

Expérimentation sur la perception humaine des essaims

Aymeric Hénard

July 17, 2023

Abstract

Ce document détaille une procédure d'expérimentation permettant d'étudier la perception humaine des essaims, et plus précisément comment les humains perçoivent les fractures pouvant se produire dans l'essaim.

1 Abstract swarm fractures

(Généré par gpt 3.5) In the context of robot swarms, social fracture refers to a breakdown in communication and coordination among the robots, which can lead to the swarm becoming less effective or even unable to accomplish its tasks. This breakdown can occur for a variety of reasons, such as technical malfunctions, changes in the environment, or the introduction of conflicting goals or objectives among the robots. In general, social fracture can be a major challenge for robot swarms, as it can limit their ability to function as a cohesive and coordinated group. To prevent or mitigate social fracture, researchers are developing various strategies, such as using machine learning algorithms to help the robots adapt to changing conditions and maintain effective communication and coordination.

2 Vocabulaire

- **Fracture de l'essaim** : Perte de la connexion sociale (communication et coordination, rupture de l'agrégat) entre un agent ou groupe d'agents et l'essaim principal.
- **Groupe F** : Agent ou groupe d'agents quittant l'essaim lors d'une fracture. Plusieurs groupe F peuvent exister simultanément.

3 Définir la fracture aux participants

3.1 Définition d'un essaim

Un essaim de robots est un ensemble de robots autonomes interagissant entre eux pour se coordonner. Ils interagissent principalement en observant via leurs capteurs le comportement des robots proches, adaptant leur comportement en fonction de leurs voisins pour se coordonner. Ainsi, un ensemble de robots coordonnés constitue un groupe uni, un essaim.

3.2 Définition d'une fracture

Il arrive parfois qu'un ou plusieurs robots ne puissent plus percevoir d'autres robots. Toute interaction devient impossible, et en conséquences il n'est plus possible pour ces robots de se coordonner. On observe alors des robots qui sont isolés du reste de l'essaim principal. Même si ces robots isolés peuvent parfois former eux-mêmes un plus petit essaim, ils restent incapables de se coordonner avec le reste de l'essaim principal, et donc isolés (vidéos exemple). On peut définir cette perte d'interaction et de coordination entre les robots de l'essaim comme étant une fracture sociale.

4 Quel est l'intérêt de cette expérimentation?

- Cette expérimentation a pour but justifier la nécessité d'ajouter des informations (en RA par exemple) dans certaines situations. Et plus précisément ici pour aider l'utilisateur à maintenir l'intégrité de l'essaim en empêchant les fractures.
- Dans un second temps, cette expérimentation peut permettre d'identifier les informations nécessaires pour un humain, lui permettant d'identifier une fracture. Cependant, elle ne pourra pas permettre de les tester et identifier des seuils. En effet, cela peut faire l'objet d'une expérimentation à part entière. En conséquence, l'identification de ces paramètres seront l'ouverture vers une analyse plus poussée, probablement pas réalisable dans le cadre de cette thèse.

5 Cas de fracture d'un essaim

Dans le contexte d'un essaim implémentant le modèle des boids de Reynolds, il est possible d'identifier différentes situations menant à une fracture de l'essaim. Bien que la plupart de ces situations ne sont possibles qu'en utilisant ce modèle, elles permettent cependant d'étudier la perception humaine des fractures de l'essaim, en proposant différents types de fracture. Avec certaines de ces fractures, il devient possible de ne faire varier qu'un paramètre et d'observer son impact sur la perception de l'humain, et ainsi identifier les paramètres importants dans la perception et idéalement leurs seuils.

