

LEVEL SET BRIAN SEGMENTATION WITH AGENT CLUSTERING FOR INITIALISATION

Fast Level Set Based MRI Tissue Segmentation with Termite-Like Agent Clustering for Parameter Initialization

David Feltell and Li Bai

School of Computer Science & IT, Nottingham University, Triumph Rd, Nottingham, UK

Keywords: Level Set, Multi Agent Clustering, Image Segmentation.

Abstract: This paper presents a novel 3D brain segmentation method based on level sets and bio-inspired methodologies. Level set segmentation methods, although highly promising, require manual selection of seed positions and threshold parameters, along with manual reinitialisation to a new level set surface for each candidate region. Here, the use of swarm intelligent mechanisms is used to provide all the statistical data and sample points required, allowing automatic initialisation of multiple level set solvers. This is shown by segmentation of white matter, grey matter and cerebro-spinal fluid in a simulated T1 MRI scan, followed by direct comparison between a commercial level application - FMRIB's FAST - and the ground truth anatomical model.

1 INTRODUCTION

The solution to initial value problems by level set methods has transformed the study of evolving interfaces and, along with the surge in available processing power, provides a unique and promising methodology for three-dimensional image segmentation. However, the initialization of surfaces and tuning of parameters almost universally requires human intervention. By using agent based clustering and segmentation algorithms, we can hope to automate this process to a large extent, thus no longer relying on expert knowledge or requiring multiple tedious trials for an acceptable result.

The paper begins by reviewing the history of level set methods, leading to the method currently used in this work. Following this is a brief review of the mechanisms of swarm intelligence, leading to previous attempts at data clustering and image segmentation using agent based principles. Next, the implementation details are given of the level set solver, followed by the agent-based initialization algorithm. Finally, preliminary results of the system are shown along with a discussion of its merits and the project's future direction.

1.1 Level Set Method

Level set methods for image segmentation rely on an evolving closed surface defined by a moving inter-

face, the front, which expands from a point out into the image, fitting itself to the region it is released within, and smoothing any noise. Among the advantages of level set methods is the natural ability for a single surface to seemingly split apart and merge without losing its identity. To accomplish this we add a further spatial dimension, ϕ , to our problem space, then we can in effect have a single surface in $(n + 1)$ -dimensional space, even though from observation of n dimensions we see two or more spatially distinct closed surfaces. This is the essence of the level set method and allows for topological changes to be handled naturally. This is easily visualised in the 2D case: for example Figure 1, where two 2D surfaces are projected to a 3D volume, where it is shown they are mathematically part of the same object. Numerically, each point in space U includes a scalar ϕ that can vary dynamically as time progresses, via a speed function F . Functionally, by taking a slice at time t of all points in U where $\phi = 0$, we are taking the 0^{th} level set of the function ϕ .

If we initialise ϕ to the signed distance from an initial closed surface, then we can visualise ϕ as the height at a certain point, and all the points in space with a height of 0 make up our surface. Thus, as well as handling topological changes naturally, the level set approach also has the advantage that the gradient across the surface at any given point on the interface can be found using $\nabla\phi$, meaning the local curvature can be determined. A curvature term gives the signed

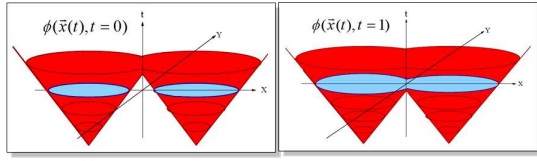


Figure 1: Illustration of level set function surface S (red) and evolution of moving interface/zero level set (blue). Topological change of two separate fronts (left) into a single front (right) handled naturally via higher dimensional function. After (Sethian, 1996).

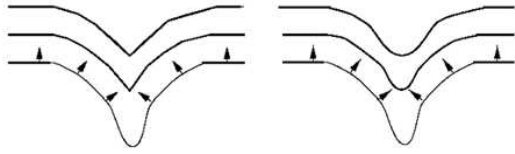


Figure 2: Illustration of surface evolution without curvature (left) and using a curvature term (right). After (Sethian, 1996).

