We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



18

Image Processing on MR Brain Images Using Social Spiders

Moussa Richard, Beurton-Aimar Marie and Desbarats Pascal
LABRI - University of Bordeaux
France

1. Introduction

Image segmentation is one of the most difficult tasks for digital image processing. This is an important step for quantitative analysis of images as brain images and so for studying many brain disorders. Indeed, structural changes in the brain can be the result of brain disorders. The quantification of these changes, by measuring area of regions of interest, can be used to characterize disease severity or evolution. The classical way of manual marking out of cerebral structures in MRI[1] images by an expert is obviously a time consuming process. Moreover, these manual segmentations are prone to large intra/inter-observer variability.
A lot of researchers have spent many years trying to solve this problem and proposed lots of methods. Unfortunately, these methods are dedicated to particular solutions. There is no generic method for solving the image segmentation problem. One main difficulty comes from that two types of noise are presented in medical images: physical noise due to the acquisition system, for example, Optical, X-rays and MRI, and physiological noise due to the patient status. Thus, the segmentation remains a challenging task. Image segmentation algorithms subdivide images into their constituent regions, with the level of subdivision depending on the problem to be solved. Robust, automatic image segmentation requires the incorporation and efficient utilization of global contextual knowledge. However, the variability of the background, the versatile properties of the target partitions that characterize themselves and the presence of noise make it difficult to accomplish this task. Considering the complexity, it is often required different methods in the segmentation process according to the nature of the images. In this work, we propose to use a MAS[2] to realize image segmentation and particularly, a MAS based on social agents.
A MAS is composed of heterogeneous unembodied agents carrying out explicitly assigned tasks, and communicating via symbols. On the contrary, many extremely competent natural collective systems of multiple agents (e.g. social spiders and social ants) are not knowledge based, and are predominantly homogeneous and embodied; agents have no explicit task assignment, and do not communicate symbolically. A common method of control used in such collective systems is stigmergy, the production of a certain behaviour in agents

[1] Magnetic Resonance Imaging.
[2] Multi-Agent System.

as a consequence of the effects produced in the local environment by previous behaviour (Wooldridge, 2002).
Ramos and Almeida have explored the idea of using a digital image as an environment for artificial ant colonies (Ramos & Almeida, 2000). They observed that artificial ant colonies could react and adapt appropriately their behaviour to any type of digital habitat. Ramos et al. investigated ant colonies based data clustering and developed an ant colony clustering algorithm which he applied to a digital image retrieval problem. By doing so, they were able to perform retrieval and classification successfully on images of marble samples (Ramos et al., 2002). Liu and Tang have conducted similar works and have presented an algorithm for grayscale image segmentation using behaviour-based agents that self reproduce in areas of interest (Liu & Tang, 1999). Hemernehl et al. have shown how an intelligent corpus callosum agent, which takes the form of a worm, can deal with noise, incomplete edges, enormous anatomical variation, and occlusion in order to segment and label the corpus callosum in 2D mid-sagittal images slices in the brain (Hemernehl et al., 2001). Bourjot et al. have explored the idea of using social spiders as a behaviour to detect the regions of the image. The principle is to weave a web over the image by fixing silks between voxels (Bourjot et al., 2003) .
In the goal to provide a more generic method not dedicated to a specific noise, we have adapted a Multi-Agent model based on Bourjot's idea for implementing a new way of performing an image segmentation.

2. Context

The MAS is a distributed system composed of a group of agents, which interact between them through an environment. Agents are classified in two categories: cognitive and reactive. Cognitive agents have a global view of the environment, they know the task for which they work. Reactive agents only know a restricted part of their environment, they react to environmental stimuli and can modify this environment by adding or removing informations. But reactive agents do not communicate directly together and do not know the complex task for which they work: they have a restricted set of simple features and they only apply them (Chevrier, 2002).
In biology, many natural systems composed of autonomous individuals exhibit abilities to perform skilled complex tasks without overall control. They can be adapted to their environment to survive or to improve the collective functioning. This is the case of social insects colonies (Camazine et al., 2001) such as termites, ants (G. Theraulaz & Spitz, 1997) or spiders (Gleizes & Marcenac, 1999) which are actually an evidence of remarkable abilities to perform collaborative tasks such as: construction of complex nests, bridge construction, efficiently resources research and capturing prey. The study of collective movements of migratory flocks of birds or fish stocks also shows that the collective task is the result of interactions between autonomous individuals (Theraulaz & Spitz, 1997a). The immune system is also a representation of a complex system operation composed of a set of autonomous agents (Ballet et al., 2004).
Insects colonies like spiders ones are groups of reactive agents: each one knows locally what it has to do, but no one knows the more complex task for which they work. Such insects are called social agents. Two behaviours can be found in social agents system: cooperation and competition. Agents can cooperate to perform their task through the environment. They can also be in competition with other agents to be the first to perform their own task.

Social spiders belong to natural spider species whose individuals form relatively long-lasting aggregations. Whereas most spiders are solitary and even aggressive toward con-specifics, hundreds of species show a tendency to live in groups and to develop collaborations between each other, often referred to as colonies. For example, as shown in figure 1, such spiders of 5mm in body length are capable to fix silks up to a volume of $100m^3$ (Jackson, 2007). This technique is used to trap big size preys.



Fig. 1. Social spiders web silking.

Social spiders have been defined by the biologists to present stigmergic process (Theraulaz & Spitz, 1997b). The characteristics of these societies and the importance of the silk in the various behaviour have created a different model from the social insects one. During their artificial cycle, social spiders have the abilities to perform several actions as: `Fix A Silk`, to `Move Forward` and to `Move Backward`. This model have characteristics which sufficiently distinguishes the levels of the realized spots, the society organization and the communication supports. Indeed, social spiders correspond to an interesting model for three reasons (Chevrier, 2002):

1. Social spiders do not present any specialization in morphology and ethology;
2. An isolated social spider presents behavioural characteristics very close to lonely species;
3. Social spiders show spectacular organization and cooperation forms, in particular, the web construction and the prey capture or its transportation phenomenon.

These are the reasons why this model can be easily implemented in computer programs and used to achieve complex tasks. Before presenting the social spiders segmentation method, we will explore the commonly used approaches to segment images.

3. Image segmentation

On the first hand, we will present the different categories of image segmentation methods and particularly the two ones used for the evaluation tests: Region Growing (Shapiro & Stockman, 2001) which is a region-based method and Otsu (Chen et al., 2001) which is a voxel-based method in order to compare them with our method. Then, on the second hand, we will discuss the noise problem when attempting to segment an image.

