SMA et Traitement d’images

Pierre Le Dez ([p5ledez@enib.fr](mailto:p5ledez@enib.fr)) Ilias Maoudj ([i5maoudj@enib.fr](mailto:i5maoudj@enib.fr))

Ce document présente les résultats obtenus lors du projet de Systèmes Multi-Agents (SMA). Le sujet portait sur les SMA pour le traitement d’images. Nous avons décidé de mettre en place une version simplifié de l’algorithme présenté par MAHDJOUB et al. (2006) [1].

## Travaux précédents

Dans la publication originale, les étapes de réalisation sont les suivantes :

1. Pré-traitement de l’image
   1. Filtre de Nagao permettant de réduire le bruit tout en conservant les informations sur les contours.
   2. Rogner l’image pour ne conserver que l’information utile et réduire la taille de l’environnement des agents.
2. Calcul du gradient avec un filtre de Kirsch « amélioré » puis calcul du Gradient Vector Flow (GVF)
3. Exécution du système multi-agents.

### Environnement

L’environnement est une image en niveaux de gris. Chaque pixel contient un niveau de gris ainsi qu’un booléen indiquant si le pixel a été visité par un agent. Les agents perçoivent à la fois le GVF et le gradient. La perception du GVF est plus large que celles du gradient.

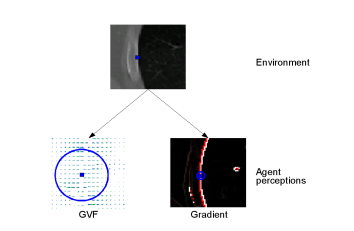


Figure 1. Deux perceptions de la même image. Le carré bleu représente l'agent et le cercle sa perception (tiré de [1])

### Agents

Quatre agents différents cohabitent :

1. Agent d’exploration.
   1. L’agent parcourt l’image à la recherche d’un contour.
   2. S’il détecte un contour, il crée un agent suivant le contour.
   3. Si l’agent ne trouve aucun contour avant un temps défini, il disparait.
2. Agent suivant un contour.
   1. L’agent établit toutes les directions possibles du contour
   2. Si l’agent trouve plusieurs directions, il se transforme en agent nœud.
   3. Si l’agent trouve une seule direction, il vérifie si le segment entre sa position et celle de son voisin est « acceptable », c’est-à-dire si elle suit effectivement le contour. Si c’est le cas, le segment entre les deux positions est tracé, sinon l’agent devient un agent nœud.
   4. Si l’agent ne trouve pas de direction correcte, il se transforme en agent fermant un contour.
3. Agent fermant un contour.
   1. L’agent tente de se connecter avec un autre agent fermant un contour. S’il réussit effectivement à se connecter alors un segment est tracé entre les deux agents fermant le contour et ils disparaissent.
4. Agent nœud.
   1. L’agent nœud négocie avec les agents suivant un contour pour établir les segments définissant le contour.

## Notre travail



L’appel du programme se fait sous la forme :

./progX <Nom\_du\_fichier> <Nombre\_agents> <-f Taille\_du\_noyau>

Les arguments <Nombre\_agents> et <-f Taille\_du\_noyau> sont optionnels.

* Par défaut, le nombre d’agents est établi à 1000.
* L’argument « -f » permet de réaliser un filtre médian sur l’image d’entrée et l’entier qui suit définit la taille du noyau (5 donne un noyau 5x5). Par défaut, il n’y a pas de pré-traitement.

### Environnement

L’environnement est une image en niveaux de gris. L’agent connait la valeur du pixel courant, il sait aussi si le pixel sur lequel il se trouve à déjà été visité ou non. Nous n’avons pas fait de calcul de GVF.

### Agents

Nous avons trois agents différents :

* Agent d’exploration

Cet agent perçoit les pixels autour de lui (3x3) et calcule à chaque position le gradient (méthode de Kirsch) selon les 8 directions.

* + Il se déplace selon une direction et ne change pas de direction avant d’arriver au bord de l’image.
  + S’il trouve un contour potentiel, il crée un agent suivant un contour à la position du contour potentiel.
* Agent suivant un contour