'sharpness' of the interface at a point, allowing for a smoothing effect, overcoming noise and preventing leaks. This is illustrated in Figure 2. The typical formulation for the ϕ update function then becomes:

$$\phi_t = |\nabla\phi|(\alpha F + (1 - \alpha)\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}) \quad (1)$$

Where: ϕ_t is the time derivative of ϕ ; α controls the the level of smoothness in the surface; F is the data-dependent speed function; $\nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}$ gives local curvature at a given point.

Updating the values of ϕ at each point in space requires a numerical technique to evolve the zero level set from an initial specification. Naively then, we could initialise a surface in space, create a suitable speed function F , and numerically integrate until some condition is met and the surface extracted as all points where $\phi = 0$ (for example (Phillips, 1999)). An advantage here is sub-cell accuracy - the ϕ values can be interpolated to find points on a scale that is more accurate than the discrete embedding it is operating within. The first problem here is that the whole state-space must be evaluated each iteration, rather than just the surface. Secondly a small timestep is required to prevent numerical instabilities. These two caveats make the naive method undesirable for use outside of specialist fields.

Algorithms have been developed to overcome these issues by only performing updates on regions near the surface, rather than integrating over the entire state space. The most well-known is the Narrow Band approach (D. Adalsteinsson, 1995; Sethian, 1999),

where numerical integration is performed within a band of points initialised around the surface, though when the zero level set reaches the edge of this band, the band must be reset. This dramatically increases efficiency over the naive method, with little or no effect on accuracy. However, further optimization in this vein has come in the form of the Sparse Field approach (Whitaker, 1998). With this method the narrow band is reduced to the smallest workable size and the reinitialisation requirement is based on a purely local update, rather than a global update of the entire band. The reduction in accuracy is tolerable for most applications and its implementation has allowed the level set family of methods to achieve real-time performance levels in complex 3D applications (for example see (Lefohn et al., 2003; Lefohn et al., 2004)).

The Sparse Field approach has been taken even further in the work of (Karl, 2005), and is the method presented here. In this work the narrow band is lowered to simply two linked lists operating in a discrete space, L_{in} and L_{out} , representing the inside and outside boundaries of the zero level set, respectively. No attempt at sub-cell accuracy is made and most operations use integer math. The values of ϕ are also kept constant depending on their status and are updated discretely. There are four such values representing inside, inside edge, outside edge, and outside the volume, set so that a rough gradient can be found at any point (for example: 3, 1, -1, -3, respectively). The level set algorithm approximation itself is implemented in a two-pass process:

In the first pass elements in L_{in} and L_{out} are checked using the speed function F . If an expansion/contraction is provoked, the relevant list element is switched to the opposing list. If an element finds itself surrounded by cells with ϕ values of opposite sign, it is removed from its list. This fact allows for splits and merges in the topology.

The second phase involves an approximate Gaussian smoothing term G , that is, taking the (weighted) average of ϕ values from the area surrounding an element of L_{in} or L_{out} . Depending on the outcome of G , expansion/contraction adjustments to the lists and ϕ values are performed, similar to expansion via F . This has the effect of smoothing away noise as well as sharp protuberances in the surface.

Unlike previous algorithms, the process of expansion via the speed function and contraction via curvature are not intrinsically linked into the same update. In this algorithm, F is run on L_{in} and L_{out} a number of times. After this initial expansion phase a number of G runs are similarly done on L_{in} and L_{out} . The ratio of F to G runs determines the smoothness of the solution (in an analogue to the α parameter in (1)).

Further optimization can be made when the ratio of F runs is much higher than G runs. Elements visited on the list that have reached a locally optimal position will remain in that position on the next run of the speed function. Therefore they can be removed from successive F runs on that pass.

As mentioned in their paper title, this method departs almost entirely from the PDE based methods, but maintains many of the advantages. Sub cell accuracy is not possible in this generic form, but it would be a simple addition to use this method as a prototype generator for slower, but more accurate methods. The inclusion of the ϕ embedding, albeit much simplified, allows local topology to be determined at any point to discrete-level accuracy. Thus, the Gaussian function has the information required to approximate the 'sharpness' at a point, and to expand or contract as a result (in a reasonable approximation to the effect of curvature). The lack of the requirement to solve PDEs holds several advantages. No timestep is required in the update process - the algorithm is entirely discrete. The method maintains the implicit ability of splits and merges along the front, without relying on any gradient calculation, eliminating the need for expensive entropy-satisfying spatial derivative schemes.