Voici les différents cas de fracture identifiables d'un essaim implémentant les boids de Reynolds :

- Les fractures causées par l'environnement (exemple : un obstacle qui provoque la séparation du groupe en deux).
- Les fractures causées par un état de l'essaim instable. Notamment à cause d'un mauvais équilibre des paramètres.
 - Une force de répulsion trop importante provoquant une dispersion des agents. Chaque agent devient alors un groupe F. Ce cas peut apparaître lors d'une tentative de couverture de zone mal maîtrisée par un opérateur.
 - Un flocking instable, où chaque agent est proche de la limite du champ de perception de ses voisins. En conséquence, chaque légère perturbation provoque la perte d'un ou plusieurs agents.
 - Quand les agents sont suffisamment éloignés les uns des autres (exemple : couverture de zone) et que les paramètres incitent les agents à se rapprocher les uns des autres, l'essaim se fracture en plusieurs groupes si l'attraction est trop forte.
- Les fractures causées par la panne d'un agent. Dans le cas d'un essaim de robots, des problèmes matériels peuvent apparaître. En conséquence, de nombreux dysfonctionnements peuvent provoquer la fracture d'un agent avec l'essaim principal :
 - Arrêt de l'agent
 - Ralentissement
 - Blocage de la direction : avancer tout droit
 - Blocage de la direction : rotation permanente
 - Accélération
 - ...

6 Question de recherche

L'expérimentation présentée dans ce document a pour objectif d'étudier la perception humaine des essaims en cas de fracture, et plus précisément de répondre aux questions suivantes :

- Un humain perçoit-il les fractures de l'essaim?
- Un humain perçoit-il les fractures avant que celles-ci n'apparaissent?

- (Secondaire) Quelles sont les informations qui permettent à l'humain d'identifier et/ou prévoir une fracture potentielle?

7 Hypothèses de recherche

Cette section regroupe les différentes hypothèses de recherche que l'on souhaite vérifier grâce à cette expérimentation. Voici les différentes hypothèses envisagées :

1. Les humains perçoivent les fractures des essaims.
2. Les humains ne perçoivent pas les fractures avant que celles-ci n'apparaissent.
3. Les humains ne perçoivent pas le moment/seuil exact de la fracture de l'essaim.
4. Une distance relative "marquée" entre les agents de l'essaim permettrait de mieux identifier les fractures que les distance relative "homogène".
5. Une direction relative "différente" permettrait de mieux identifier et prévoir les fractures qu'une direction relative "similaire"
6. Une direction relative "centres de gravité" permet de mieux identifier et prévoir les fractures qu'une direction relative "toute direction".

8 Expérimentation

Après une phase de présentation de l'auto-organisation d'un essaim et des comportements, différents clips (simulations) sont présentés les uns après les autres au participant. Ces clips peuvent contenir ou non des fractures de l'essaim. En visionnant ces clips, l'utilisateur doit dire si ce clip contient une fracture ou non. Il peut répondre après l'apparition de la fracture si le clip en contient une, mais aussi avant si il pense qu'une fracture va apparaître. Pour cela, il lui suffira d'appuyer sur l'un des deux boutons proposés. Une fois le choix du participant prononcé, le clip continue jusqu'à sa terminaison. Puis le clip suivant se lance.

8.1 Déroulement

- Présentation au participant de l'expérimentation, signature du formulaire de consentement.
- (Facultatif) Questionnaire pour recueillir des informations sur le participant.
- Phase d'initiation aux essaims et comportements collectifs (contenu à définir)
- Début de l'expérimentation, les clips vont défiler les uns après les autres et le participant doit répondre à chaque clip, tout en explicitant son choix à l'oral.
- Fin de l'expérimentation, remplissage du questionnaire de fin.

8.2 Objectif du participant

Les participants doivent faire de leur mieux pour :

- Choisir la bonne réponse pour chaque clip
- Valider cette réponse dès qu'ils sont sûr d'eux (implicitement le plus rapidement possible)

8.3 Caractéristiques des clips

Les clips présentés aux participants sont caractérisés comme tel :

- Les clips sont d'une durée variable en fonction du scénario
- Certains clips pourront montrer des situations similaires
- Quand un clip montre un fracture, il existe un autre clip proposant le même comportement d'essai, mais avec des paramètres légèrement différents pour qu'il n'y ai pas de fracture.
- Les clips sont visionnés jusqu'à la fin par tous les participants
- Les clips contiennent 2 flags temporels :
 - Le premier correspond au début du clip (car il n'y a pas de changement de paramètre durant le clip).
 - Le second flag correspond au moment où la fracture se produit.
- Potentiellement, plusieurs algo seront utilisés pour proposer suffisamment de situations différentes.

