

# SMA et Traitement d'images

## Résumé

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 simplifiée du SMA présenté par MAHDJOUR et al. (2006) [1]. Le système se base sur des agents réactifs détectant des contours.

**Keywords :** Système multi-agent, traitement d'image, détection de contours, agents réactifs

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

Ilias Maoudj ([i5maoudj@enib.fr](mailto:i5maoudj@enib.fr))



## Table des matières

Résumé.....	1
Table des matières .....	2
Table des figures.....	2
Travaux précédents .....	3
Environnement.....	3
Agents.....	3
Notre travail .....	4
Environnement.....	4
Agents.....	4
Le programme .....	5
Résultats .....	6
Pistes à explorer .....	7
Travaux cités.....	9

## Table des figures

Figure 1. Deux perceptions de la même image. Le carré bleu représente l'agent et le cercle sa perception (tiré de [1]) .....	3
Figure 2. Agent d'exploration (3x3 pixels noirs), Agent suivant un contour (1x1 pixel blanc) et Agent noeud (3x3 pixels blancs) évoluant dans l'environnement.....	5
Figure 3. (a) image entrée représentant une IRM des poumons d'un patient ; (b) résultat obtenu par Mahdjoub et al. ; (c) notre résultat ; (d) notre résultat superposé à l'image d'entrée ; (e) une imagerie du résultat obtenu ; (f) une imagerie du résultat superposé à l'image d'entrée. ....	6
Figure 4. (a) Image d'une IRM du cerveau ; (b) notre résultat ; (c) notre résultat superposé à l'image originale.....	7
Figure 5. (a) Image d'une IRM de la tête ; (b) notre résultat ; (c) notre résultat superposé à l'image originale.....	7
Figure 6. Imagerie d'un résultat comportant une détection imprécise du contour .....	7
Figure 7. Imagerie d'un résultat comportant des artefacts .....	8

## Travaux précédents

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

1. Pré-traitement de l'image
  - a. Filtre de Nagao permettant de réduire le bruit tout en conservant les informations sur les contours.
  - b. 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.

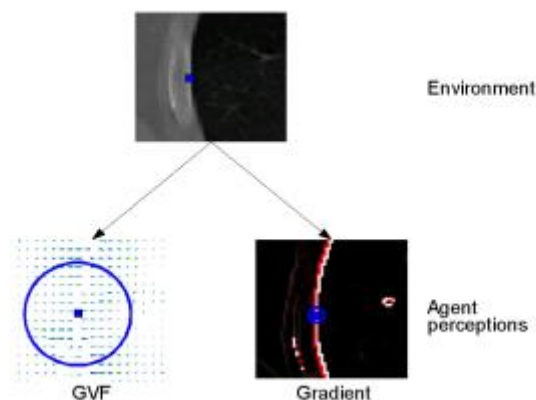


Figure 1. Deux perceptions de la même image. Le carré bleu représente l'agent et le cercle sa perception (tiré de [1])

## Agents

Quatre agents différents cohabitent :

1. Agent d'exploration.
  - a. L'agent parcourt l'image à la recherche d'un contour.
  - b. S'il détecte un contour, il crée un agent suivant le contour.
  - c. Si l'agent ne trouve aucun contour avant un temps défini, il disparaît.
2. Agent suivant un contour.
  - a. L'agent établit toutes les directions possibles du contour
  - b. Si l'agent trouve plusieurs directions, il se transforme en agent nœud.
  - c. 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.

- d. Si l'agent ne trouve pas de direction correcte, il se transforme en agent fermant un contour.
3. Agent fermant un contour.
  - a. 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.
  - a. L'agent nœud négocie avec les agents suivant un contour pour établir les segments définissant le contour.

## Notre travail

### Environnement

L'environnement est une image en niveaux de gris. L'agent connaît la valeur du pixel courant, il sait aussi si le pixel sur lequel il se trouve a 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 disparaît en créant un agent nœud.

- Agent nœud

Cet agent communique avec les autres agents nœuds présents dans l'image.

- Il envoie un message aux autres en indiquant sa position.
- Il reçoit une réponse des autres suffisamment proche de lui, il détermine le plus proche de lui, trace un segment vers sa destination et meurt.
- Si l'agent ne reçoit aucune réponse après 250 cycles, il meurt.

Contrairement à Mahdjoub et al., seuls nos agents nœuds négocient avec d'autres agents nœuds. Les agents d'exploration et les agents suivant un contour ont un comportement réactif.

## Le programme

L'appel du programme se fait sous la forme :

```
./progX Medic_C.ppm 10 -f 5
```

`./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.

Le programme ouvre 3 fenêtres :

- Environnement : L'image prétraitée (ou non) avec le visuel des agents se déplaçant.
- Résultat : Le résultat obtenu avec uniquement les contours détectés.
- Superposé : Le résultat superposé à l'image prétraitée.

Le programme se ferme lorsque 100% des pixels de l'image ont été explorés.