3.1 Classical approaches

Image segmentation consists on partitioning an image into a set of regions that covers it. After this process, each voxel is affected to a region and each region corresponds to a part of the image. The discontinuity between the regions constructs the contour of the object. The segmentation approaches can be divided into three major classes (Pham et al., 2000). The first one corresponds to voxel-based methods which only use the gray values of the individual voxels. The second one is the edge-based methods detect edges, for example, this can be done by computing a luminacy function. The last one, the region-based methods which analyze the gray values in larger areas for detecting regions having homogeneous characteristics, criteria or similitude. Finally, the common limitation of all these approaches is that they are based only on local information. Sometimes, a part of the information is necessary. Voxel-based techniques do not consider the local neighbourhood. Edge-based techniques look only for discontinuities, while region-based techniques only analyze homogeneous regions.

3.1.1 Region Growing method

The *Region Growing* method consists on building a region from one chosen voxel and then adding recursively neighbours whose grayscale difference with the original voxel is below a threshold (Shapiro & Stockman, 2001).

This method tries to grow an initial region by adding to this region the connected voxels that do not belong to any region. These voxels are the neighbourhood voxels already in the region and whose grayscale is sufficiently close to the area. When it is not possible to add voxels, we create a new region with a voxel that has not been selected yet, then we grow the region.

The method ends when all the voxels were chosen by a region.

3.1.2 Otsu method

Otsu has developed a multi-level thresholding method (Otsu, 1979) . Its aim is to determine, for a given number of regions, the optimum values of different thresholds based on the variance of subdivisions created.

The basic method consists on separating the foreground from the background. In this case, we search the optimal threshold to split the voxels in two regions. For a threshold t, it is possible to compute *the between-class variance* σ^2(t). This measure is derived from the average intensity μ_1, μ_2 and μ of Regions [0; t], [t + 1; L] and [0; L] where L is the maximum intensity.

The Equation 1 introduce the computation of σ^2, where w_1 and w_2 represent the proportion of voxels in the class [0; t] and [t + 1; L] compared to the total number of voxels.

$$\sigma^2(t) = w_1(t)(\mu_1(t) - \mu)^2 + w_2(t)(\mu_2(t) - \mu)^2 \tag{1}$$

The Otsu method shows that the optimal threshold t^* is obtained for a between-class variance. The method consists on computing the variance for all possible thresholds (t $\in$ {1; . . .; L - 1}) and determining its maximal value.

This method could be extended easily to the computation of M Regions with M - 1 thresholds $\{t_1; t_2; \ldots; T_{M-1} - 1\}$ ($t_1 < t_2 < \ldots < t_{M-1}$). The between-class variance is defined then as follows:

$$\sigma^2(t_1, \ldots, t_{M-1}) = \sum_{M-1}^{M} w_k(\mu_k - \mu)^2 \tag{2}$$

where w_k represent the proportion of voxels in the class $[t_{k-1}; t_k]$[3], μ_k the intensity average of this same Region and μ the intensity average of the class [0; L].
For each M-1-uplet, we compute thresholds of the between-class variance. The optimal thresholds, $(t^*, \ldots, t^*_{M-1})$, correspond to the maximum value of the between-class variance.
Chen et al. propose an algorithm that minimizes the number of necessary computation to obtain a faster algorithm (Chen et al., 2001). This method had been implemented for our evaluations tests.

3.2 Noise problem

Noise in biological images is still a major issue for the segmentation process. Two major contributions exist to explain medical imaging noise: the acquisition noise and the physiological noise. The acquisition noise can be identified as and sometimes reduced to physical noise meaning that it depends on the physical method of acquisition (Hendee & Ritenour, 2002; Vasilescu, 2005), for example, MRI noise caused by MRI machines. For the physiological noise, there are three main contributions : the respiratory cycle, the cardiac cycle and the scattered noise mainly due to physiological liquid action. Some solutions have been proposed intrinsically combined with the acquisition process. Another way is to use external monitoring to keep track of the cycles themselves and to reconstruct the cycle mainly using low order polynomial function. This function could be used to aggregate liquid noise regions into one in the post-treatment process (Bankman, 2000). Medical images exhibit all these types of noise characteristics. These characteristics need to be taken into account for all image processing. One important image processing operation is image segmentation. Image segmentation is often the starting point for other processes, including registration, shape analysis, motion detection, visualization, quantitative estimations of linear distances, areas and volumes. In these cases, segmentation involves categorizing voxels into object regions based on their local intensity, spatial location, neighbourhood, or shape characteristics. But, methods are often optimized to deal with specific medical imaging modalities such as MRI, or modelled to segment specific anatomic structures such as the brain, the liver and the vascular system. It is clear that a single segmentation technique is not able of yielding acceptable results for all different types of medical images. Still, the simplest way to obtain good segmentation results is segmentation by hand. This segmentation requires expertise in the field, in our case, a doctor is the person concerned. However due to the huge number of images to treat and the complexity of 3D images, it is not acceptable to consider that segmentation by manual contour tracing is an efficient solution. So, automatic image segmentation appears to be the right way to perform this task.

4. Social spiders model

The multi-agent system is composed of an environment and a set of agents. For segmentation purpose, the environment is created from a given grayscale image: it is a matrix of gray voxels. System and agents have a life cycle. Figure 2 shows that a cycle of the system consists in executing the three behaviours of each agent. This life cycle is transposed to a step. The number of steps to be executed is given by the user. Algorithm 1 presents a description of figure 2 where for a given number of time steps, each spider computes its life cycle. Its

[3] $t_0 = 0$ and $t_M = L$.

complexity is about $O(Steps.N)$, where $Steps$ is the number of time steps and N the number of spiders.

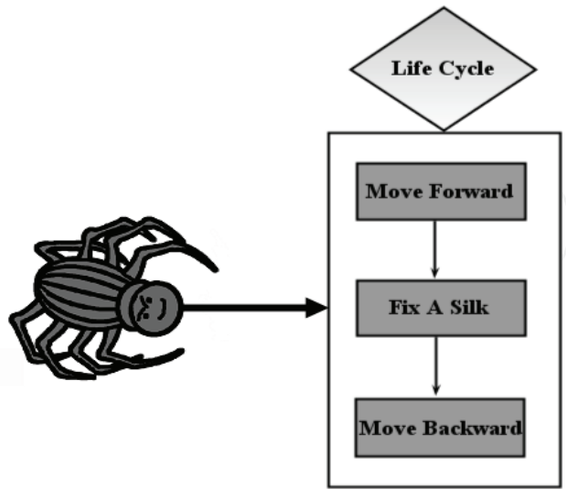


Fig. 2. System overview.