8.4 Paramètres à maintenir stable durant l'expérimentation

Voici les différents paramètres qu'il est nécessaire de maintenir constant pour éviter toute altération des résultats obtenus :

- Position du participant par rapport à l'essai
- Point de vue du participant (RV ou écran)
- Champ de perception des agents
- Environnement
- Nombre d'agents
- Algo des agents
- Visuel des agents (et taille)

8.5 Données

Différentes données seront préparées lors de la mise en place des clips :

- Horodatage de la fracture
- Catégorisation des clips

Différentes données seront récoltées durant l'expérimentation :

- Horodatage de la réponse du participant
- Le résultat de la réponse du participant (si il ne répond pas la réponse est considérée comme étant: non il n'y a pas de fracture)
- Vidéo ou audio : Thinking Out Loud, mais uniquement si cela ne les empêche pas de se concentrer (pour éviter de ralentir les participants [[Hertzum et al., 2009](#)]), car on leur demande en priorité d'être performant.

Enfin, des données seront récoltées à la fin de l'expérimentation :

- Un questionnaire subjectif (à définir)

9 Démarche d'analyse des résultats

9.1 Métrique : “Fracture visibility score”

La métrique “Fracture Visibility Score” donne une information sur la visibilité de la fracture d'un clip. En effet, plus sa valeur est élevée, plus la fracture est visible. Cependant, quand sa valeur est proche de 1, il est difficile d'observer une fracture. Cet indice de visibilité se base les distances entre les agents et les groupes. En effet, pour l'obtenir on mesure la distance la plus significative entre deux groupes fracturés, que l'on compare avec la distance moyenne entre les agents du groupe principal. **En conséquence, cette formule ne s'applique que s'il existe une fracture dans l'essaim.** On obtient la formule suivante :

$$FractureVisibilityScore = SignificantDistanceFracture / MeanKNNDistance(BiggerCluster)$$

Cette métrique est utilisé au moment de la fracture, et au moment où sa valeur est la plus élevée, indiquant le moment où la fracture est la plus visible.

9.1.1 SignificantDistanceFracture

Pour mesurer la valeur de “SignificantDistanceFracture” à l'instant t d'un clip, il est nécessaire de réaliser une clusterisation des agents de l'essaim en fonction de leurs perceptions. On considère deux agents comme appartenant au même groupe si l'agent A perçoit l'agent B ou est perçu par l'agent B. En appliquant cette règle sur tous les agents de l'essaim, on obtient plusieurs groupes si une fracture existe. Une fois les groupes obtenus, on va chercher la plus grande distance des plus petites distances séparants les groupes. On réalise l'algorithme suivant :

Algorithm 1 SignificantDistanceFracture

```
groupes ← GetGroupes(agents)
nbGroupes ← Size(groupes)
while nbGroupes >> 2 do
  minDistance ← float.MaxValue
  grp1 ← null
  grp2 ← null
  for each k in groupes do           ▷ Chercher la plus petite distance qui sépare deux groupes
    for each g in groupes do
      if k ≠ g then
        dist ← MinDistanceBetween(k,g)
        if dist << minDistance then
          minDistance ← dist
          grp1 ← k
          grp2 ← g
        end if
      end if
    end for
  end for
  grp3 ← grp1 + grp2                 ▷ Fusionner les deux groupes les plus proches
  groupes.Remove(grp1)              ▷ Supprimer de la liste les groupes d'origines
  groupes.Remove(grp2)
  groupes.Add(grp3)                  ▷ Ajouter à la liste le groupe fusionné
  nbGroupes ← Size(groupes)         ▷ Mettre à jour la taille de la liste
end while
result ← MinDistanceBetween(groupes[0],groupes[1])
```

9.1.2 MeanKNNDistances

Pour mesure la valeur de “MeanKNNDistances”, on se base sur le groupe le plus important, c'est-à-dire celui contenant le plus d'agents. Pour chaque agent de ce groupe, on va récupérer les N agents les plus proches, et moyenniser les distances les séparants. Ici, N vaut 3.

