) , il est nécessaire d'assembler les points pour reformer les traces. Cette procédure est une procédure complexe puisqu'on ne veut pas perdre les fourmis même si elles sont détectées de manière intermittentes. J'ai écrit un premier programme pour reconstituer les traces en `Python`, et une seconde version qui gérait en théorie mieux les croisements en `Scilab`, mais je n'ai jamais eu le temps de la tester. Si ce programme-là vous intéresse, contactez-moi :)

6 Licence

`AntsTracking` est un logiciel libre, vous êtes libre de le redistribuer et de le partager selon les termes de la licence GPL version 3 ou supérieure (disponible à l'adresse <http://gnu.org>).

7 Contact

En cas de problème avec le logiciel, n'hésitez pas à me contacter à l'adresse <mailto:maxime.woringer@ens.fr>, j'essaie de résoudre les bugs rapidement.

8 Références

- `AntsTracking` a été conçu pour un projet de TIPE (travail d'initiative personnelle encadrée) en 2011 intitulé «Étude du mouvement de la fourmi *Crematogaster scutellaris* hors du nid» (Lucile Rey-Cadilhac, Laurence Boulet, Lucas Torres et Maxime Woringe). Le rapport de

TIPE est disponible en ligne à l'adresse <http://www.eleves.ens.fr/home/woringer/>.

- Ce même programme a été réutilisé en 2012 par un groupe de TIPE de deuxième année sur les mêmes fourmis, mais avec un équipement technique plus performant.

Pour conclure : un petit chat, c'est vraiment trop mignon.