Parameter selection for level set segmentation generally requires at least some level of human interaction. Specifically in the case of image segmentation, initialization of a level set solver requires at minimum seed locations, an ideal data value, and an acceptable noise threshold to be preset. Also, differing data classes in the same problem space, for example tissue types in medical scans, each require their own level set surface with their own set of initialization parameters. Ideally the task of assigning seed locations and calculating level set parameters would be automated. These issues lend themselves well to an agent based approach, which will be discussed in the following section.

1.2 Multi-Agent Swarm Based Algorithms

Agents are independent entities existing in some environment - sensing, processing and modifying the environment based on internal reasoning. For multi-agent systems, the swarm intelligence and self-organisation paradigms have become popularized in many disciplines as an explanation for the apparent mismatch in complexity of agent versus complexity of task (Camazine et al., 2001), and as a unique engineering metaphor (Bonabeau et al., 1999). Such systems of agents can focus on simple stimulus-response functions using purely local stimuli with little or no

cognition or direct communication. Tasks are accomplished by exploiting the non-linearity inherent in such a massively parallel system, rather than relying on individual agent complexity. An agent modifying the environment at a particular location allows another agent in that same location at a later time to sense this new state and respond accordingly. In this way the environment is used for indirect communication, termed *stigmergy*. Stigmergy in biological systems is further enhanced by the complexity of the environment, for example diffusion is utilized in many biological processes to spread information in the form of chemical gradients. Activation/attraction and inhibition/repulsion functions can thus be designed to control agent interactions based solely on local environmental state and any internal state, relying on quantitative information reinforcement and decay as well as qualitative signaling via the environment to control the weighting between possible responses. A global-level task or structure may then be many times more complex than an individual can perceive or accomplish, yet through parallel application of simple rules with indirect non-linear couplings, we see the spontaneous emergence of a solution. Given this non-linearity, a tiny change in a parameter can result in a drastically different solution, but (if the system is well formed) one that still conforms to a valid set of stable solutions - that is, the system exhibits *multistability*. We have demonstrated this engineering paradigm previously in modeling the building behaviour of *Macrotermes* termites (Feltell and Bai, 2004; Feltell et al., 2005). The use of an attractive cement pheromone to coordinate soil clustering behaviour has provided the inspiration for the swarm-intelligent clustering mechanism used here.

Swarm based clustering algorithms have been developed to cluster sets of n -dimensional data, generally projected onto a 2D grid, by taking inspiration from brood sorting and corpse clustering in ants (Monmarche et al., 1999; Monmarche, 1999; Kanade and Hall, 2003; Schockaert et al., 2004) as well as building behaviour in termites (Vizine et al., 2005). The approach does not need any prior knowledge of the problem space or number of clusters, and as a bonus gives a visual representation of the clusters on the 2D grid.

Agent based approaches have also been developed to directly segment an image, again using inspiration from natural systems such as ant pheromone trail networks (Ramos and Almeida, 2000), artificial life (Liu et al., 1997; Liu and Tang, 1999; Bocchi et al., 2005), social spiders (Bourjot et al., 2003), and even termites, bloodhounds and children (Fledelius and Mayoh, 2006). Here the agents interact directly

within the n -dimensions of the problem space, rather than outside it. This distinction from swarm based data clustering allows spatially localized regions to be processed independently - possibly requiring only a subset of the image space to be explored.

If the two perspectives can be combined, along with other swarm-inspired mechanisms, we could ideally form a localized data clustering algorithm, whereby ideal seed locations and other parameters would be found by agents within the image environment as they cluster similar voxels together.

Swarm intelligent systems emphasize robustness and diversity over accuracy, and so find solutions to complex problems that are 'good enough', but which require minimal agent capability in terms of both cognition and interaction. One of the major lures of level set algorithms is the tolerance for error, both in the problem space and, within limits, the initialization parameters. As long as 'good enough' seed location, ideal value and acceptable range parameters can be determined, the solution to the level set equations should be near-optimal in all but the hardest cases. With this assumption in mind, well located seeds can then be acceptable sample points used to approximate the voxel mean and range within a class.