9.2 Métrique : “Separation Speed”

La métrique “Separation Speed” a pour objectif de quantifier la vitesse d’éloignement du groupe fracturé de l’essaim. Cette métrique réunie deux informations : l’orientation relative entre les deux groupes et la vitesse relative entre les deux groupes. Plus sa valeur est grande, plus le groupe fracturé s’éloigne vite de l’essaim. Pour le moment, cette valeur est mesurée comme ceci :

Algorithm 2 SeparationSpeed

```
Find fracture frame
Identify the leaving group
Identifying the two last agents linked from the swarm and the leaving group
Measure the change in distance between the two agents on several past frames from the fracture
frame.
Use that difference to calculate a speed
Return that speed
```

On peut aussi envisager comparer la vitesse d’éloignement par rapport au centre de masse du groupe. A discuter.

9.3 Catégorisation des clips

Dans le but de pouvoir interpréter les résultats en terme de comportement d’essaim, il est pertinent de proposer des catégories pour les différents clips. On peut dans un premier temps identifier 4 comportements élémentaires présents dans les clips ayant des fractures ou non :

- Flocking (coordination du mouvement)
- Swarming (en opposition au flocking)
- Expansion
- Densification (en opposition à l’expansion)

D’autres comportements apparaissent, mélangeant plusieurs des comportements élémentaires :

- Croissant (Flocking expansif horizontal)
- Path formation like (flocking en aller retour)
- Shrapnel (visible uniquement lors d’une fracture)

On peut cependant plus ou moins confirmer la classification visuelle des clips en utilisant certaines métriques mettant en avant certains aspects des comportements.

9.3.1 “TowardsCenterOfMass”

La métrique “TowardsCenterOfMass” permet d’évaluer la direction que prend l’essaim par rapport au centre de masse. Sa valeur varie entre 0 et 1 et on peut identifier 3 paliers. Entre 0.45 et 0.55, c’est le palier stable, l’essaim ne va ni vers le centre de masse, ni ne s’éloigne. On y retrouve le flocking notamment. Au dessus de ce palier, de 0.55 à 1, ce sont les agents qui se dirigent vers le centre de masse. Enfin, le palier de 0 à 0.45, ce sont les agents qui se dirigent à l’opposée du centre de masse. Cette métrique ne se base que sur la direction des agents, et ne prend pas en compte les changements de positions des agents. Elle à l’avantage de prévenir le mouvement des agents avant qu’ils ne bougent réellement. Cette métrique s’obtient selon l’algorithme suivant :

9.3.2 “EffectiveGroupMotion”

La métrique “EffectiveGroupMotion” permet d’évaluer la distance parcourue par les agents par rapport à la direction parcourue par le centre de masse des agents. La valeur de cette métrique varie de 0 à 1. Lorsque la valeur est proche de 0, cela implique que les agents bougent beaucoup plus que le centre de masse, voire que le centre de masse ne bouge pas. Cependant, lorsque la valeur est proche de 1, cela implique que le centre de masse avance autant que les agents, comportement que l’on peut retrouver dans un flocking.

Algorithm 3 TowardsCenterOfMass

```
centerOfMass ← CenterOfMass(agents)           ▷ Correspond à un une position dans l'espace
res ← 0
for each a in agents do
  dirA ← GetSpeed(a)
  dirCOM ← centerOfMass − GetPosition(a)     ▷ Soustraction de deux positions dans l'espace
  if  $\|dirA\| \equiv 0$  then
    angle ← 90                                  ▷ Si l'agent ne bouge pas, angle neutre
  else
    angle ← Angle(dirA, dirCOM)              ▷ Angle relatif entre les deux vecteurs
  end if
  res ← res + (angle/180)                    ▷ Division de l'angle par 180 pour obtenir une valeur entre 0 et 1
end for
result ← 1 − (res/Count(agents))
```

