TRACKING MOVING OBJECTS WITH COEVOLUTIONARY SNAKES

P Liatsis and C Ooi

Control Systems Centre, Dept of EE&E, UMIST, United Kingdom

Abstract: *A new Symbiotic Genetic Algorithm (SGA) based Active Contour Model (Snake) is proposed to track the B-Spline contour of obstacles. It exploits the local control properties of the B-Spline to decompose the contour into subcontours and optimizes each subcontour in separate Genetic Algorithms (GA). Unlike GA-based Snake, a SGASnake can track the obstacles outline more robustly. Application-specific inter-population genetic operators are introduced to reinforce the symbiotic relationship via migration of genetic material. The use of symbiosis dramatically reduces the combinatorics of the search space, when compared to GAs. Results of tracking objects in real road scenarios demonstrate its robustness to noise and stability of convergence when compared to its GA counterpart.*

1. INTRODUCTION

Active Contour Models or Snakes were first proposed by Kass et al (1) as a framework to locating objects in images by deforming a contour to fit the actual outline of the object. Its strength is in its ability to maintain the geometric properties of the contour, while being deformed by the features of interest. A good underlying optimization algorithm is required for the snake to realize its full potential. Various algorithms have been proposed with their relative strengths and weaknesses. Here, a new Symbiotic Genetic Algorithm based snake is proposed to overcome some of the weaknesses of existing algorithms.

This paper commences by first explaining the Active Contour Model. Next, the advantages of using a cubic spline model as its internal shape model are highlighted. Then, the strength and weaknesses of different optimization methods are discussed. Extension of a GA into Co-evolutionary Genetic Algorithms is then discussed. The remainder of the paper concentrates on Symbiotic Genetic Algorithm Snakes (SGASnakes). It will be shown that the use of cubic splines allows the contour to be broken down into subcontours which are then optimized by separate Genetic Algorithms working in parallel. The design of this SGASnake is presented together with description of its fitness function and new application-specific genetic operators. Tracking results with SGASnakes are then presented and compared to Genetic Algorithms(GAs). The paper concludes by outlining the major advantages gained by the use of SGASnakes.

2. ACTIVE CONTOUR MODEL

Let us consider a contour represented by v(s)=(x(s), y(s)). The aim is to find the contour that minimizes the Snake Energy (adapted from (1)):

$$E_{snake}(v(s)) = \sum_{i=1}^{N} \left[E_{int}(v_i) + E_{img}(v_i) \right] \qquad (1)$$

where $E_{int}(v(s))$ and $E_{img}(v(s))$ are the Internal Energy, Image Energy and Constraint Energy, respectively. The discrete case assumes N contour points, denoted by $\mathbf{v_i}$. $\mathbf{v(s)}$ is also known as the internal shape model of the snake.

The internal energy maintains the shape of the contour and consists of the first two derivatives of $\mathbf{v(s)}$:

$$\mathrm{E}_{\mathrm{int}} = \sum_{\mathrm{i=1}}^{\mathrm{N}} \left[\alpha_{\mathrm{i}} |\mathrm{v}'_{\mathrm{i}}|^{2} + \beta_{\mathrm{i}} |v''_{i}|^{2}\right] \tag{2}$$

where α_i and β_i are positive coefficients. The first term makes the contour smooth while the second makes it rigid; they are referred to as Membrane Energy and Thin Plate Energy.

Image Energies are energies derived from useful features, such as edges. They are used to attract the contour to its final location and are collectively known as Image Energy. One popular image energy is the Gradient Energy, $\mathbf{E_{gradient}}$ and can be defined as in (1):

$$\mathrm{E}_{\mathrm{intensity}} = -|\nabla I(x,y)|^{2} \tag{3}$$

where $\mathbf{I(x,y)}$ is the intensity image.

3. INTERNAL MODELS

The internal model of the snake controls the deformability of the contour. Good internal contour models only permit acceptable deformation so as to minimize the probability of the contour being caught in spurious local energy minima.