2 IMPLEMENTATION

2.1 Level Set Solver

The level set solver is derived from the model of Shi & Karl (Karl, 2005). We define two sets of discretized points, representing the inside and outside layers between which lies the level set surface. As well as updating these sets as the surface expands or contracts we must maintain the values of ϕ , such that $\phi = 3$ inside the volume enclosed by the level set surface, $\phi = 1$ along the inside layer, $\phi = -1$ along the outside layer, and $\phi = -3$ outside the volume (note: the sign is reversed here compared to the original algorithm to be more in line with other level set methodologies). Finally, we must ensure that no orphaned points exist. That is, inside points must always have at least one outside point adjacent to them, and similarly outside points must always have at least one inside point adjacent. The update routine for both inside and outside layers is similar, though varies in parameter value. Following is a summary of the general update routine:

Let:

- Current time step be t .
- Two sets of points defining our two discretized layers be S_i and S_j .

- ϕ constants in space be: b_i along S_i ; b_j along S_j ; a_i in the volume beyond S_i ; a_j in the volume beyond S_j .
- Three-dimensional cartesian point vector, $\mathbf{p} = (pi + pj + pk)$.
- Neighbourhood of \mathbf{p} , $N(\mathbf{p}) = \{(pi + i + pj + pk), (pi - i + pj + pk), (pi + pj + j + pk), (pi + pj - j + pk), (pi + pj + pk + k), (pi + pj + pk - k)\}$.
- Context dependent speed function, $F(\mathbf{p}) : \mathfrak{R} \rightarrow \mathfrak{R}$, with condition f_i required for update to occur. Here and throughout this work it is assumed voxel intensity value is in $[0, 1] \in \mathfrak{R}$.

Then, for each $\mathbf{p} \in S_i$, iff $F(\mathbf{p}) \implies f_i$:

1. Increment time step, t .

$$t \leftarrow t + 1 \quad (2)$$

2. \mathbf{p} is added to S_j

$$S_j \leftarrow \{S_j \cup \{\mathbf{p}\}\} \quad (3)$$

3. ϕ at point \mathbf{p} is set to constant, b_j .

$$\phi(\mathbf{p}, t) \leftarrow b_j \quad (4)$$

4. \mathbf{p} is removed from S_i , whilst all relevant neighbours of \mathbf{p} are added to S_i .

$$S_i \leftarrow \{S_i - \{\mathbf{p}\}\} \cup \{\forall \mathbf{r} \in N(\mathbf{p}) \mid \phi(\mathbf{r}, t) = a_i\} \quad (5)$$

5. ϕ at all relevant neighbours of point \mathbf{p} is set to constant, b_i .

$$\phi(\{\mathbf{r} \mid \phi(\mathbf{r}, t) = a_i\}, t) \leftarrow b_i, \forall \mathbf{r} \in N(\mathbf{p}) \quad (6)$$

6. All points in S_i that are in not neighbored by a point in S_j are orphaned points and must be removed.

$$S_i \leftarrow S_i - \{\forall \mathbf{r} \in S_i \mid N(\mathbf{r}) \cap S_j = \emptyset\} \quad (7)$$

7. ϕ at all orphaned points in S_i is set to constant, a_j .

$$\phi(\{\mathbf{r} \mid N(\mathbf{r}) \cap S_j = \emptyset\}, t) \leftarrow a_j, \forall \mathbf{r} \in S_i \quad (8)$$

In practice, all these steps can be combined into a single update loop. The update loop is performed on both the inside and outside layers. Let our inside surface be S_{in} and the outside surface be S_{out} , then: $a_{in} = 3$, $a_{out} = -3$, $b_{in} = 1$, $b_{out} = -1$, $f_{in} \Leftrightarrow (F(\mathbf{p}) > 0)$, $f_{out} \Leftrightarrow (F(\mathbf{p}) < 0)$. The steps (3)..(6) deal with the surface evolution. The steps (7) and (8) handle shock propagation, that is, they remove points as the curve crosses over itself to maintain an unambiguous closed surface.

