

To Boldly Split: Partitioning Space Filling Curves by Markov Chain Monte Carlo Simulation

Daniel Valdes and Abhir Bhalerao

Department of Computer Science
The University of Warwick
Coventry, UK, CV4 7AL
{dvaldes, abhir}@dcs.warwick.ac.uk

Abstract. Space filling curves are a class of fractals that are important mathematical descriptions of the appearance and shape of natural objects. There is growing interest in the modelling of such curves to measure pathology in medicine and biology. This work presents a method of modelling fractal curves, such as the boundary of brain white matter, and partitioning such curves in to segments having equal fractal dimension. Since the solution space, for a given number of contour points and a required set of partitions is very large, we employ a Bayesian framework of reversible-jump Markov chain Monte Carlo (MCMC) and a sampler based on the Metropolis-Hastings test. We detail the algorithm and present results on both simple contours (animal silhouettes) and space-filling brain contours and show the convergence characteristics of the method. We discuss its use for building compact local statistical shape models.

1 Introduction

Being able to partition a contour into a set of meaningful local parts has many applications in machine vision, image retrieval, terrain classification, handwriting recognition etc. Indeed, certain types of curves, e.g. fractals, can only be characterised by their local properties and are either repetitive arrangements of similar patterns or the result of applying a syntactical rule over a set of scales.

Mandelbrot's [1] work on fractal geometry demonstrated how fractal dimension could characterise many natural phenomena, such as the folding of the human brain cortex, and it has motivated clinical studies that use 3D MRI to understand the folding process and correlate changes in fractal dimension with age and disease [2,3,4,5]. In [2], a study reported the effects of age related white matter structural changes on estimates of fractal dimension. A similar study in [3] performed a three-dimensional fractal analysis on the white matter surface to quantify its complexity. A volumetric method based on the FFT was used in [4] to address the question of whether the human cerebral cortex is indeed self-similar. Kontos et al. [5] provided a study of MRI activation patterns using space filling curves to relate brain structures with brain functions.

Brain imaging has empowered medicine and neuroscience with the ability to use the images to identify structural brain changes associated to neurodegenerative diseases. Since the brain structure is so complex and varies across subjects, methods to identify

morphological variability can be useful for diagnosis, treatment and to assess the structural basis of normality and disease [6]. Statistical shape modelling (SSM) is concerned with the analysis of shapes of sub-structures and aims to describe variations across groups of individuals (taking into account for example age, sex, genetic background and disease state). Shape modelling has also proven that brain variation can be successfully captured [7], due to the effectiveness of capturing the variation of sets of contours (shapes) [8]. By this approach, a set of training shapes is used to build the model and then a given test-shape can be compared against the model. Fractals and space filling curves defy such analysis because each example is, so to speak, its own exemplar and the relative complexity of such shapes: (1) make it hard to establish meaningful correspondences between shapes; and (2) require a much larger training size. This reflects that work on shape modelling is constrained by many unsolved problems including segmentation of structures, dependency on manual data processing and difficulties in modelling *local* versus *global* variation. A local SSM approach was proposed in [9] and it was demonstrated how local deviations from the fractal characteristics of space-filling curves could be analysed based on pose-alignment and an SSM. This work allows the pose variation between symmetrical parts of the curve to be eliminated and, in theory, allows the SSM to represent the interesting variation between parts. However, an open question of this work was what should be a good partition of a given contour into local parts and how can this be determined in a simple and efficient way.

Several methods for contour partitioning have been reported in the image analysis literature. Richards and Hoffman [10] developed shape descriptors known as codons that decompose contours at negative curvature extremes. Asada and Brady [11] introduced a method based on Marr's primal sketch representation of intensity changes in gray level images to describe significant changes in curvature by matching the multi-scale convolutions of a shape. Baruch et al. [12] reported a method that uses the differential chain code of the edge to segment it into sections with constant curvature. Furthermore, Wuescher and Boyer created a technique for partitioning contours into constant curvature segments using a nonlinear blip filters matched to the impairment signature of the curvature computation [13]. More recently Cronin [14] proposed a method based on parsing concavity to partition digital contours into concave and convex sections. The use of Curvature Scale-Space (CSS) as proposed by Mokhtarian [15] and later applied by Abbasi et al [16] in their work on contour modelling for retrieval (the University of Surreys SQUIDS retrieval-by example database), is also an important and useful development.