The simplest model involves moving all contour points directly. However, due to the large number of contour points, the processing time is long. Undesirable contour features such as contour self-intersection frequently occur since every contour point can move independently. Alternatively, a polygon approximation of the contour can be used, where the vertices of this polygon are treated as the control points of the closed B-Spline curve. The actual points on the B-Spline curve, known as the interpolation points, are interpolated using these control points. The shape of the B-Spline curve is controlled by its knot sequence, the order of the underlying piecewise polynomial and the location of the control points. It is generally accepted that a 3rd degree B-Spline (cubic spline) is sufficient to represent all real shapes. A k-th order B-Spline is, in general, C^{k-1} continuous. This continuity, however, is influenced by the placement and multiplicity of the knots. The number of times, μ, a knot is repeated reduces the continuity of the curve at the knot to $C^{k-1-\mu}$.

With B-splines, each control point can only influence the shape of the curve in its local neigbourhood (see Figure 1(a)). This is exploited here to break the contour down into subcontours. This enables the complete curve $\mathbf{v(s)}$ with $\mathbf{m}$ control points to be decomposed into $\mathbf{m}$ subcontours, each defined by 4 control points.

4. COMPUTATION MODELS

There are various algorithms used to deform the contour to fit the actual object outline. Kass et al (1) proposed the use of Euler-Lagrangian Method, where the energies are converted to forces moving the contour in the appropriate direction.

The problem of finding the best shape can be viewed as finding the best combination of contour points location. However, combinatorial explosion makes it infeasible to test all possible combinations. In Amini et al (2), dynamic programming is used to manage this combinatorial explosion by optimizing the contour one segment at a time. Its advantage is the possibility to enforce hard constraints not considering segments that violate them. However, as other segments are stationary while a segment is optimized, the final contour is potentially not the best possible contour.

Genetic Algorithm (GA). GA manages combinatorial explosion in a different way. It encodes solutions as chromosomes and maintains a population of chromosomes. Better chromosomes (parents) in the population are used to generate new chromosomes (children) which are used to replace worse chromosomes. As the children are statistically more likely to be fitter than their parents, the overall fitness of the population increases. Repetition of this process allows the population to converge to a final solution. This process is known as the evolution of the GA. As each chromosome represents a search point, a GA can be viewed as a multi-point search algorithm which is more likely to find the global optimum results.

The chromosome for a GA-based snake encodes the actual location of individual control points (MacEachern and Manku (3)), hence, there is no ambiguity on the contour location. Since each chromosome represents a complete contour, there is no potential bias to particular shapes. Seo and Inoue (4) allow the GA to switch between two different search phases during its evolution process. These different phases have different settings to either emphasize the exploration of the search space or the genetic material in the population. However, in both of these applications, the size of the neighbourhood is considerably small [5x5].

Large contour movement is desirable because it allows the contour point to escape from the localized energy pits created by noise. It also means that the final snake contour can be found in just one evolution. In existing GAs, larger movement is achieved by optimizing the contour indirectly, i.e., optimizing the parameters of the contour such as an encoding of the polygon in Toet and Hajema (5) or Fourier Descriptors in Undrill et al (6).

Symbiotic Genetic Algorithms (SGAs). A Genetic Algorithm mimics the natural evolution within a species (population) using the survival of the fittest strategy. Co-evolutionary Genetic Algorithms (CGAs) aim to exploit interaction between different species by evolving multiple populations in parallel and having their evolution affected by that of others. The particular CGA of interest is the SGA, where all populations are required to co-operate to achieve the best results.

5. SYMBIOTIC GENETIC ALGORITHM SNAKE (SGASnake)

Here, a SGA is used to optimize the internal contour model (cubic spline) of the snake. This is accomplished by decomposing the contour into a number of subcontours, each defined by four control points. Assuming a neighbourhood of height **H** and width **W**, the search space is reduced to $\mathbf{n(HW)^4}$ from $\mathbf{(HW)^n}$ for the GA-based snake. Typical values for **H**, **W** and **n** are 30, 30 and 13, respectively.

5.1. Chromosome Design

We use a similar approach to MacEachern and Manku (3). Individual control points can move in a neigbourhood window. Each location in the window is given an index. Because of their size, the movement windows overlap. The chromosome is an integer string containing the index value for each of the 4 control points, representing the location they moved to. Since the chromosome has large gene values per gene, the genetic information is concentrated in the gene values and is not spread thinly along the length of the chromosome.