The algorithm is then:

Let:

- $D(\mathbf{p})$ be the problem specific data term, specifically the voxel gray level value at point \mathbf{p} .
- $J(\mathbf{p})$ be the problem specific speed function, specifically: $J(\mathbf{p}) = 1$ for $|(D(\mathbf{p}) - T)| < \epsilon$ and $J(\mathbf{p}) = -1$ for $|(D(\mathbf{p}) - T)| > \epsilon$, where: T is the ideal data value; ϵ is the acceptable error.
- $G(\mathbf{p})$ be a Gaussian smoothing approximation, specifically $G(\mathbf{p}) = (\phi(\mathbf{p}, t) + \sum_{\mathbf{r} \in N(\mathbf{p})} \phi(\mathbf{r}, t)) / 7$. With the condition $G(\mathbf{p}) = 0$ iff $|G(\mathbf{p})| < 1$, tighter areas can be explored, as smoother curves will neither contract nor expand. This produces more accurate solutions in most cases, but tends to give less smooth surfaces.
- t_J be the number of speed runs to perform; t_G be the number of Gaussian smoothing runs to perform.

Then:

- A Initialize S_{in} and S_{out} to small surface(s) about given seed locations.
- B Perform (2)..(8) with: $i = out$; $j = in$; $F(\mathbf{p}) = J(\mathbf{p})$.
- C Perform (2)..(8) with $i = in$; $j = out$; $F(\mathbf{p}) = J(\mathbf{p})$.
- D Repeat from (B) while $t < t_J$.
- E Perform (2)..(8) with: $i = out$; $j = in$; $F(\mathbf{p}) = G(\mathbf{p})$.
- F Perform (2)..(8) with $i = in$; $j = out$; $F(\mathbf{p}) = G(\mathbf{p})$.
- G Repeat from (E) while $t < t_J + t_G$.
- H Set $t = 0$. Repeat from (B) for t_S iterations.

2.2 Swarm based Parameter Initialization

The parameter specification for the level set solver as well as the number of tissue types requiring individual segmentation is beyond the capability of a level set algorithm alone. Instead, in this work we show how we can use a collection of agents following similar rules to previous agent clustering algorithms, but with the agents embodied within the image space itself, finding good seed locations as well as performing minor preprocessing functions.

Agents have real-valued position and direction, using simple nearest neighbour approximation when sensing the underlying discrete image voxels. They wander the image in an initially random direction,

however as they move they lay a quantity of attractive 'pheromone' in visited voxels. The quantity of pheromone is proportional to the divergence of voxel intensity $|\nabla^2 \cdot D|$ at their current location. A high divergence value often indicates a heterogeneity in the image - specifically an interface between two regions. Here, pheromone deposition further has the restriction $D > 0.1$ to avoid segmenting irrelevant black regions. Pheromone diffuses through the image lattice, with the edge of the lattice and near-black ($D < 0.1$) voxels acting as sinks. The addition of a positive reinforcement pheromone mechanism allows the agents to use gradient following behaviour in coordinating toward promising seed locations. As pheromone diffuses away it causes that area to receive less attention, ultimately meaning less suitable seed locations are abandoned. This reflects in many ways the clustering behaviour of termites in nest construction (Bruinsma, 1979), where heterogeneities stimulate deposits of soil laden with a cement pheromone, which in turn attracts other termite builders to the site.

This clustering behaviour, in the termite case, creates regular spaced piles or pillars of soil. Piles close to one-another thus compete for the termites attentions, eventually resulting in regularly spaced pillars. In this work, as with the termite paradigm, this mechanism means seed locations tend not to become too localised. The combined effect ultimately increases the agents' sampling efficiency within large images.

The generalised update routine for an individual agent is given as follows:

Let:

- t be the current time step of an agent.
- Real-valued point vector, \mathbf{p} be the current position of an agent
- Scalar pheromone field, $\rho(\mathbf{p})$ be the pheromone concentration at location \mathbf{p} .
- Speed, s be a scalar speed value for the agent.
- Vector \mathbf{v} be the continuous direction vector of the agent.