Algorithm 1 Segmentation method

Require: Voxels: Matrix of Voxels $\in \mathbb{N}^3$, Steps $\in \mathbb{N}$ and Conf: Configuration parameters

```
Create colonies and spiders from the Conf.
while Steps - - > 0 do
    for Each spider S do
        MoveForward(S).
        FixASilk(S, Voxels).
        MoveBackward(S).
    end for
end while
```

4.1 Environnement

The environment is composed of gray voxels. Each voxel is a position for spiders and allows them to access to other voxels. For a spider, the neighborhood of a voxel V which is coloured in blue in figure 3 is defined in two ways:

- The voxels around V, which are named *Local Neighbourhood* coloured in gray except those in green;
- And all voxels linked to V by a silk, which are named *Silked Neighbourhood* coloured in green.

Set of all reachable voxels are named *Access*.

4.1.1 Spider agent

As previously mentioned, spiders are reactive agents. They are defined by an internal state composed of a set of parameters values, a current position and the last voxel where a spider has silked. These spiders have three behaviour abilities:

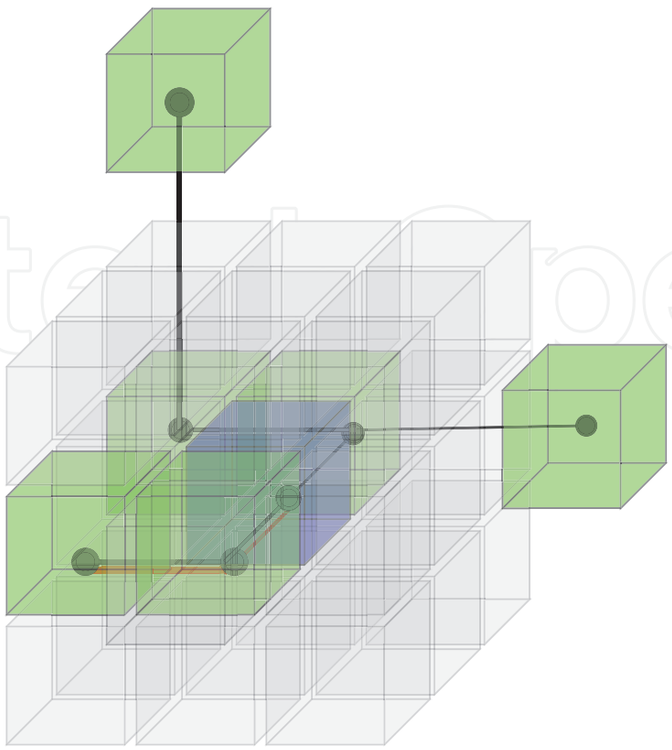


Fig. 3. Neighbourhood of a voxel.

1. Move Forward;
2. Fix a Silk[4];
3. Move Backward[5].

Spiders are grouped in a set called a colony. Spiders of a same colony share the same set of parameters values. Their aim is to detect the same region. For doing that, they communicate locally via silks.
An agent life cycle, as shown in figure 2, consists in firing each behavioural item according to a probability which depends of parameters values and environmental characteristics at the spider position.

4.1.1.1 Move forward

Moving Forward is computed according the way where there are several colonies trying to detect multiple regions. Each spider reaches one neighbour and reacts on it as described in figure 3. A weight function is defined to compute the probabilities of the *Access* voxels in order to select the best voxel to move to. So, for each voxel V in the neighborhood, the probability to move to V is as below:

$$P(Move(V)) = \frac{W(V)}{\sum_{a \in Access} w(a)} \tag{3}$$

[4] Weave a dragline between two voxels.
[5] Return to the last fixed voxel.

$w(V)$ is constant if V is in the *Local Neighbourhood*, else if V is in the *Silked Neighbourhood*, we distinguish the drag-lines woven by its colony and those woven by other colonies. Two simulation parameters, *attractself* and *attractother*, are used to compute this probability. These parameters are respectively the attraction for drag-lines woven by its colony and drag-lines woven by other colonies. A function F is also used to express drag-lines-counts influence on weight. This function express the saturation of the voxel bounded by a parameter called *saturationvalue*. So, for a voxel V in the *Silked Neighbourhood*, we have the following weight :

$$\begin{aligned} self &= attractself \,.\, F(draglines_{self}) \\ other &= attractother \,.\, F(draglines_{other}) \\ W(V) &= self \,+\, other \end{aligned} \tag{4}$$

Algorithm 2 describes the movement function by affecting to the weights of the neighbourhood voxels whether the weight of the colony of the spider whether its weight function and finally move the spider to a voxel from the neighbourhood according to its weight value. Its complexity is $O(Nei)$ where Nei is the maximum of neighbourhood of a voxel.

Algorithm 2 Move Forward

Require: S: Spider, Voxels: Matrix of Voxels $\in \mathbb{N}^3$.
1: weights[Size(Access(position(S)))]: weights of the neighbourhood voxels.
2: **for** i $\in$ {0, . . ., Size(Access(position(s)) - 1)} **do**
3: weights[i] $\leftarrow$ WeightFunction(S).
4: **end for**
5: Choose a voxel from the neighbourhood according to its maximal weight and move the spider to the voxel chosen.

4.1.1.2 Fix A Silk

Here, two other parameters are used, *reflevel* and *selectivity* computed from the histogram of the image. The first one corresponds to the graylevel of the region to detect, and the second one defines the tolerance to fix a drag-line with a voxel whose graylevel is not exactly reflevel. Probability to fix a drag-line with current voxel (here, the blue voxel in figure 4 follows a Gaussian distribution whose mean is *reflevel* and standard deviation is *selectivity*). Algorithm 3 consists on choosing a random number and evaluating it with a Gaussian function to add or not a drag-line. Its complexity is constant.