In this paper, we address the problem of partitioning contours to support the construction of local statistical shape models, where the contour partitioning needs to be guided by the fractal dimension. We show that the resulting algorithm is also able to find symmetries in curves, which may be of use in other applications. Given a closed contour of N points and M being the number of required partitions, if $M \ll N$, it is not hard to see that the number of potential partitions becomes very large¹. Because of the need to search a large solution space, the problem is cast into a Bayesian

¹ For example: if $N = 512$ and $M = 16$, then in the case that each partition starts with 32 points, and we allowed neighbouring partitions to exchange $\frac{1}{2}$ their points by moving the border, there would be $16^{16} \approx 2^{19}$ possible configurations to test.

framework by setting up a stochastic search within the parameter space configurations of M required partitions. The proposed method draws samples from a parameter space using a reversible-jump Markov chain Monte Carlo simulation [17] where each state of the chain represents a likely partition and can be assessed using a posterior probability. In the following, we describe in detail how the contour partitioning problem can be cast into this framework and illustrate its operation on a set of simple object outlines (animal silhouettes) and brain white matter contours. We conclude by discussing the potential of this approach.

2 Markov Chain Monte Carlo Simulation for Contour Partitioning

Let a piece-wise linear contour of N points be defined by the coordinates: $X = \{\vec{x}_i\}$, $i = 1 \dots N$, and let a given partition of M exclusive subsets be the parameter sets $\Theta = \{\Theta_1, \Theta_2, \dots, \Theta_M\}$. Let the posterior probability of a given partition be given by

$$P(\Theta|X). \tag{1}$$

Then by Bayes' rule, $p(\Theta|X) \propto p(X|\Theta)p(\Theta)$, which is the product of the likelihood of the data given a partition Θ and the *prior probability* of the partition Θ .

As the aim of the partitioning process is to end up with λ_M contour subsections Θ_k of more or less equal size ($card[\Theta_j]$) and equal fractal dimension, we can

1. Model the prior as

$$\pi_n(k; \Theta_j) = \frac{e^{-\lambda_n} \lambda_n^k}{k!}, \tag{2}$$

which is a Poisson density with rate parameter λ_n .

2. Model the data distribution by the likelihood function

$$l(\Theta_j|X) = e^{-|FD(\Theta_j)-t|}, \tag{3}$$

which favours the partition having a fractal dimension, FD , of t .

3. Penalise the posterior by a density to keep the number of partitions to be the desired number, $\lambda_M = M$, the required partition size,

$$\pi_M(k; \Theta) = \frac{e^{-\lambda_M} \lambda_M^k}{k!}. \tag{4}$$

Taken together, we can now expand the posterior (1) as

$$p(\Theta|X) \propto \pi_M(card[\Theta]; \Theta) \prod_j^M l(\Theta_j|X) \pi_n(card[\Theta_j]; \Theta_j) \tag{5}$$

In the simulations presented below, for the likelihood function we use a value of $t = 2.57$ based on studies of brain contours. Note that this probability model is being setup to assume that there are *local* parts of the contour which have fractal dimension t . Of course, if it were a fractal and not discretely sampled then by definition this assumption would hold at arbitrary scales.

2.1 Stochastic Sampling and Reversible Jumps

Markov chain simulation (or MCMC) is a method based on drawing values of parameters θ from functions that approximates it, and then draws are altered to obtain a better approximation to the target posterior distribution $p(\theta|X)$. Since the samples are drawn sequentially and the distribution that generates these depends only on the previous value drawn, the set of states form a Markov chain [18]. To ensure the success of the algorithm it is necessary to create a Markov process whose stationary distribution is the specified $p(\theta|X)$ and then run the simulation for a sufficient length of time.