Algorithm 4 EffectiveGroupMotion

```
CM ← CenterOfMass(agents(t))                 ▷ Position du centre de masse à l'instant t
pastCM ← CenterOfMass(agents(t − 1))        ▷ Position du centre de masse à l'instant t-1
distCM ← Distance(CM, pastCM)              ▷ Calcul de la distance parcourue par le Centre de masse
meanDist ← 0
for each a in agents do
  currentPos ← GetPosition(a(t))           ▷ Position de l'agent à l'instant t
  pastPos ← GetPosition(a(t − 1))          ▷ Position de l'agent à l'instant t-1
  distAgent ← Distance(currentPos, pastPos) ▷ Calcul de la distance parcourue par un agent
  meanDist ← meanDist + distAgent
end for
meanDist ← meanDist/Count(agents)
if meanDist  $\equiv 0$  then
  resultat ← 0
else
  resultat ← distCM/meanDist
end if
```

9.3.3 “Order”

La métrique “Order” est une métrique très utilisée dans la littérature, et mesure l’alignement d’un essaim. Sa valeur varie de 0 à 1. Quand la valeur est proche de 1, cela implique que tous les agents de l’essaim vont dans la même direction. Cette métrique évalue la direction et complète la métrique “EffectiveGroupMotion” qui évalue le mouvement du groupe. Ainsi, la combinaison de ces deux métriques permet d’évaluer efficacement un flocking.

Algorithm 5 Order

```
or ← Vector3.zero
for each a in agents do
  speed ← GetSpeed(a)
  orientation ← Normalise(speed)
  or ← or + orientation
end for
resultat ←  $\|or\|$ /Count(agents)
```

9.4 De plus ...

- Dans un premier temps, analyser uniquement les réponses des participants, et vérifier leur taux de bonnes réponses. Il y a 3 taux : les clips possédant une fracture, ceux ne possédant pas de

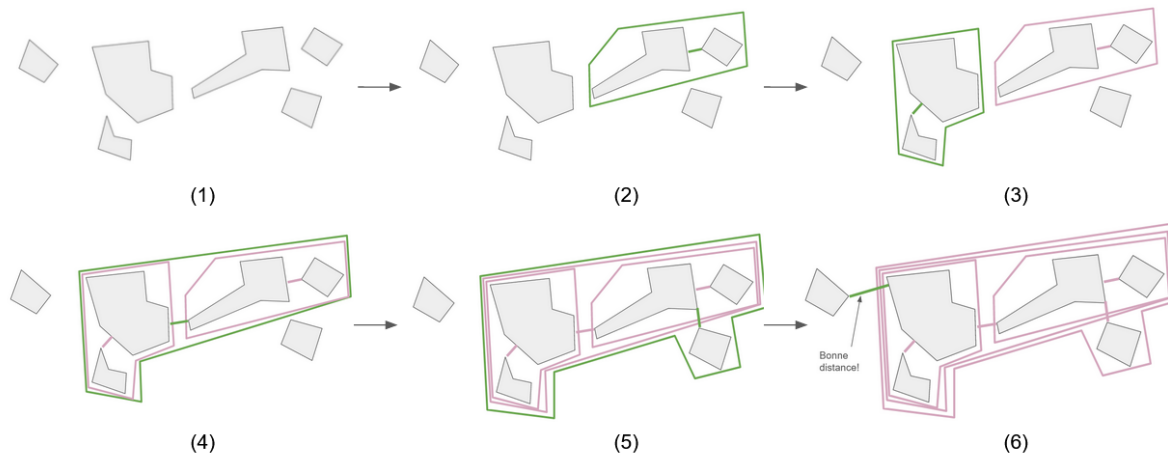


Figure 1: Schéma expliquant comment la valeur de “SignificantDistanceFracture” est obtenue.

fracture, et les deux en même temps. En profiter pour voir si il y a des moutons noirs parmi les participants.