Algorithm 3 Fix A Silk

Require: S: Spider, Voxels: Matrix of voxels $\in \mathbb{N}^3$.
1: p $\leftarrow$ random(0, 1).
2: **if** p $<$ Gauss(level(position(S))) **then**
3: Add a drag-line between postion(S) and lastfixed(S).
4: **end if**

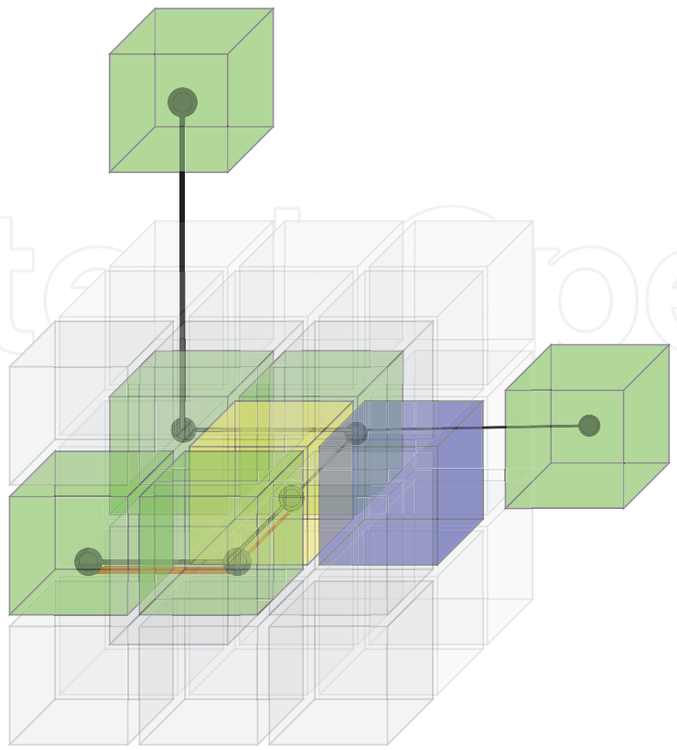


Fig. 4. Decision to fix a dragline with current voxel.

4.1.1.3 Move backward

This behavioural item allows spiders to come back to the last silked voxel (here, the yellow voxel in figure 4). This action is fired depending on the probability value defined as *backprobability*.
The aim of this action is to detect connected region of voxels: spiders can not go far of voxels they have silked. As it is shown by (Bourjot et al., 2003), disable this item leads to create drag-lines between two unconnected groups of voxels. Algorithm 4 test the possibility to return to the last fixed drag-line if the condition is satisfied in ligne 2. Its complexity is also constant.

Algorithm 4 Move Backward

Require: S : Spider
1: p ← random(0, 1).
2: **if** p < backprobability(colony(S)) **then**
3: position(S) ← lastfixed(S).
4: **end if**

4.2 Simulation process

Simulation process is based on stigmergy: each spider lets informations on the environment, those are used by other agents or by itself in a next cycle. Image segmentation emerges from the global task achieved by all the spiders: after a certain number of system time steps,

drag-lines are created. Degree of a voxel defines number of drag-lines coming in or out of this voxel. In our case where there is a multiple regions detection, degree is given according to a colony. Global degree is the sum of degree of each colony. Region detected by a colony is composed of voxels having higher degree for this colony.

4.3 Problems

The social spiders method raises two problems:

1. There is a few number of parameters to fix;
2. Computing the number of time steps required or defining a stop condition.

Fixing the parameters can be problematic if it is done empirically. Indeed, to compare the results of social spiders segmentation among several images, we must be sure that the computation of the parameters will be in an equivalent manner in all cases.
Similarly, it is important that the stop condition meets the same criteria between different acts of segmentation. Otherwise, the results could be evaluated: a number of time steps not big enough leads to a poor qualitative analysis, and a number of time steps too high could increases the simulation time of the method without improving the quality of the results.

4.4 Solutions

First of all, we define the set of parameters to be fixed and its usage in the program. Then, we propose some ideas to reduce their number. Finally, we propose solutions to the two problems evaluated above.

4.4.1 Auto-detection of parameters

The method of social spiders has several parameters that will define the regions to be detected. These parameters are presented in Table 1.

Parameter	Description
reflevel	Grayscale of the colony
selectivity	Acts on the probability to weave a dragline from the current position of the insect
attractself	Attracts toward the draglines of the same colony
attractother	Attracts toward the draglines of the other colony
backprobability	Probability to turn back
$w(v)$	Weight of a voxel with a direct neighbourhood
saturationvalue	Upper bound of the weight computed inside a voxel

Table 1. Colony parameters.

First, it must know the number of colonies (the number of type of regions) and each of the properties, in particularly the grayscale reference and the standard deviation parameter. We have seen that the computation of these empirical parameters could be an obstacle for comparing the results produced by the social spiders method: if the parameters are not computed in the same way in each case, the comparison became unreliable because they

depends on the computation method. We will see in this part a method for determining the simulation parameters. This method will be used thereafter in all segmentation processes.
Each colony has a grayscale (or intensity) of reference which will serve the spiders of the colony to determine if they should fix a silk. There is two possibilities:

1. Determining "manually" the intensities, which is unattractive except where the intensities are known by the user;
2. Using a method to determine automatically the intensities.

The automatic detection will allow us to obtain optimal parameters without the user knowledge about the image. Thereafter, we will appoint an intensity level of voxels of this intensity.
We will use the histogram of the image to determine the intensity of interest to be detected. Indeed, a region inside an image usually implies a peak more or less important in the histogram of the image, but one can note that this histogram is particularly sensitive to noise. Therefore, we will introduce a method to reduce the noise effects on the histogram by smoothing it and allowing to obtain a maximum number of representative of the number of Regions intensities of the image.

Histogram smoothing

Histogram smoothing will reduce the peaks caused by noise. The method proposed here is done by time step. At each time step, the degree of intensity n becomes the average degrees of intensity n - 1, n, n + 1. We propose a method that detect automatically the optimal settings for the segmentation of the image. The method determines the number of settlements and the parameters *reflevel*, *selectivity* and the number of spiders in each colony.
First, we will determine the maxima in the histogram as described above. Then, the histogram will be smoothed until fewer maxima are significant. Indeed, the maxima caused by noise are on the peaks of the histogram whose slopes are small and will be erased by smoothing them in few time steps. Finally, the maxima representing a region is located on a peak having an important slope which requires a large number of smoothing in order to be eliminated. However, the risk to clear up an interesting maxima is not zero.
In order no to fall into aberrations, we add the condition that the number of maxima, and therefore the number of regions to detect, must be fixed by the user which is often a well known parameter. For example, The number of regions to detect in the brain is 10.
We get a series of maxima M_1, M_2, . . ., M_k. The number of maxima, k, determines the number of spiders colonies that will be used. Each parameter is the maximum *reflevel* of the colony. Then, we will partition the intensities in as many Regions as maxima detected. To do this, simply find the minimum level of intensity between two maxima. This yields a series of minima m_1, m_2, . . ., m_{k-1} so that $0 < M_1 < m_1 < \ldots < m_{k-1} < M_k <$ L where L is the maximum intensity. The k Regions intensities obtained are [0; m_1],]m_1; m_2], . . .,]m_{k-1}; L].
The variance of each class provides the parameter *selectivity* of each column. The number of spiders per colony is the same for all colonies.
We can now determine the main parameters of the social spiders method. It remains to determine the numbers of spiders per region and the attraction of the silks on spiders.