Then:

1. Increment time step, t .

$$t \leftarrow t + 1 \quad (9)$$

2. Lay pheromone proportional to the logarithm of the absolute divergence. Logarithmic scale is used here to avoid floating point accuracy overflow.

$$\log(\rho(\mathbf{p})) \leftarrow \log(\rho(\mathbf{p}) + |\nabla^2 D(\mathbf{p})|) \quad (10)$$

3. Calculate new direction vector from weighted gradient of $\log \rho$.

$$\mathbf{v} \leftarrow \mathbf{v} + \alpha(\mathbf{v} - \nabla \log \rho) \quad (11)$$

4. Calculate speed scalar based on local density of ρ .

$$s \leftarrow 1 + \frac{\beta}{1 + \log(\rho(\mathbf{p}))} \quad (12)$$

5. Calculate updated position based on normalized direction vector and speed value.

$$\mathbf{p} \leftarrow \mathbf{p} + \frac{\mathbf{v}}{\|\mathbf{v}\|} \cdot s \quad (13)$$

Where: α is the weighting of pheromone gradients in movement; β controls the maximum speed of an agent.

The full agent algorithm is then:

- A Initialize set of n agents, M , each with random position within the image and random direction.
- B Perform (9)..(13) for each $m \in M$.
- C Diffuse pheromone: $\partial_t \rho = d \cdot \nabla^2 \rho$
- D Repeat from (B) while $t < t_A$ time steps.
- E Select the top C voxel locations with highest pheromone level ρ , to get full seed list, Q .
- F Run k-means clustering on Q , classifying into prespecified number of sets, Q_i , where $i = \{1, 2, \dots, q\}$.
- G Within each Q_i calculate mean $\mu(Q_i)$ and standard deviation $\sigma(Q_i)$, giving ideal data value $T = \mu(Q_i)$ and error threshold $\varepsilon = \sigma(Q_i) + k$, where k is a constant, within a class.

Once T and ε have been calculated for a seed set Q_i , the level set routine can be run, initialising L_{in} and L_{out} surfaces as pseudo-spheres about each seed location in Q_i . The agent algorithm need only be run once, then the level set algorithm run once for each seed set.

Additionally, in practice the diffusion step can be moved into a separate thread to take advantage of modern day multiple core processors. With the simplest distributed setup, with little control on synchronization or load balancing, the algorithm can still function correctly - an implicit advantage of the robust swarm metaphor.

The image is finally extracted as all voxel positions where $\phi > 0$, thus all points lying on and within L_{in} .

3 RESULTS

The dataset used comes from BrainWeb's online MRI simulator (McConnell BIC, 2007) with default parameters: modality T1, slice thickness 1mm, noise

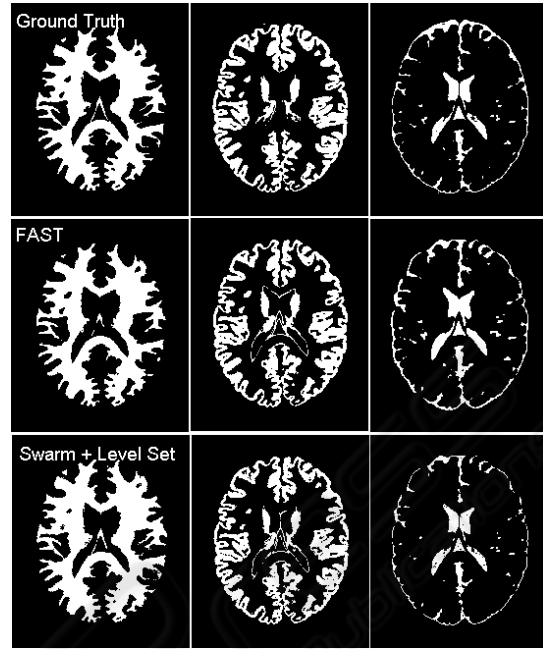


Figure 3: Comparison of ground truth to FAST and the swarm initialised level set solution presented here. Images used in both FAST and the present work have been pre-processed by BET. From left to right: white matter; gray matter; cerebro-spinal fluid.