The Metropolis-Hastings (MH) algorithm can be considered as a generalisation of a class of Markov chain simulation methods for drawing samples from posterior distributions. Given a candidate-generating density, $p(\theta, \theta')$, of moving from state θ to θ' , it is clear that the stationary distribution is reached when $p(\theta'|X)p(\theta, \theta') = p(\theta|X)p(\theta', \theta)$ is satisfied for all θ, θ' . The MH algorithm performs rejection sampling by comparing the validity of the two states by forming an acceptance ratio

$$\alpha(\theta, \theta') = \min \left[1, \frac{p(\theta'|X)p(\theta, \theta')}{p(\theta|X)p(\theta', \theta)} \right],$$

and then, if $\alpha = 1$, the move $\theta \rightarrow \theta'$ is accepted, otherwise it is accepted with probability $\alpha \geq r \sim U[0, 1]$. Note that it is easy to see that if the candidate-generating densities are equal the state changes $\theta \rightarrow \theta'$ and $\theta' \rightarrow \theta$, then the chain automatically moves to a higher probability state. For the contour partitioning, we allow three moves in the state space: splitting a chosen contour partition at a random point; merging a random pair of neighbouring partitions and altering the position of a boundary between neighbouring segments (see figure 1). The simulation is initialised with a random number of partitions (e.g. 12) and equal candidate-generating probabilities: $P(split) = P(merge) = P(alter) = \frac{1}{3}$, with moves chosen at random. Note that for the split move, the random position is first chosen in the range $1 \dots N$, then the (current) segment in which it lies is split at that point. This ensures that larger segments are more likely to be split than smaller ones.

2.2 Estimating Fractal Dimension

The fractal dimension of a curve estimates of how much space it occupies relative to its length. For a planar curve, it can be calculated by the number, $N(\delta)$, of area

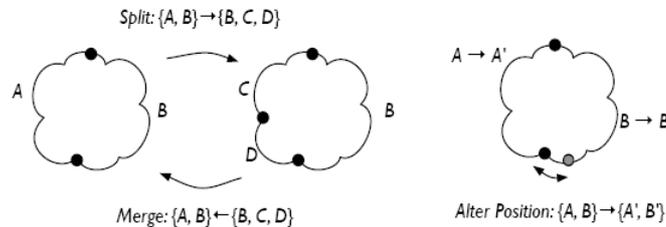


Fig. 1. Illustration of the (1) Split, Merge and (2) Alter Position moves

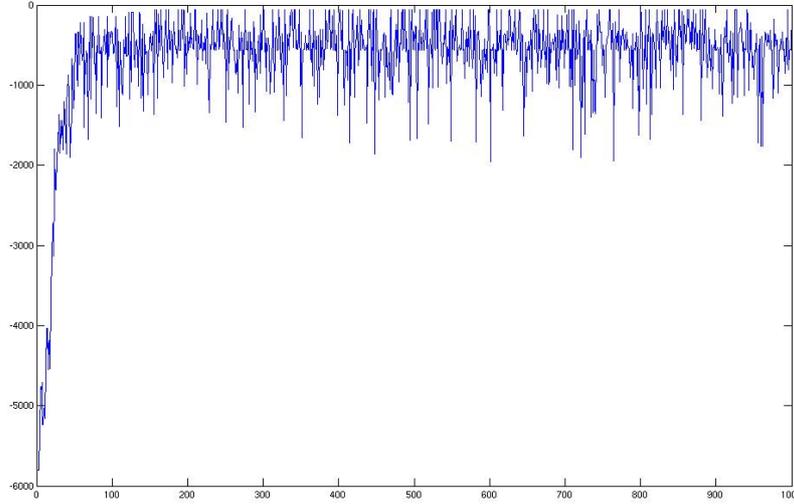


Fig. 2. Typical convergence characteristics. Plot shows log posterior against iteration (move) number. (This plot is for the brain white matter contour ‘slice 179’ used in the experiments below).