4.4.2 Stop condition

We have seen that the social spiders method has a life cycle which is repeated a number of times until the image is segmented. The number of time steps will influence two important points on the result of segmentation:

1. The quality of segmentation;
2. The execution time required to achieve this result.

At each time steps, the spiders will weave between the voxels that will be used to determine to which region belongs the voxel. If the number of time steps is not enough, the number of voxels that do not belong to any region will be important and the result will be of poor quality. On the contrary, if the number of time steps is too large, the spiders will only increase after a certain time the silks already existing without providing any new information. This last point will have as effect a longer execution time for an approximatively identical quality result.
As for the parameters, the computation of the number of time steps is an important point to obtain comparative results that are credible. Rather than fixing a number of time steps, it is possible to determine a stop condition to be verified before every time steps.

Definition Let β be the number of silks fixed between voxels whose degree is zero during a time steps.

The result of socials spiders segmentation depends on the silk which will be fixed between the voxels. During a time steps, when β tends towards zero, we can consider that the system stabilizes and the spiders only reinforce the existing silks.
It is possible to detect when β remains at zero. This moment determines to stop the simulation. We can improve this condition by adding two parameters:

- A threshold that determines when β may be considered invalid;
- The number of authorized β zero before stopping the simulation.

The threshold may be determined by the number of spiders. Indeed, at each time steps, each spider has the possibility to fix a silk. The number of silks fixed during a time steps is bounded by the number of spiders.

5. BrainWeb application

In this section, we present the model generated by Brainweb[6] and its various components. We introduce the validation steps and the comparison technique used for the evaluation. Finally, the results are exposed and discussed.

5.1 BrainWeb

Brainweb is a Simulated Brain Database which contains a set of realistic MRI data volumes produced by an MRI simulator. These data can be used by the neuroimaging community to evaluate the performance of various image analysis methods in a setting where the truth is known. BrainWeb offer two anatomical models: normal and multiple sclerosis. For both of these, full 3D data volumes have been simulated using three sequences (T1-, T2-, PD- weighted) and a variety of slice thicknesses, noise levels, and levels of intensity

[6] http://www.bic.mni.mcgill.ca/brainweb/.

non-uniformity. As we are interested by the noise levels, The noise added by BrainWeb has Rayleigh statistics[7] in the background and Rician statistics[8] in the signal regions. The "percent noise" number represents the percent ratio of the standard deviation of the white Gaussian noise versus the signal for a reference tissue.

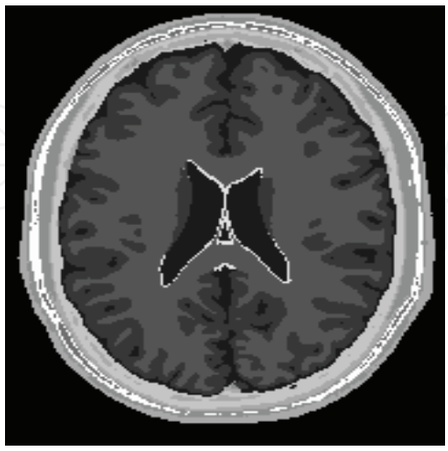


Fig. 5. Slice of the 3D model.

The parameters of the model to be used in this study are:

- Modality: T1
- Slice thickness: 1mm
- Noise: X%
- Intensity non-uniformity: 20%
- Dimensions: 181 x 217 x 181 voxels

Where X is the variability of the noise in the image. It will goes from 0% till 9%.
We can distinguish ten items in figure 5 are listed in Table 2.

5.2 Experimentation

In this part, we compare the social spiders method with other segmentation methods. These comparisons allow us to determine whether the social spiders method leads to a good segmentation compared to traditional segmentation methods:

- A classification method by thresholding: the Otsu method;
- A region-based method: the Region Growing method.

To compare these methods, we need to establish criteria to be used on all test images. We compare the results on several points:

1. The number of regions;
2. The pourcentage of voxels with correct labels before and after post-processing CL_b and CL_a between the truth model and the segmentation result.

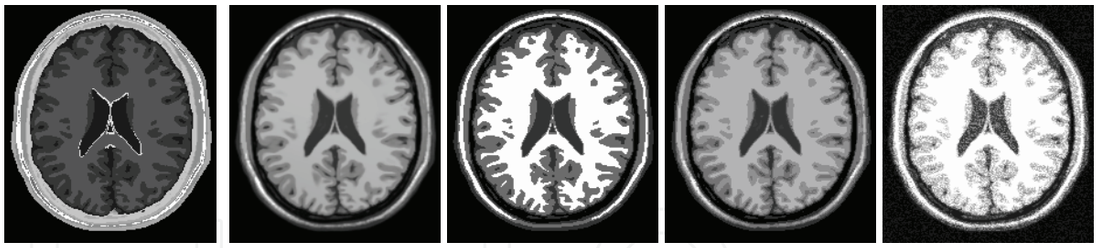
[7] http://en.wikipedia.org/wiki/Rayleigh_distribution.
[8] http://en.wikipedia.org/wiki/Rice_distribution.

Element	Pourcentage	Form
background	42.2%	
csf	5.2%	
grey matter	12.7%	
white matter	9.5%	
fat	2.1%	

Element	Pourcentage	Form
muscle skin	8.7%	
skin	10.2%	
skull	5.1%	
glial matter	0.1%	
connective	4.2%	

Table 2. BrainWeb Model: the various components.