3% and non-uniform intensity 20%. For comparison with ground truth and FAST (Zhang et al., 2001) the number of classes is set $q = 3$. The various parameters, unless otherwise stated, are set as follows:

$$\alpha = 2; \beta = 2.5; d = 0.015; k = 0.02; t_S = 10; t_A = 3000; t_J = 30; t_G = 3$$

The parameter values above for α, β, d and k , were found in part using a real-valued genetic algorithm, using the similarity to the ground truth images, minus the similarity between one another, as a fitness function. This process has proven useful so far, yet is only partially utilised here and remains a direction for future work. Other values are chosen intuitively to balance execution time and accuracy.

From Figure 3 it can be seen how the approach, even in this early stage, can compete with popular solutions such as FMRIB's FAST, which uses a method based on Markov random fields. In both solutions, preprocessing has been performed using FMRIB's popular BET (Smith, 2002) to skull strip and normalise the image voxels.

Quantitative measurement on accuracy is approximately possible given the hand segmented ground truth images available. In the above case, FAST achieves approximately 92%, 85% and 57% accuracy, whereas the swarm initialised level set algorithm

achieves 86%, 83% and 55%, for white matter, grey matter and cerebro spinal fluid, respectively. FAST remains superior, but the small score difference certainly indicates the presented approach is well worth further investigation.

In terms of execution time, from the point of file loading to final file output, FAST outdoes the performance of the presented work by just over double. The above solution was found by the swarm initialised level set algorithm in approximately 17 mins, whereas FAST finished in approximately 8 mins. It is worth noting that the swarm approach presented is still in early development, and as discussed in the next section, has the advantage of naturally supporting a distributed implementation.

4 DISCUSSION AND FUTURE WORK

The level set solver is an approximate discrete solution to a continuous problem. Although the algorithm used in this work is particularly fast, it simply does not have the power of a narrow band or even sparse field approach. These methods not only reflect the naive approach more directly, but also have sub-cell accuracy, where actual surface points can be interpolated within image voxels using the real-valued gradient of ϕ . They can also vary in speed along different areas of the surface, allowing for a more global curvature force effect. The discretisation of ϕ values to simply the set $\{-3, -1, 1, 3\}$ means the curvature term is much more rigid, and does not distinguish on a local level between corners of differing sharpness. Ideally the current level set solver approximation would work as a rapid prototype for a more accurate narrow band based method. This remains a direction for future work.

One benefit of using a swarm based system is in the robustness to a noisy environment. In this regard it should be possible to include an atlas mechanism, using only the most minimal registration with the candidate image, and restrict agent interaction within or near the atlas template for a given region. This would give the agents' environment a significant amount of extra information, which can be used to focus agent efforts - increasing efficiency and allowing for segmentation of more difficult (too smooth/too noisy) regions.

The preprocessing by BET could be removed for future versions. BET simplifies segmentation significantly, and prevents areas such as the skull being segmented as part of brain matter. However, preliminary results of segmentation without BET pre-

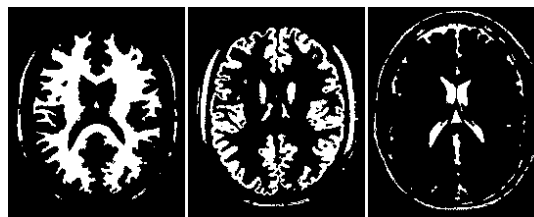


Figure 4: Solution found using current algorithm without BET preprocessing. All parameters are set similar - the number of classes remaining at 3. From left to right: white matter, gray matter, cerebro-spinal fluid.

processing are promising in their own right, though show several mismatched regions (see Figure 4), and it is likely that with more classes and ground truth samples, along with evolutionary algorithm parameter tuning, the method can be applied to multiple modalities and varying image spaces without the need for any preprocessing.

Another benefit of the multi-agent paradigm is its naturally distributed nature. As has been eluded to, the diffusion of pheromone can already be ported to another processing unit to greatly increase efficiency. A further extension could see the image space and/or agents split to run parallel. Certainly each region's level set solver could in future run on separate processes to dramatically decrease execution time.

REFERENCES

- Bocchi, L., Ballerini, L., and Hssler, S. (2005). A new evolutionary algorithm for image segmentation. *Applications on Evolutionary Computing*, pages 264–273.
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Bourjot, C., Chevrier, V., and Thomas, V. (2003). A new swarm mechanism based on social spiders colonies: From web weaving to region detection. *Web Intelligence and Agent System*, 1(1):13–32.
- Bruinsma, O. (1979). *An Analysis of Building Behaviour of the Termite *Macrotermes Subhyalinus* (Rambur)*. PhD thesis, Landbouwhogeschool, Wageningen.
- Camazine, S., Deneubourg, J.-L., Franks, N., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton University Press.
- D. Adalsteinsson, J. S. (1995). A fast level set method for propagating interfaces. *J. Comp. Phys.*, 118:269–277.
- Feltell, D. and Bai, L. (2004). Swarm robotics for construction. In *24th SGA International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Cambridge, UK.

- Feltell, D., Bai, L., and Soar, R. (2005). Bio-inspired emergent construction. In *IEEE 2nd Intl. Swarm Intelligence Symposium*, Pasadena, CA, USA.
- Fledelius, W. and Mayoh, B. (2006). A swarm based approach to medical image analysis. In *24th IASTED International Conference on Artificial Intelligence and Applications (AIA 2006)*, pages 150–155, Anaheim, CA, USA. ACTA Press.
- Kanade, P. and Hall, L. (2003). Fuzzy ants as a clustering concept. In *22nd International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 227–232.
- Karl, Y. S. W. (2005). A fast level set method without solving pdes. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2005)*, volume 2, pages 97–100.
- Lefohn, A. E., Cates, J. E., and Whitaker, R. T. (2003). Interactive, gpu-based level sets for 3d segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2003)*, pages 564–572.
- Lefohn, A. E., Kniss, J. M., Hansen, C. D., and Whitaker, R. T. (2004). A streaming narrow-band algorithm: Interactive computation and visualization of level sets. *IEEE Transactions on Visualization and Computer Graphics*, 10:422–433.
- Liu, J. and Tang, Y. (1999). Adaptive image segmentation with distributed behavior-based agents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:544–551.
- Liu, J., Tang, Y., and Cao, Y. (1997). An evolutionary autonomous agents approach to image feature extraction. *IEEE Transactions on Evolutionary Computation*, 1:141–158.
- McConnell BIC (2007). Brainweb: Simulated brain database, Montreal Neurological Institute. <http://www.bic.mni.mcgill.ca/brainweb/>.
- Monmarche, N. (1999). On data clustering with artificial ants. In Freitas, A. A., editor, *Data Mining with Evolutionary Algorithms: Research Directions*, pages 23–26, Orlando, Florida. AAAI Press.
- Monmarche, N., Slimane, M., and Venturini, G. (1999). On improving clustering in numerical databases with artificial ants. In *ECAL '99: Proceedings of the 5th European Conference on Advances in Artificial Life*, pages 626–635, London, UK. Springer-Verlag.
- Phillips, C. L. (1999). The level set method. *The MIT Undergraduate Journal of Mathematics*, 1:155–164.
- Ramos, V. and Almeida, F. (2000). Artificial ant colonies in digital image habitats - a mass behaviour effect study on pattern recognition. In *2nd Int. Workshop on Ant Algorithms (ANTS 2000)*, pages 113–116, Brussels, Belgium.
- Schockaert, S., Cock, M. D., Cornelis, C., and Kerre, E. E. (2004). Fuzzy ant based clustering. In *International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS 2004)*, pages 342–349, Brussels, Belgium.
- Sethian, J. (1996). Level set methods: An act of violence - evolving interfaces in geometry, fluid mechanics, computer vision and materials sciences. *American Scientist*.
- Sethian, J. (1999). *Level Set Methods and Fast Marching Methods*. Cambridge University Press.
- Smith, S. (2002). Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155.
- Vizine, A., de Castro, L., Hruschka, E., and Gudwin, R. (2005). Towards improving clustering ants: An adaptive ant clustering algorithm. *Informatica*, 29(2):143–154.
- Whitaker, R. (1998). A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231.
- Zhang, Y., Brady, M., and Smith, S. (2001). Segmentation of brain mr images through a hidden markov random field model and the expectation maximization algorithm. *IEEE Trans. on Medical Imaging*, 20(1):45–57.