- Ensuite, vérifier les réponses obtenues après la fracture, et celles avant la fracture. Cela permettra d’observer si les gens prévoient ou perçoivent la fracture. Analyser les taux de bonnes réponses.
- Ensuite, observer le temps moyen de réponse par rapport à la fracture, pour voir combien de temps après la fracture les gens répondent. Même chose pour les réponses avant la fracture.
- Enfin, comparer l’état de l’essaim au moment de la réponse avec les états de l’essaim sauvegardés, c’est à dire au début de la simulation et au moment de la fracture. Essayer d’étudier l’impact des paramètres étudiés, c’est-à-dire la vitesse et la direction.

10 Discussion

Différents points méritent une attention particulière :

- L’apprentissage durant l’expérimentation : Les différents clips proposés apporteront de l’expérience aux participants. Cependant, deux choix sont possibles, soit de passer directement au clip suivant quand l’utilisateur valide sa réponse, soit laisser la vidéo continuer. En fait, si on ne laisse pas la vidéo continuer, les utilisateurs n’auront pas le même temps de visionnage des clips, ni la même expérience. En effet, un utilisateur qui répond trop vite et en conséquence qui ne voit pas le résultat de ces choix risque de continuer ainsi jusqu’à la fin, car il n’aura pas d’apprentissage sur l’erreur. A l’inverse, un participant qui prend son temps peut vérifier ses choix et ainsi acquérir une expertise différente. En conséquence, pour éviter tout biais dans l’expérimentation, il est préférable de laisser les participants visualiser les clips dans leur intégralité et ainsi leur offrir la même expérience.
- Il faut faire attention au visuel des robots. En effet, les Monas possèdent des caractéristiques visuelles permettant d’identifier leur direction. Il faudra peut être les remplacer par des formes simplifiées.
- si je reproduit des scénarios, vérifier qu’ils n’ont pas appris et qu’en fait l’apprentissage est suffisamment
- Se poser la question si les informations atomiques sont suffisantes, ou que les participants voient des informations plus complexes, comme la densité, le centre de gravité.

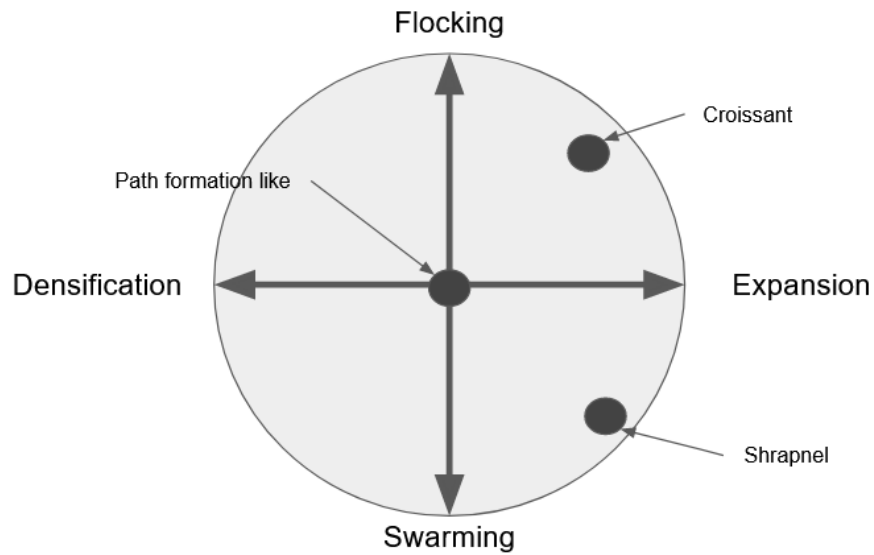


Figure 2: Schéma des catégories

- Potentielle Xp 0 : on pense que les critères sont ça (pour ne pas avoir à identifier les critères avec des humains)
- pointage spatiale des participants (tactile ou souris?)
- ne pas oublier de catégoriser les clips
- préparer des clips qui se fracture un peu mais qui se reforment

References

[Hertzum et al., 2009] Hertzum, M., Hansen, K. D., and Andersen, H. H. (2009). Scrutinising usability evaluation: does thinking aloud affect behaviour and mental workload? *Behaviour & Information Technology*, 28(2):165–181.