The number of regions allows us to determine whether the method considered detects a number of regions close to reality. It is possible that classical methods detect regions with insignificant size. That is why we add to the total number of regions, the number of regions having insignificant size. For the classical methods of segmentation, we consider a region as insignificant if its size is less than 10 voxels.
The computation of the number of regions is done on the segmentation method result on which a labelling is added to the connected components to consider the regions connected.
The execution time to be given comes from the simulation of the methods on a machine equipped with an Intel Quad Q9550 (4 cores having 2.83GHz) and 4GB of RAM. The operating system of this machine is a Linux kernel 2.6.21 x86_64. The brain images are composed of 10 regions which correspond to the structure of the brain. The size of the test volumes is 181x217x181 voxels. The slice presented in the screen capture here is the 94th slice of the 3D image. Furthermore, for the three methods compared here, the parameters which were empirically set are those which have produced the best results.
Figures 6, 7, 8, 9, 10 and 10 present the results of the different image segmentation techniques applied on a brain image with different levels. Their informations are explored in table 3.
As we can see from the results in table 3 that the Region Growing method (threshold = 406) has the lowest execution time, the lowest CL_a and the biggest number of regions. the Otsu method (thresholds = 406, 1175, and 1990) has produced a result similar to Region Growing with less number of regions. The social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) has an important execution time with an additional region that corresponds to the extra voxels that were not detected by any spider. This region is not connected, the voxels that compose it are scattered throughout the image. It is therefore possible to perform a post-processing that would link these voxels to the colony that have a strong presence in their neighborhoods.

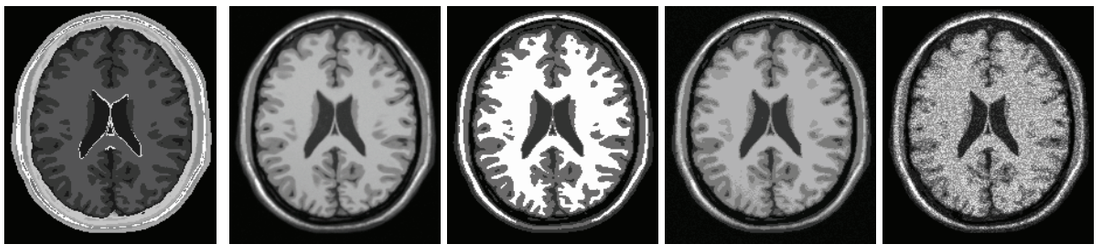


(a) (b) (c) (d) (e)

Fig. 6. 3D segmentation of Brain image without noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

Despite the fact that the difference between region-based segmentation methods is small, CL_a of the result of the spiders is the best. However, as the spiders methods is a stochastic method, we do not expect to get maximum CL_a. Let us test that this CL_a will remain stable when adding noise.

For that, we added noise to the original images. This noise added by brainweb has Rayleigh statistics[9] in the background and Rician statistics[10] in the signal regions. The "percent noise" number represents the percent ratio of the standard deviation of the white Gaussian noise versus the signal for a reference tissue. The results and the statistics of the different image segmentation techniques applied on the brain noisy image are presented below with different noise levels.



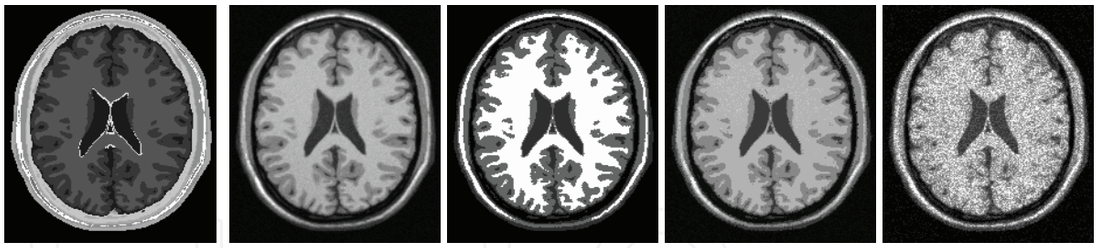
(a) (b) (c) (d) (e)

Fig. 7. 3D segmentation of Brain image with 1 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

For 1% of noise, the Region Growing (threshold = 410) method has produced an oversegmentation with an important decrease of the CL_a and the best time to process. The Otsu (thresholds = 410, 1168 and 1966) method has lightly oversegmentated the image despite the fact that its CL_a has not decrease a lot when adding the 1% of noise. The CL_a of the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) has increased a little bit and remained stable as the best one in term of performance. This correlation is the result of a big computation time.

For 3% of noise, the Region Growing (threshold = 420) method continues to make an oversegmentation with an improvement of the CL_a and a stability of the time to process. This improvement is the result of a good post-processing. The Otsu (thresholds = 420, 1123 and

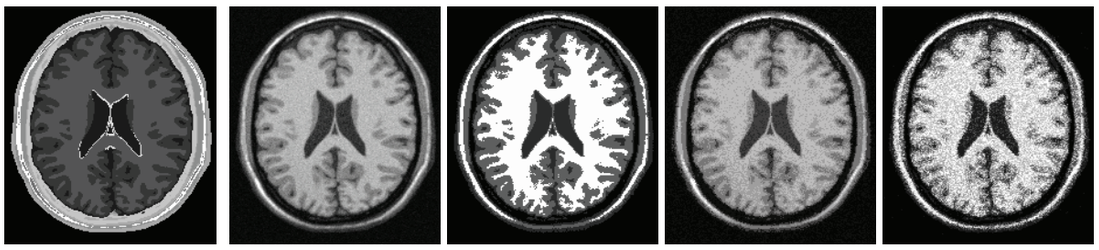
[9] http://en.wikipedia.org/wiki/Rayleigh_distribution.
[10] http://en.wikipedia.org/wiki/Rice_distribution.



(a) (b) (c) (d) (e)

Fig. 8. 3D segmentation of Brain image with 3 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

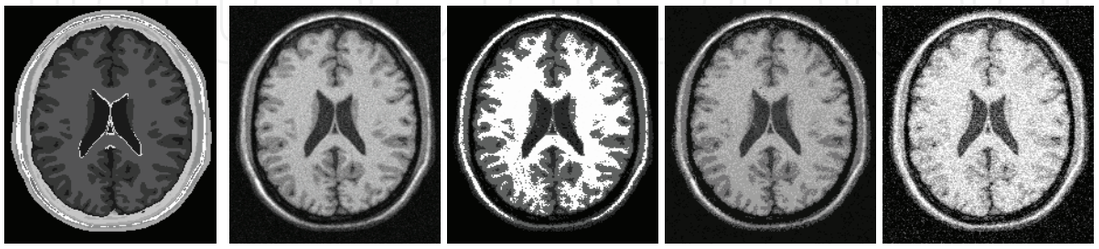
1889) method begin to be in competition with the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) by being very close in term of CL_a. The time to process of these two methods have made a small variation with the best time for the Otsu method.