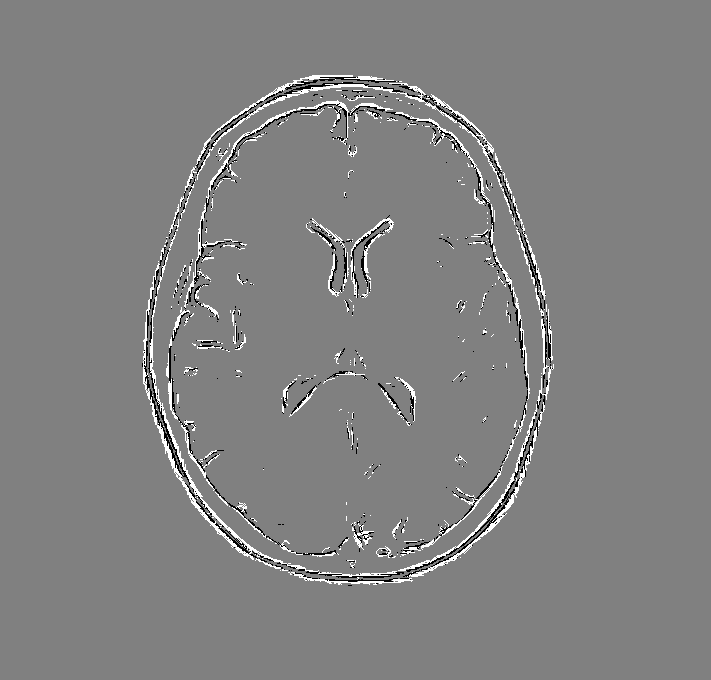
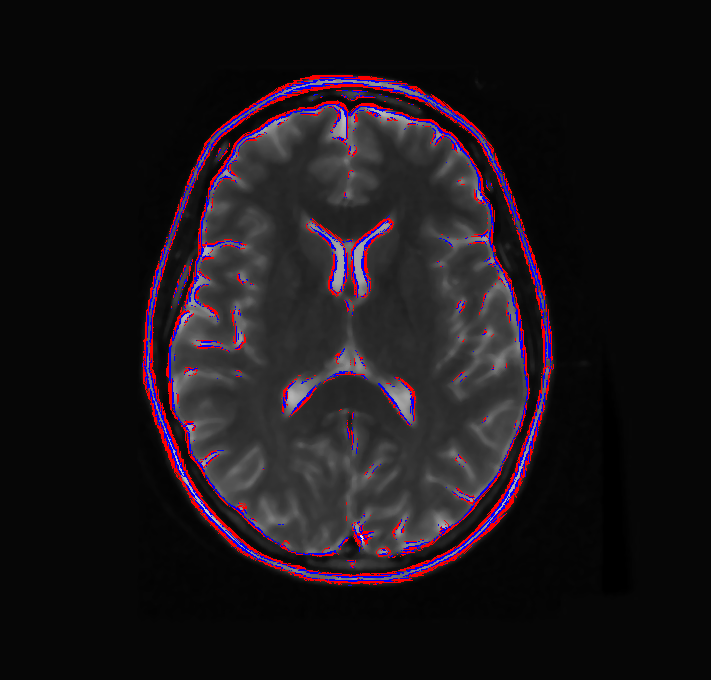
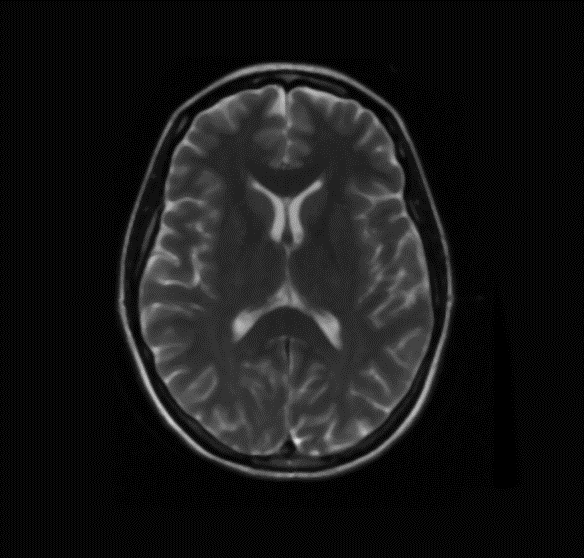
Cet agent perçoit les pixels autour de lui (3x3) et calcule à chaque position le gradient (méthode de Kirsch) selon les 8 directions.

* + Il se déplace dans la direction orthogonale au gradient le plus élevé.
  + Si l’agent se déplace sur un pixel déjà visité il disparait en créant un agent nœud.
* Agent nœud



|  |  |
| --- | --- |
| (a) | (b) |
| (c) | (d) |

Figure 2. (a) input image of a patient representing his lungs; (b) Mahdjoub et al. result; (c) our result without input image; (d) our result superposed to input image



1. (b) (c)

Figure 3. (a) input image of a brain MRI; (b) edge detection result; (c) result superposed to original image

1.  (b) (c)

Figure 4. (a) input image of a brain MRI; (b) edge detection result; (c) result superposed to original image

# Travaux cités

[1] Jason Mahdjoub, Zahia Guessoum, Fabien Michel, Michel Herbin. A multi-agent approach for theedge detection in image processings. 4th European Workshop on Multi-Agent Systems - EUMAS’06, Dec 2006, Lisbonne, Portugal. hal-02404201