*Figure 2. Agent d'exploration (3x3 pixels noirs), Agent suivant un contour (1x1 pixel blanc) et Agent noeud (3x3 pixels blancs) évoluant dans l'environnement*

## Résultats

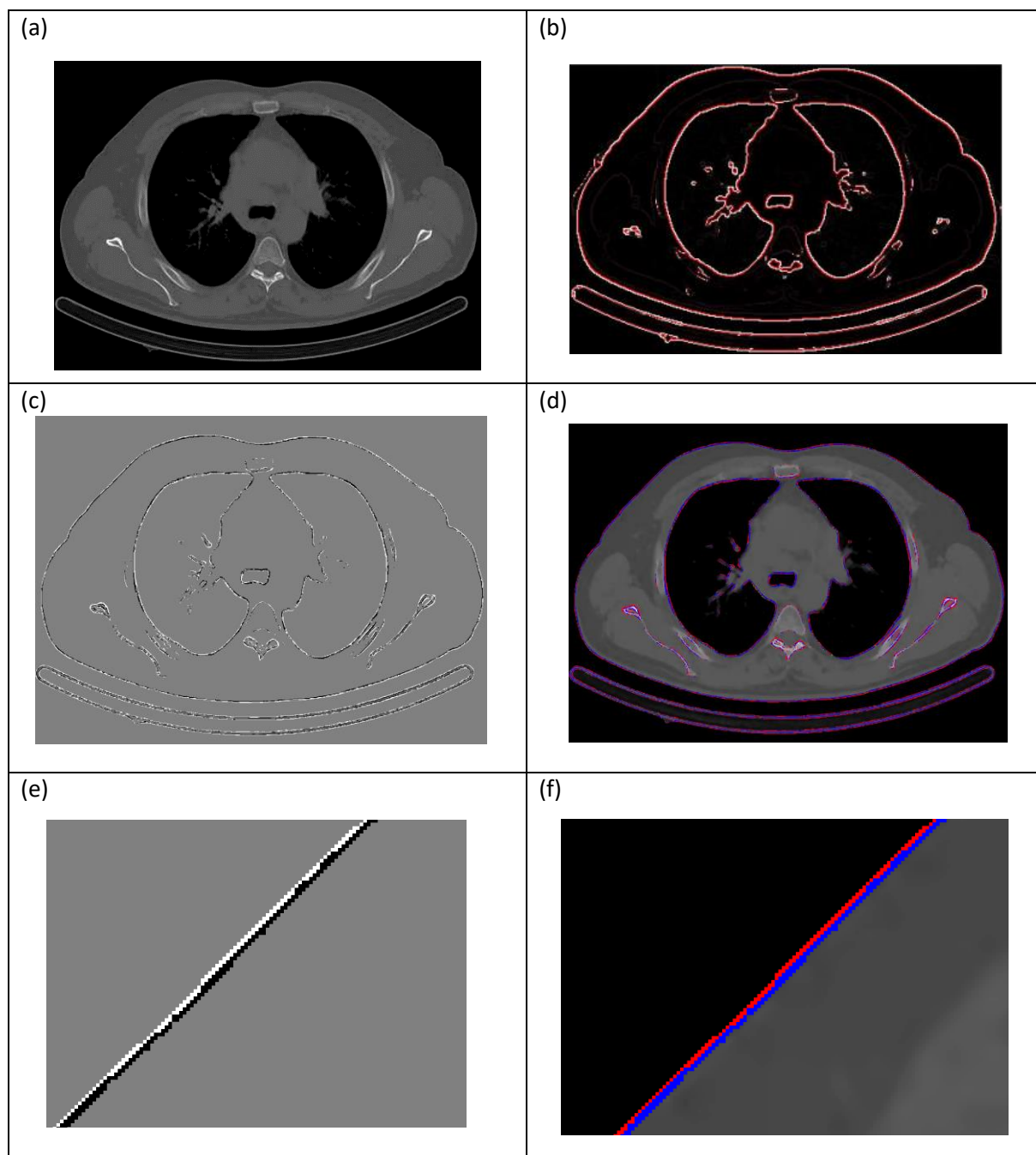


Figure 3. (a) image entrée représentant une IRM des poumons d'un patient ; (b) résultat obtenu par Mahdjoub et al. ; (c) notre résultat ; (d) notre résultat superposé à l'image d'entrée ; (e) une imagerie du résultat obtenu ; (f) une imagerie du résultat superposé à l'image d'entrée.

Les pixels blancs sur les images de résultat (c) et (e) représentent un gradient positif et les pixels noirs un gradient négatif. Dans les images (d) et (f), les pixels rouges et bleus représentent respectivement un gradient positif et négatif. Les pixels verts sur l'image (d) représentent les segments établis par les agents nœud.

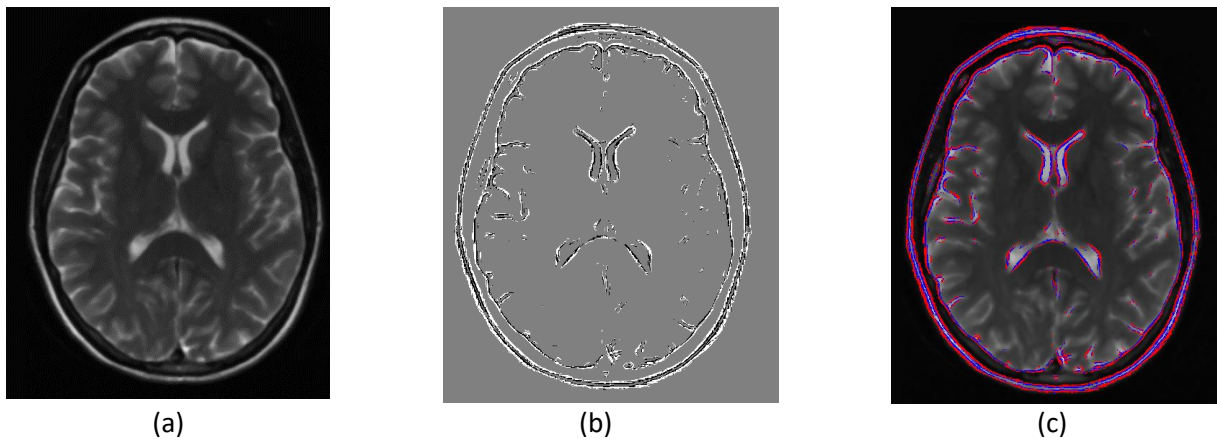


Figure 4. (a) Image d'une IRM du cerveau ; (b) notre résultat ; (c) notre résultat superposé à l'image originale

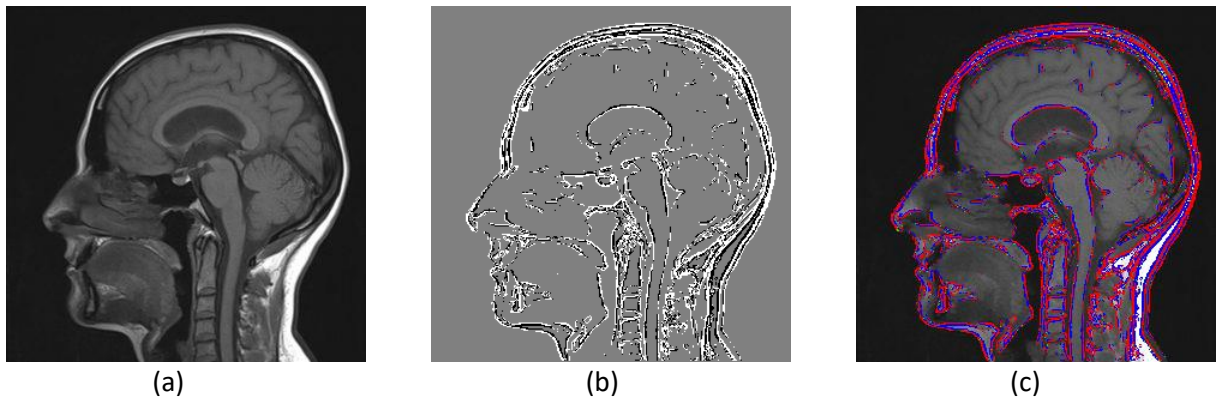


Figure 5. (a) Image d'une IRM de la tête ; (b) notre résultat ; (c) notre résultat superposé à l'image originale

## Pistes à explorer

Pour le moment, les seuils de détection des contours sont écrits en dur dans le code. Une idée est d'exploiter l'histogramme perçu localement par l'agent pour avoir une détection adaptable.

De même la détection est parfois imprécise. Il est encore possible d'améliorer la détection pour avoir un résultat plus fin (c'est-à-dire éviter d'avoir un tracé du contour « épais » avec 2 lignes de pixels) et plus précis (c'est-à-dire éviter les lignes de pixels vides entre le gradient positif et négatif).

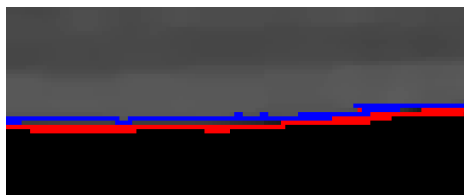
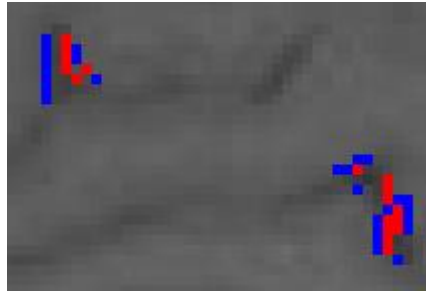


Figure 6. Image d'un résultat comportant une détection imprécise du contour

Un autre axe d'amélioration serait de ne retenir que les contours effectivement fermés et éviter de garder les artefacts



*Figure 7. Imagerie d'un résultat comportant des artefacts*



## Travaux cités

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