(a) (b) (c) (d) (e)

Fig. 9. 3D segmentation of Brain image with 5 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

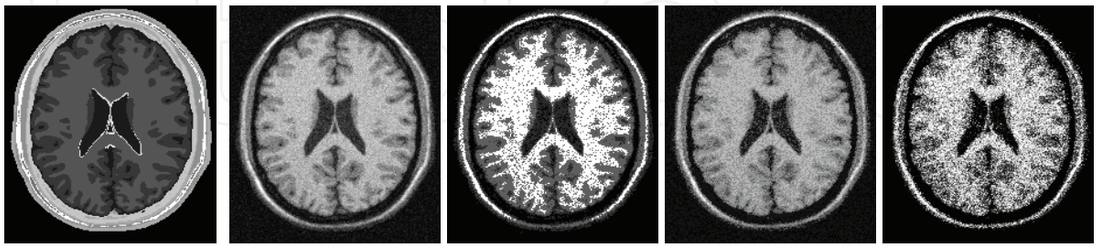
For 5% of noise, the Region Growing (threshold = 442) method has decreased the less between the three methods having always done an oversegmentation and remaining the worst. The Otsu (thresholds = 442, 1099 and 1823) method remains in challenge with the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) by continuing to be very close in term of CL_a.



(a) (b) (c) (d) (e)

Fig. 10. 3D segmentation of Brain image with 7 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

For 7% of noise, the Region Growing (threshold = 500) has decreased the lot with the oversegmentation remained due to the complexity of the image and the additional noise effect. The Otsu (thresholds = 500, 1155 and 1880) took the lead from the social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) with an effect on the post-processing which is done badly. This is mainly due to the non-selected voxels presented on the contour of the image.



(a) (b) (c) (d) (e)

Fig. 11. 3D segmentation of Brain image with 9 % of noise: a) Ground Truth Slice, b) Original Slice, c) Otsu thresholding, d) Region Growing, e) Social spiders.

% of noise		**Otsu**	**Region Growing**	**Social spiders**
0 %	**Regions**	917	70231	11
	Regions $>$ **10 voxels**	203	1006	11
	CL_b	67.91 %	68.02 %	70.51 %
	CL_a	67.90 %	67.70 %	69.13 %
1 %	**Regions**	1150	135093	11
	Region $>$ **10 voxels**	205	1609	11
	CL_b	67.90 %	59.90 %	71.23 %
	CL_a	67.89 %	60.03 %	69.51 %
3 %	**Regions**	2503	224653	11
	Region $>$ **10 voxels**	198	4890	11
	CL_b	67.17 %	57.63 %	67.96 %
	CL_a	67.16 %	64.51 %	67.62 %
5 %	**Regions**	9669	241482	11
	Regions $>$ **10 voxels**	209	3483	11
	CL_b	66.11 %	57.23 %	67.85 %
	CL_a	66.14 %	64.28 %	66.84 %
7 %	**Regions**	23138	249995	11
	Regions $>$ **10voxels**	317	4673	11
	CL_b	63.83 %	49.73 %	63.26 %
	CL_a	64.21 %	54.54 %	62.92 %
9 %	**Regions**	42392	186376	11
	Regions $>$ **10voxels**	330	2935	11
	CL_b	60.69 %	47.13 %	53.21 %
	CL_a	62.06 %	47.41%	55.02 %

Table 3. 3D results: Brain image with different levels of noise.

For 9% of noise, the Region Growing (threshold = 539) method continues to decrease the lot with also a decrease of the number of regions and the best time to process. The Otsu (thresholds = 539, 1172 and 1866) method has remained the best in term of CL_a while the

social spiders method (time steps = 1000, backprobability = 0.1, $attract_{self}$ = 0.4, $attract_{other}$ = 0.6 and saturation = 50) decreases a lot due to the important number of non-silked voxels. Therefore, as shown in figure 12, it appears that the social spiders segmentation is robust to noise effect from 1% to 5% to and could be better for the rest. It also shown that social spiders method is better the region growing method (p-value = 0.08983) and have conducted similar results to that done by the Otsu method. This robustness has however led to an additional region in the image with a big influence on time to process.

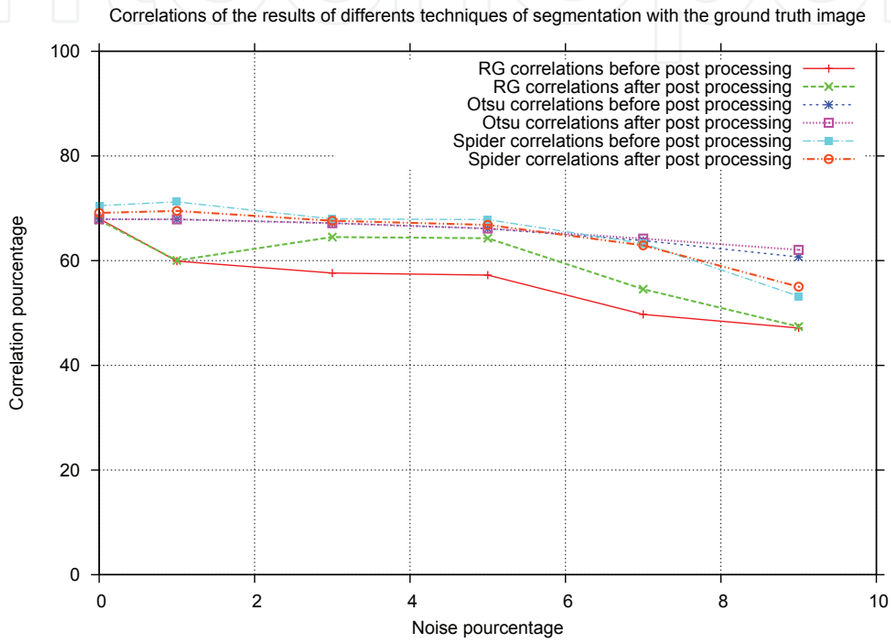


Fig. 12. Correlation of different segmentation techniques.