elements, δ , needed to cover the curve. The estimate, then, of the length of the curve is $N(\delta)\delta$. The Hausdorff-Besicovitch dimension D of a curve is defined as some measure $M_d(\delta)$ for which $N(\delta)\delta^d$ abruptly changes from zero to infinity [19]. Since D is often finite, it can be shown that for sufficiently small δ

$$D = -\frac{\log N(\delta)}{\log(\delta)}. \tag{6}$$

For practical purposes, by taking a set of meshes of varying δ over the curve, it is possible to count $N(\delta)$ and plot $\log N(\delta)$ against $\log(\delta)$.

2.3 Practical Considerations

To keep the posterior calculations numerically stable, they are performed in the *log* domain. Taking natural *logs* of (5) we can replace the products by sums of *log* likelihoods and log Poisson densities. Also, for small M , the factorial of ($\text{card}[\Theta_j]$) can become large, so it is convenient to use Stirlings approximation of factorial,

$$\ln n! \approx n[\ln n - 1], \tag{7}$$

thus:

$$\begin{aligned} \ln p(\Theta|X) &\propto m(\ln[\lambda_M] - \ln[m] - 1) - \lambda_M \\ &\quad - \sum_j^M |fd(\Theta_j) - t| \\ &\quad + \sum_j n_j(\ln[\lambda_n] - \ln[n_j] - 1) - \lambda_n, \end{aligned} \tag{8}$$

where $n_j = \text{card}[\Theta_j]$ and $m = \text{card}[\Theta]$.

As with all MCMC implementations, trial and error was used to determine at what number of steps convergence was being achieved. For the contours presented below (each with approximately 3000 points) we began by running 1000 steps, however convergence was observed after about 300 steps (see figure 2). Also, we manually selected one point on the contour as a fixed point which marked a line of symmetry: for all contours, this was one of the two points along the principal axis of the contour.

3 Experimental Results

We present results of contour partitioning using MCMC simulation on real contours: animal silhouettes (birds, fish, squids) and brain white matter outlines. The contours were sampled with approximately 3000 points and we chose to vary the number of required partitions, λ_M e.g. 2, 4, 8 etc. and set the mean number of points per partition, $\lambda_n = N/\lambda_M$. This required number of partitions (2, 4 and 8) were only used with illustrative purposes to demonstrate the ability of the algorithm to find symmetries, but others can be used. The simulations were initialised with $M = 12$ and $P(\text{split}) = P(\text{merge}) = P(\text{alter}) = \frac{1}{3}$ and a single point was user selected along a natural symmetry line. Each of the figures show the starting configuration of random partitions and the converged results. The plots of the resulting partitions present a colour

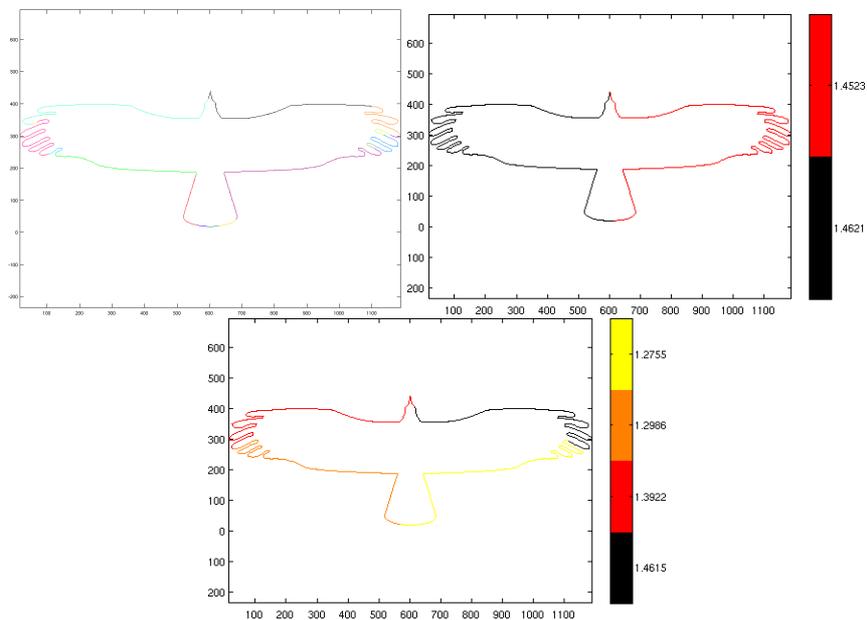


Fig. 3. Example results on bird contours: random starting partitions (top left), 2 (top right) and 4 (bottom) partitions respectively

Table 1. Results for the brain contour #179 : $\lambda_M = 8$

Partition	1	2	3	4	5	6	7	8
Points	378	455	441	432	440	403	380	366
FD	1.7107	1.5794	1.6357	1.6138	1.5362	1.6208	1.4745	1.5955

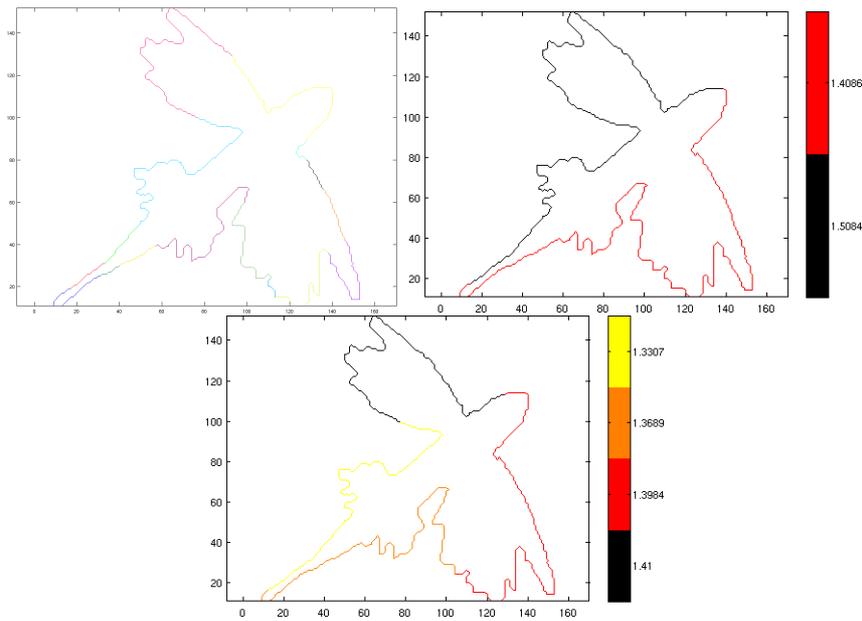


Fig. 4. Example results on SQUID database contours: random starting partitions (top left), 2 (top right) and 4 (bottom) partitions respectively

code going from black for the partition with the highest fractal dimension, to yellow, depicting the lowest.

For the bird (figure 3) and squid contour (figure 4), the data clearly has a one-fold symmetry along its length and we would expect this to be found. The results for 2, and 4 partitions are depicted. The partition into 2 is almost perfect in both cases. In the case of the squid, the symmetry is close to that required on the right-hand side of the head. These are relatively low curvature elements and the fractal dimension will not change much in moving the partition boundary from the centre of the head to the converged point. The results for 4 partitions are less obvious but it is encouraging to see that the partitioning is mirrored along the principal axis in both cases.

Figures 5 and 6 show results on a contour taken from white matter segmentation of two slices of a MRI brain image. Here, the plots of the resulting partitions present a colouring code going from black for the partition with the highest fractal dimension, to yellow, depicting the lowest. Tables 1 and 2 show the partitioning results in a quantitative way, describing the number of constituent points and the fractal dimension of the

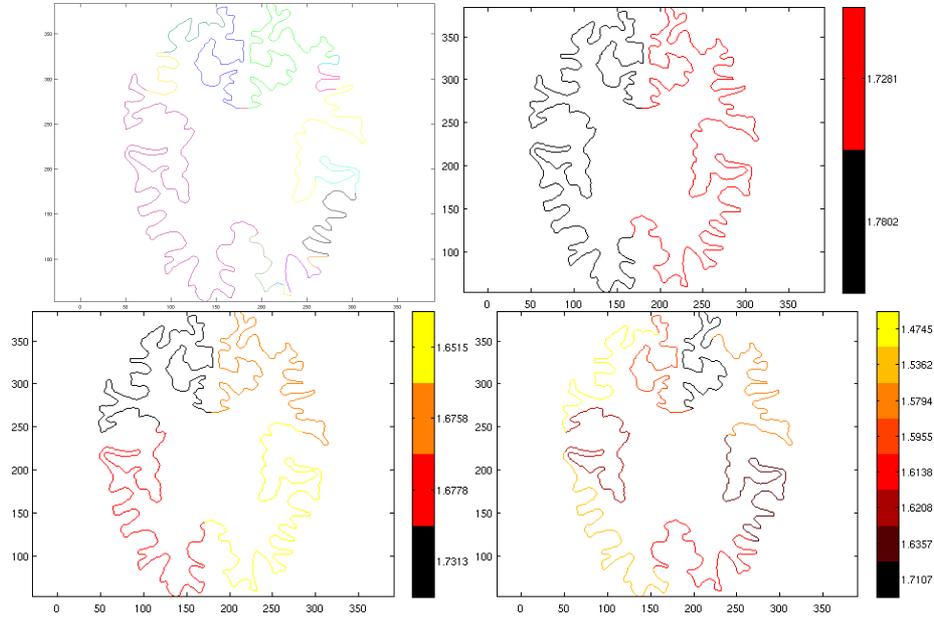


Fig. 5. White matter brain contour partitioning - slice #179. Top left: start configuration of MCMC using arbitrary partitioning. Results shown for $\lambda_M = 2$ (top right), $\lambda_M = 4$ (bottom left) and $\lambda_M = 8$ (bottom right) partitions after 1000 iterations.

Table 2. Results for the brain contour #182 : $\lambda_M = 8$

Partition	1	2	3	4	5	6	7	8
Points	325	343	361	394	430	406	435	424
FD	1.6689	1.5502	1.5259	1.6208	1.5316	1.5929	1.6689	1.5973

resulting partitions. The contours are similar as they are taken from brain slices which are close by to each other (slices #179 and #182). The MCMC partitioning was run for 1000 steps and 2, 4, and 8 partitions were sought starting from 15 random segments. A point on the line which separate the left and right hemispheres of the brain, (on the sagittal line) was marked by hand. The symmetries of the two halves, and the anterior and posterior lobes are correctly found in both case ($\lambda_M = 2$ and $\lambda_M = 4$). The result when 8 partitions are sought is harder to interpret. For slice #179 only 6 partitions are found at convergence, whereas for slice #182, 8 were found. This illustrates that in the former, the likelihood is sufficiently strong so as to drive the result away from the Poisson rate of $\lambda_M = 8$, where as in the latter, 8 partitions is the better solution. Nevertheless, there is good agreement in both cases for the other partitions. If we were to use this to partition all the brain slices, one way to achieve better correspondence between slices would be to initialise neighbouring slices with converged partitions from the current slice.

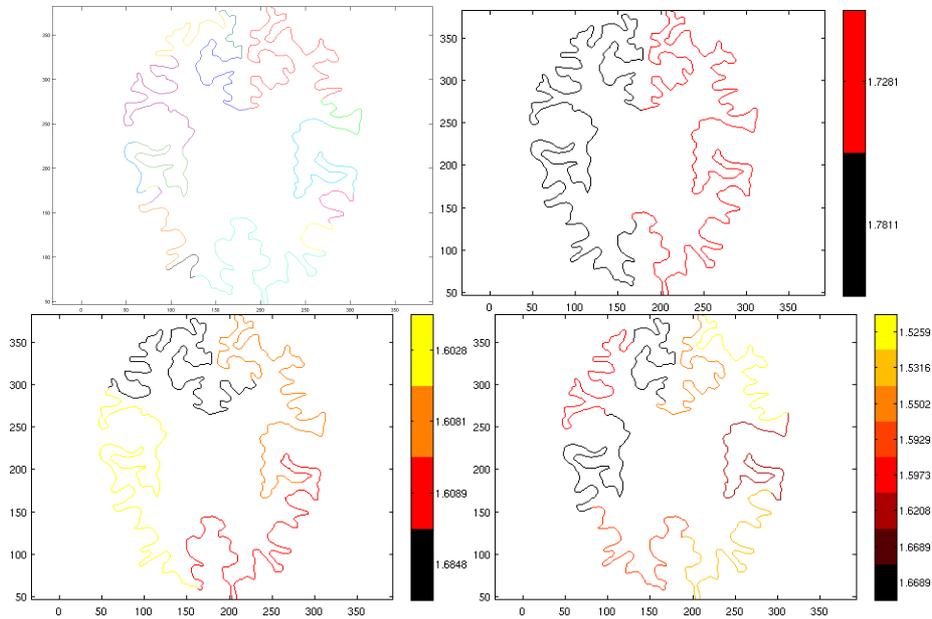


Fig. 6. White matter brain contour partitioning - slice #182. Top left: start configuration of MCMC using arbitrary partitioning. Results shown for $\lambda_M = 2$ (top right), $\lambda_M = 4$ (bottom left) and $\lambda_M = 8$ (bottom right) partitions after 1000 iterations. Compare with figure 5.

4 Discussion

For a genuine fractal curve, by definition, any arbitrary partition should have the same fractal dimension. Even contours found in biology are not always fractals but can exhibit self-similarity, they must be discretely sampled and they may be made up of parts which are fractal and non-fractal. Brain contours are space filling (in 2D and 3D) and the presented method shows potential in being able to produce a reasonable set of partitions. Ultimately, the veracity of the result can only be confirmed by the construction of the shape models. In [20] it is shown how the resulting sets from contour partitioning are useful for analysis of local variation in brain structures.

The aim here is to be able to partition the contour into a set of objects which are related by affine symmetry and then variations from these local symmetries can be identified [9]. Moreover, another objective of the idea of generating sets of partitions from contours is to establish a way of determining meaningful local sets of shapes. We believe that that this model could have useful application in brain morphometrics and computational anatomy. To be more precise, the provided method can be adapted for clinical diagnosis software for assessing changes in local shape variation of anatomical structures, such as, white/gray matter. Finally, is possible to address the problem of image database retrieval where the objective can be to discover images which contain objects similar to query objects, in this case brain sections.

The proposed MCMC simulation approach has some important characteristics. It is only necessary to set the number of required partitions and the simulation is able effectively search the state space and adapt locally to shrink and grow to equalise the fractal dimension over the whole contour. This would not be possible to do exhaustively for nontrivial numbers of segments. By using a Bayesian framework and a stochastic sampler, the convergence is theoretically guaranteed: in practice, our experiments show convergence after a relatively modest number of moves. It is evident that other shape descriptors can be used to generate the partitions, like in [20] where using curvature scale space consistency is used to partition the the contour into locally similar parts. Here, the idea is exploring alternative solutions to this problem in the fashion of the Bayesian paradigm and to gain insight about the complexity of the brain structure.

There is scope to model the likelihood function in different ways. We chose to use fractal dimension but mean curvature could be a viable alternative. Also, our density uses an absolute norm $|\text{fd}(\Theta_j) - t|$ and perhaps a Gaussian could be used instead. As well as investigating these questions, we are conducting analysis of several sets of brain contours and will be comparing our results with an implementations of a curvature scale space based method. An unsupervised selection of the initial point can be envisaged using the zero-crossings/extrema points generated by the contour evolutions of the CSS method [15].

References

1. Mandelbrot, B.B.: The fractal geometry of nature. Freeman, New York (1982)
2. Zhang, L., Liu, J.Z., Dean, D., Sahgal, V., Yue, G.H.: A three-dimensional fractal analysis method for quantifying white matter structure in human brain. *Journal of Neuroscience Methods* 150(2), 242–253 (2006)
3. Free, S.L., Sisodiya, S.M., Cook, M.J., Fish, D.R., Shorvon, S.D.: Three-dimensional fractal analysis of the white matter surface from magnetic resonance images of the human brain. *Cerebral Cortex* 6(6), 830–836 (1996)
4. Kiselev, V.G., Hahn, K.R., Auer, D.P.: Is the brain cortex a fractal? *NeuroImage* 20(3), 1765–1774 (2003)
5. Kontos, D., Megalooikonomou, V., Ghubade, N., Faloutsos, C.: Detecting Discriminative Functional MRI Activation Patterns Using Space Filling Curves. In: *Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology*, vol. 1, pp. 963–966 (2003)
6. Ashburner, J., Csernansky, J.G., Davatzikos, C., Fox, N.C., Frisoni, G.B., Thompson, P.M.: Computer-assisted imaging to assess brain structure in healthy and diseased brains. *Lancet Neurology* 2, 79–88 (2003)
7. Styner, M., Gerig, G., Lieberman, J., Jones, D., Weinberger, D.: Statistical shape analysis of neuroanatomical structures based on medial models. *Medical Image Analysis* 7, 207–220 (2003)
8. Cootes, T.F., Hill, A., Taylor, C.J., Haslam, J.: Use of active shape models for locating structures in medical images. *Image and Vision Computing* 12(6), 355–365 (1994)
9. Bhalerao, A., Wilson, R.: Warplets: An image-dependent wavelet representation. In: *International Conference on Image Processing, ICIP*, vol. 2, pp. 490–493 (2005)
10. Richards, W., Hoffman, D.D.: Codon constraints on closed 2-D shapes. *Computer Vision, Graphics, and Image Processing* 31(3), 265–281 (1985)

11. Asada, H., Brady, M.: Curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(1), 2–14 (1986)
12. Baruch, O., Loew, M.H.: Segmentation of two-dimensional boundaries using the chain code. *Pattern Recognition* 21(6), 581–589 (1988)
13. Wuescher, D.M., Boyer, K.L.: Robust contour decomposition using a constant curvature criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13(1), 41–51 (1991)
14. Cronin, T.M.: Visualizing concave and convex partitioning of 2D contours. *Pattern Recognition Letters* 24(1-3), 429–443 (2003)
15. Mokhtarian, F., Mackworth, A.K.: A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves. *IEEE Trans. Pattern Analysis and Machine Intelligence* 14(8), 789–805 (1995)
16. Abbasi, S., Mokhtarian, F., Kittler, J.V.: Enhancing CSS-based shape retrieval for objects with shallow concavities. *Image and Vision Computing* 18(3), 199–211 (2000)
17. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 711–732 (1995)
18. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: *Bayesian Data Analysis*. Chapman and Hall, London (1995)
19. Feder, J.: *Fractals*. Plenum Press, New York (1988)
20. Valdes, D., Bhalerao, A.: Local Shape Modelling for Brain Morphometry using Curvature Scale Space. In: McKenna, S., Hoey, J. (eds.) *Proceedings of the 12th Annual Conference on Medical Image Understanding and Analysis 2008*, British Machine Vision Association, pp. 64–68 (July 2008)