5.2. Reproduction Operators

There are essentially two categories of reproduction operators: the intra-population and the inter-population reproduction operators. The former exploit genetic information in the population, while the latter facilitate transfer of genetic material between populations. The latter are application-specific operators designed to enhance the symbiotic relationship.

Intra-population Reproduction Operators. The cross-over operators take two parent chromosomes and generate two children. Three cross-over operators are used. In uniform cross-over, the child is generated by randomly selecting either of the parents to supply the required gene value and the children are the complement of each other. This operator is very effective in exploring the search space. Arithmetic and Line Arithmetic cross-over operators (Michalewicz (9)) are chosen for their ability to exploit genetic information in genes that can take on a large number of values.

To improve the chances of generating children that are different from their parents, three chromosomes are selected and the two with the largest average distance between control points are chosen as parents.

The mutation operator randomly changes the value of one gene to one of an acceptable value for the gene.

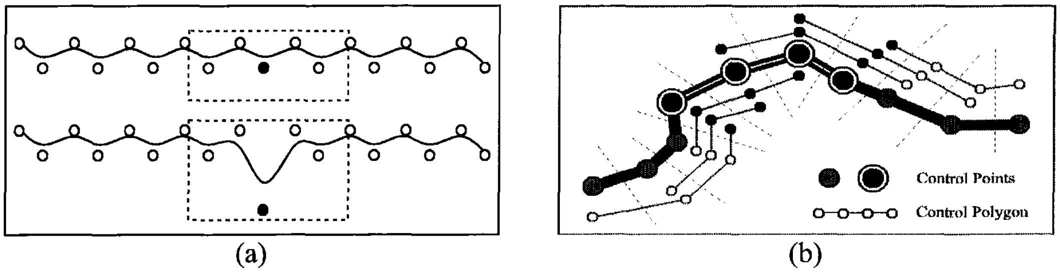


Fig. 1. (a) Local Control Property of the B-Spline. Movement of one control point (black) only changes the shape of the curve in its local neighbourhood, indicated by the rectangle. (b) Circles connected by lines represents the control points optimized in different populations. Those coloured black denote the control points of reference population.

Inter-population Reproduction Operators. The inter-population operator exploits the fact that every segment shares at least one control point with the segments before and after it (Figure 1(b)) to transfer genetic material between populations. This is desirable as it increases the chances that individual segments will place the control points near to each other. Even with an appropriate co-evolutionary fitness function, as the populations start to converge, as a compromise to other energy terms, some shared control points may converge to different points. Migration of these control points allows these different convergences to be compared and a compromise to be reached, decreasing the chances of the control points being stuck in local optima.

To simplify discussion, populations that share control points with the population of interest (*reference population)* will be known as the *candidate populations.* Chromosomes from the *reference* and *candidate* populations are known as the *major* and *minor* chromosomes or parents, as appropriate.

With *Inter-population cross-over*, a *minor* parent is selected from a randomly selected *candidate population*, to be cross-over with a *major* parent. The *major parent* is split at a randomly chosen location such that a valid child can be generated by substituting one of its ends with an appropriate segment from the *minor parent*.

The *Inter-population mutation* operator randomly selects one shared control point index from a *minor* chromosome to replace that of the *major* chromosome.

In *Inter-population replacement*, one *minor parent* is selected from *candidate populations* before and after the *reference population*, respectively. A site is randomly chosen from the overlapped region between the two *minor parents*, and both parents are split and combined to form a new valid chromosome for the *reference population.*

Inter-population cross-over changes the shape of one end of the segment, inter-population mutation changes the shape of the middle part of the segment while inter-population replacement generates a new curve segment based on the result of the optimization from two *candidate populations*.

6. FITNESS FUNCTION

The fitness function consists of several energy terms. There are two distinct parts in the fitness function: A self-fitness part enabling the comparison of different chromosomes in the same population and a SGA-fitness part measuring the fitness of the chromosome as part of the overall solution.

Self-fitness function. The image energies are ideal self-fitness terms as they can be determined without the aid of other segments on the contour. In the proposed application, only the Gradient Energy is used. The binary gradient image used is the Chamfer Image (Borgefors (7)) of the binary edge image generated using the Canny edge detector.

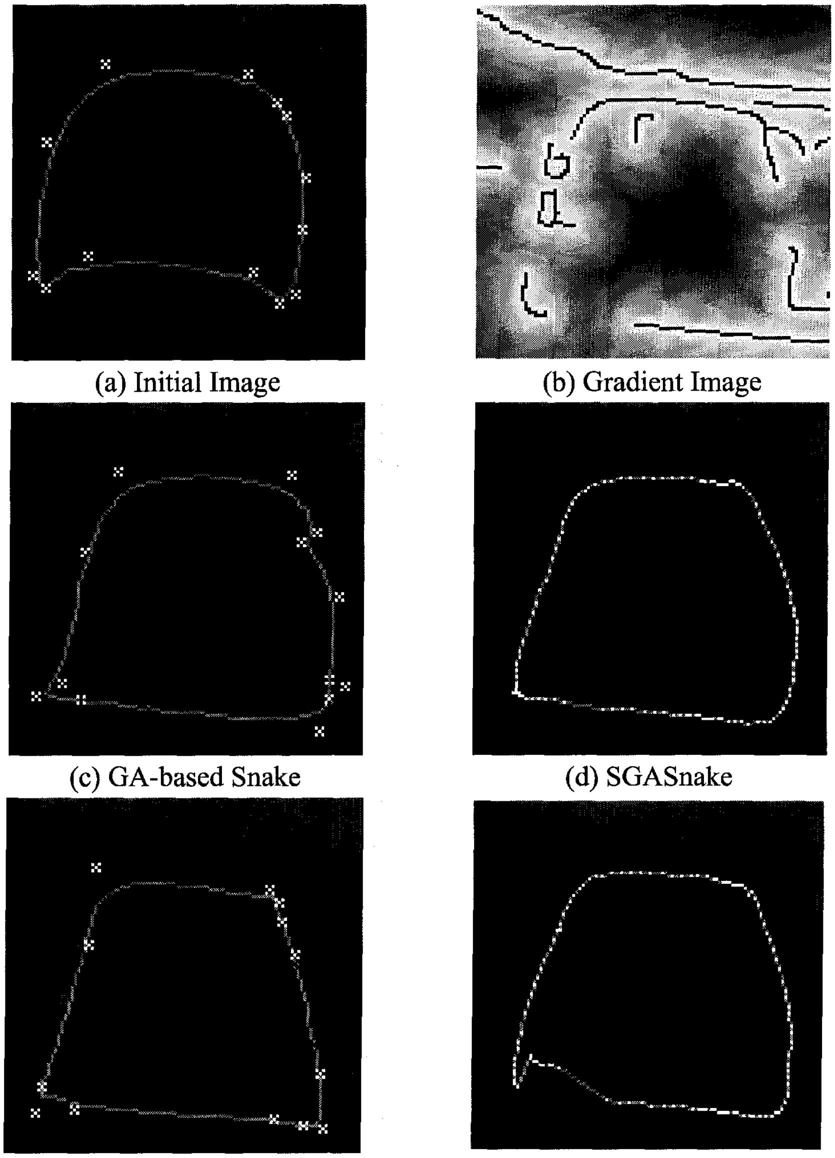
SGA-fitness functions. The first SGA-fitness function term minimizes the Euclidean distance between the control points of the current chromosome with the best chromosome in all other *candidate populations*. The second term minimizes the distance between the end points of the subcontour represented by the chromosome in question and that of the best chromosome in the neigbouring populations. The last term constrains the contour shape by counting the number of Fourier Descriptors of the contour, constructed using the current chromosome with the best chromosome from other populations, which are only a small variation (±10%) of that for the initial contour. It replaces the internal energies in eq. (2) which is ineffective if large contour movements are permitted.

Each measurement in every term and all the terms are combined together using the product operator to form the fitness value for the chromosome.

The SGASnake is a multivariable function optimization with each subcontour as a variable. As the inter-dependency of the variable is high, as predicted by De Jong and Potter (8), the optimization results improve when the algorithm computes a second fitness function, determined with random chromosomes from the *candidate populations* replacing the best chromosomes, and takes the best of the two as the chromosome's fitness value since this reduces the bias of the GA towards the best contour in the population.

7. EVOLUTION STRATEGY

Individual populations are evolved using the *modGA* method described in Michalewicz (9). The population size is 200 and is evolved for 50 generations. The best chromosome, together with another 139 selected chromosomes, is copied to the next population. The remainder of the population is generated using either the cross-over operator or the mutation operator but not both.



(a) Initial Image

(b) Gradient Image

(c) GA-based Snake

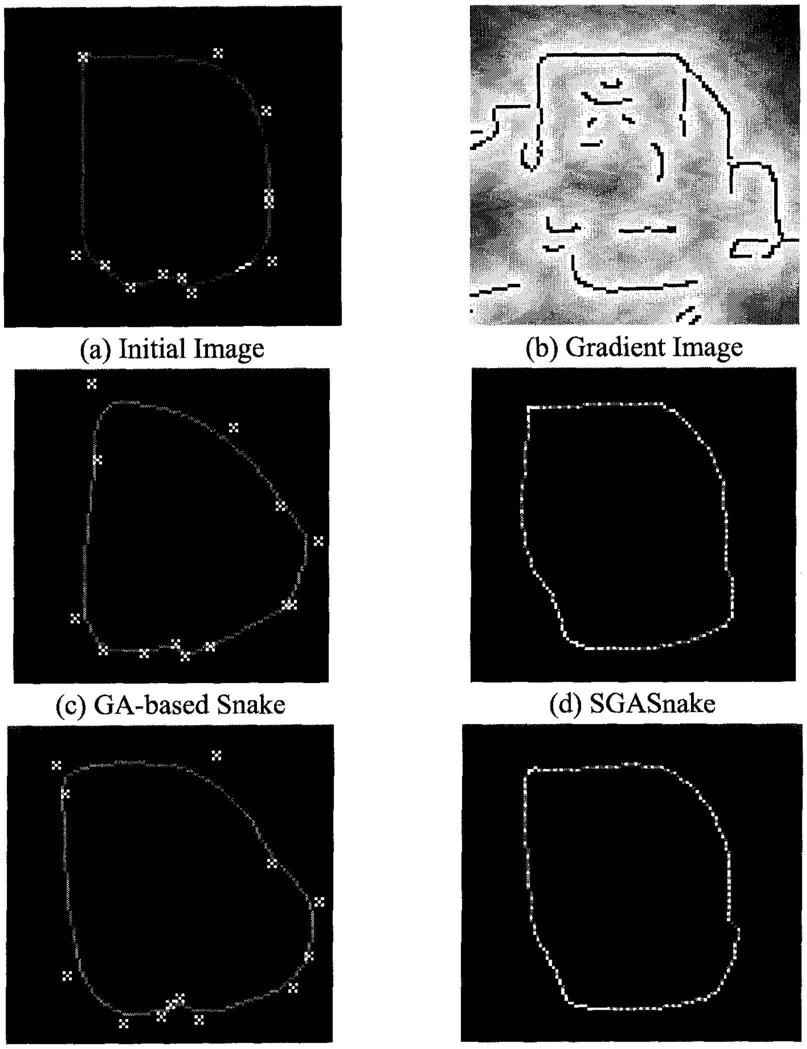
(d) SGASnake

(e) GA-based Snake

(f) SGASnake

Fig. 2. Tracking Results (1). (c) and (d) are the tracking results for the last frame (frame 8). (e) and (f) are the worst results.

The selection technique used is the two-player tournament selection. Initially, 80% of the children is generated by intra-population with the remainder by inter-population processes. This decreases linearly to 20% at the end of the evolution process. This allows the populations to find their optimal points before concentrating on generating a closed curve at the end. The Intra-population cross-over to Intra-population mutation ratio ([0.1,0.9]) is initially set at 0.6, reflecting the preference to cross-over over mutation. However, when cross-over is not effective in generating new solutions, the ratio changes to allow more mutation. At every generation, before new chromosomes are generated, when the best chromosome in the current and previous population is the same, the cross-over is assumed to be ineffective and the ratio decreases by 0.1 and vice-versa. If Intra-population cross-over is to be used, the ratio for Uniform cross-over : Arithmetic cross-over : Line Arithmetic cross-over is 2:1:1. There is no preference between the three Inter-population reproduction operators.



(a) Initial Image

(b) Gradient Image

(c) GA-based Snake

(d) SGASnake

(e) GA-based Snake

(f) SGASnake

Fig. 3. Tracking Results (2). (c) and (d) are the tracking results for the last frame (frame 8). (e) and (f) are the worst results.

8. RESULTS

The SGASnake is simulated on a serial computer to track the contour of a car and a truck. The control points of the snake contour are the critical points of the vehicles in the first image obtained using the critical point detection process in Ooi and Liatsis (10). The results from a GA-based snake with eq. 1 as the fitness function but with the internal energy replaced by a similar Fourier Descriptors based energy are also shown for comparison. The contribution made by the SGA is the stability of tracking where there are minimal variations between different contours in the 8-image sequence. The tracking results for the truck also suggest that the snake is not attracted to the strong edge at the right hand side, which is the edge of another vehicle. The performance in the presence of noise is also improved.

As for the processing time incurred, on a serial computer (Intel Celeron 500MHz, 128 MB RAM), the average time for each GA is only 133s, which is only an extra 30s compared to its GA counterpart. As the SGASnake can be easily converted for parallel processing, this performance is satisfactory.

9. CONCLUSIONS

This paper presented a new algorithm for obstacle tracking using SGASnake. Unlike most algorithms optimizing the snake contour directly, it allows for large contour movement thus making it possible to find the final contour in one evolution. Large contour movement is also desirable because it allows the contour to move away from spurious energy pits created by noise. Application-specific genetic operators were introduced to allow genetic material migration and reinforce the symbiotic relationship. A new Fourier Descriptors based energy is proposed to replace the internal energy. Compared to a traditional GA-based Snake, the SGASnake produces more stable results and is more robust to noise. A further improvement in SGASnake is to incorporate motion-based energy in the tracking of the vehicles in order to further improve the performance of the system in the case of large contour movements.

REFERENCES

[1] M. Kass, A. Witkin and D. Terxopoulos, Snakes : Active Contour Models, *1st Int. Conf. Computer Vision*, 1987, 259-268

[2] A. A. Amini, T. E. Weymouth and R. C. Jain, Using Dynamic Programming For Variational Problems In Vision, IEEE Trans. PAMI, 12, 1990, 855-867

[3] L. A. MacEachern and T. Manku, Genetic Algorithms For Active Contour Optimization, *IEEE Int. Sym. For Circuits and Systems*, 1998, 229-832

[4] W. Seo and K. Inoue, Active and Robust Contour Extraction by Biphased Genetic Algorithm, *IEICE Transaction On Information And Systems*, E77-D, 1998, 1225-1232

[5] A. Toet and W. P. Hajema, Genetic Contour Matching, *Pattern Recognition Letters*, 16, 1995, 849-856

[6] P. E. Undrill, K. Delibasis and G. G. Cameron, An Application Of Genetic Algorithms To Geometric Model-Guided Interpretation Of Brain Anatomy, *Pattern Recognition*, 30, 1997, 217-227

[7] G. Borgefors, Distance Transfomations In Digital Images, *Computer Vision, Graphics And Image Processing*, 34, 1986, 344-371

[8] K. A. De Jong and M. A. Potter, Evolving Complex Structures via Co-operative Coevolution, *4th Conference in Evolutionary Programming*, 1995, 307-317

[9] Z. Michalewicz, *Genetic Algorihtms + Data Structures = Evolution Programs*, Springer-Verlag., 1992

[10] C. Ooi and P. Liatsis, Critical Points Detection Using Genetic Algorithms, *2nd ELMAR Int. Workshop on Video Processing and Multimedia Communications*, 2000, 169-174