6. Conclusion

In this chapter, we have presented a new region-based method, the social spiders method. This method has been compared with the Region Growing Method and the the Otsu method. These comparisons focused on the nombre of voxels well labeled and the number of regions produced. The spiders method has produced a non neligeable time processing in the case of non-noisy image with a result better than the others. When noise is added, the processing time has increased a little bit but with a better result than the other region-based method and a similar result with the voxel-based method. Note that the results of social spiders method are influenced by the number of parameters to fix and particularly the repartition of the agents on the grid and the number of step to do. But automatic methods has been described to fix these parameters. Through these comparisons, we have put forward some drawbacks on the social spiders method. Particularly, we have seen that this method produced an additional region composed of the non selected voxels and that the execution time was particularly

long as discussed above. Some solutions have been considered to improve the method. The spiders seem sensitive to the topology of the image, so it is possible to guide the movement of spiders with a gradient or a laplacian. Indeed, these measures will provide informations on the possible presence of contours. It would be then possible to use other species like the social ants to guide the social spiders in order to cover all non-selected voxels.
However, the social spiders method is based on an architecture that is ideal to be parallel. This method is composed of a group of agents that can be spread over several processors. As GPU[11] allows us to share memory space betwenn processes. So, it is easy to use the social spiders program by putting the image to segment in the global memory of the GPU. First tests have shown that the processing time only depends from the number of steps fixed by the user. They also give back the simulation time to the same level of the other methods.

7. References

Ballet, P., Zemirline, A. & Marcé, L. (2004). The biodyn language and simulator. application to an immune response and e.coli and phage interaction, *JOURNAL of Biological Physics and Chemistry* 4: 93 – 101.

Bankman, I. (2000). *Handbook of Medical Imaging: Processing and Analysis*, Academic Press.

Bourjot, C., Chevrier, V. & Thomas, V. (2003). A new swarm mechanism based on social spiders colonies: from web weaving to region detection, *Web intelligence and Agent Systenms: An International JOURNAL - WIAS* 1(1): 47–64.

Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G. & Bonabeau, E. (2001). *Self-Organization in Biological Systems*, Princeton University Press, Princeton, NJ, USA.

Chen, T.-S., Liao, P.-S. & P.-C.Chung (2001). A fast algorithm for multilevel thresholding, *JOURNAL of Information Science and Engineering* 17: 713–727.

Chevrier, V. (2002). Contributions au domaine des systemes multi-agents, *Technical report*, Université Henry-Poincarré - Nancy 1.
URL: *http://www.loria.fr/ chevrier/Publi/HDR.zip*

G. Theraulaz, G. & Spitz, F. (eds) (1997). *Auto-organisation et comportements collectifs : la modélisation des sociétés d'insectes*, Editions Hermes, pp. 91 – 140.

Gleizes, M.-P. & Marcenac, P. (eds) (1999). *Coordination par le biais de l'environnement : une approche biologique*, Hermes, pp. 237–250.
URL: *http://hal.inria.fr/inria-00107820/en/*

Hemernehl, G., McInerney, T. & Terzopoulos, D. (2001). Deformable organisms for automatic medical image analysis, *Medical Image Analysis* pp. 66–76.

Hendee, W. & Ritenour, E. (2002). *Medical Imaging Physics*, Wiley and Sons.

Jackson, E. (2007). Social spiders, *Current Biology* 17(16): R650 – R652.

Liu, J. & Tang, Y. (1999). Adaptive image segmentation with distributed behavior-based agents, *IEEE Transactions Pattern Analysis and Machine Intelligence* 21(6): 544–551.

Otsu, N. (1979). A threshold selection method from gray level histograms, *IEEE Trans. Systems, Man and Cybernetics* 9: 62 – 66.

Pham, D., Xu, C. & Prince, J. (2000). A survey of current methods in medical image segmentation, *Annual Review of Biomedical Engineering*, Vol. 2, Annual Reviews, pp. 315 – 338.

[11] Graphical Processing Unit.

Ramos, V. & Almeida, F. (2000). Artificial ant colonies in digital image habitats - a mass behaviour effect study on pattern recognition, *Proceedings of ANTS'2000 - 2nd International Workshop on Ant Algorithms (From Ant Colonies to Artificial Ants), in Marco Dorigo, Martin Middendorf and Thomas Stuzle (Eds.)* pp. 113–116.

Ramos, V., Muge, F. & Pina, P. (2002). Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies, *Frontiers in Artificial Intelligence and Applications, Soft Computing Systems - Design, Management and Applications, 2nd International Conference on Hybrid Intelligent Systems, IOS Press, in Javier Ruiz-del-Solar, Ajith Abraham and Mario Köppen (Eds.)* 87: 500–509.

Shapiro, L. & Stockman, G. (2001). *Computer Vision*, Prentice-Hall.

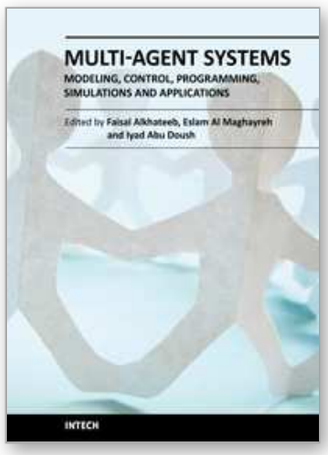
Theraulaz, G. & Spitz, F. (1997a). *Auto-organisation et comportement*, Hermes, Paris.

Theraulaz, G. & Spitz, F. (eds) (1997b). *What are the mechanisms involved in the emergence of cooperativity ? The spider model*, Hermes, Paris, pp. 85–90.

Vasilescu, G. (2005). *Physical Noise Sources*, Springer Verlag, Heidelberg, pp. 45–68.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*, Wiley and Sons.

IntechOpen



Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications

Edited by Dr. Faisal Alkhateeb

ISBN 978-953-307-174-9
Hard cover, 522 pages
Publisher InTech
Published online 01, April, 2011
Published in print edition April, 2011

A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Multi-agent systems can be used to solve problems which are difficult or impossible for an individual agent or monolithic system to solve. Agent systems are open and extensible systems that allow for the deployment of autonomous and proactive software components. Multi-agent systems have been brought up and used in several application domains.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Moussa Richard, Beurton-Aimar Marie and Desbarats Pascal (2011). Image Processing on MR Brain Images Using Social Spiders, Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications, Dr. Faisal Alkhateeb (Ed.), ISBN: 978-953-307-174-9, InTech, Available from: http://www.intechopen.com/books/multi-agent-systems-modeling-control-programming-simulations-and-applications/image-processing-on-mr-brain-images-using-social-spiders

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen