

# Object Tracking and Segmentation with a Population of Artificial Neural Networks

Luca Mussi<sup>1</sup>, Riccardo Poli<sup>2</sup>, and Stefano Cagnoni<sup>3</sup>

<sup>1</sup> Università degli Studi di Perugia, Dipartimento di Matematica e Informatica,  
Via Vanvitelli 1, I-06123 Perugia, Italy

`mussi@dipmat.unipg.it`

<sup>2</sup> University of Essex, Department of Computer Science,  
Wivenhoe Park, CO4 3SQ Colchester, UK

`rpoli@essex.ac.uk`

<sup>3</sup> Università degli Studi di Parma, Dipartimento di Ingegneria dell'Informazione,  
Viale G. Usberti 181/A, I-43100 Parma, Italy

`cagnoni@ce.unipr.it`

**Abstract.** We present a study concerning the use of artificial neural networks for object tracking/segmentation in surveillance video sequences. A number of artificial creatures, conceptually, “inhabit” our image sequences. Their task is to explore the images looking for moving objects. Their behaviour is controlled by neural networks which are evolved with an Evolutionary Algorithm (EA). In our system we encourage the EA to evolve a population of individuals with behaviours which are as different as possible. In addition, we require populations to satisfy some minimal requirements necessary for the object tracking behaviour to emerge from the interaction of individuals. Population performance is evaluated on artificial video sequences and some results are discussed.

## 1 Introduction

Detection, tracking and segmentation of moving objects is a critical task in computer vision [1, 2] and, in particular, in surveillance applications. Even if many methods to deal with this problem have been proposed, at present, there are still some issues to be addressed before a really effective general purpose tracking system can ever be developed. Because of this, new techniques and paradigms are continuously being tested. Within these, computational intelligence methodologies such as Artificial Neural Networks (ANNs) have been employed.

While many papers, for example [3–6] use neural networks as global filters to identify certain pixel characteristics, and hence localize objects of interest, in recent work [7], a swarm intelligence approach has been used to achieve real-time tracking as the emerging property of collective behaviour.

In this paper we present an approach, inspired to a segmentation method originally proposed in [8], to object tracking/segmentation based on the use of artificial creatures which “inhabit” video sequences. These creatures can see a very small portion of the current frame through a retina centered on their position. They are also aware of relative distance and state of other individuals in

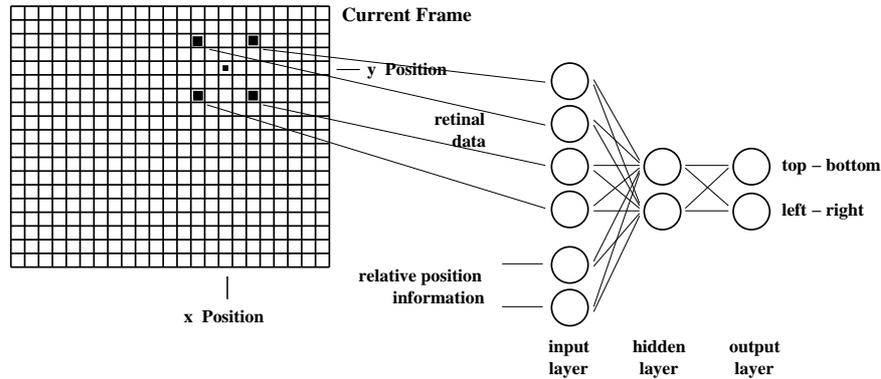


Fig. 1. Artificial creatures architecture.

the population, which we will also call *swarm* in the following. An ANN, acting as their nervous system, decides the next move each creature should perform based on current input information. The creatures' behaviour is evolved through a two-step Evolutionary Algorithm (EA). In a first phase we try to obtain individuals that satisfy a set of minimal behavioural requirements, expressed by a small specialised training set. These requirements effectively ensure that the creatures perform the main functions required of any tracking/segmentation filter. Then, a population, initially made up of clones of the best individual evolved in the first phase, is evolved. In this second phase the objective is to differentiate the behaviours of the individuals in the population as much as possible, while still demanding that they satisfy the requirements imposed in the first phase.

Tracking and segmentation are performed by the population as a whole. As we will show, differentiating creatures' behaviour provides better performances than when using a population of individuals with the same behaviour. With the right set of minimum requirements for the first phase and the differentiation provided by the second phase, the swarm has an emergent behaviour that results in finding and tracking a moving object, additionally providing a segmentation of the object itself.

The paper is organised as follows. In Section 2, we describe the architecture of individuals and their inputs. In Section 3, the evolution procedure is presented in detail. In Section 4, we report some results obtained with two artificial video sequences that confirm the effectiveness of our approach. Finally, in Section 5 we propose some future research directions.

## 2 System Architecture

Our approach to object tracking is novel and untested. As a start, in this paper, we will therefore focus on verifying its strengths and weaknesses in controlled conditions. So, we will limit our attention to surveillance applications based on a fixed b/w camera placed in an indoor environment with stable lighting

reference / population status	relative position	inputs	
no individuals on the moving object	-	0	0
closest creature on the object	top-right	-1	1
closest creature on the object	top-left	-1	-1
closest creature on the object	bottom-right	1	1
closest creature on the object	bottom-left	1	-1

**Table 1.** Coding of the two input values, connected to other individuals position, for a specific creature

conditions. In this situation, it is well known that the initial image of a video sequence can be taken as a reference background. Any moving objects can then be easily identified by computing the difference between the gray levels of the pixels in the current video frame and corresponding pixels of the background.

In our tracking system, a global “supervisor” keeps track of the position of all the population individuals. At every cycle, it provides data inputs to each individual and collects output responses to update their state. Furthermore, while retrieving input information from the frames, it also marks pixels as part of the tracked object in accordance with its classification policy. At the moment, the attention criterion utilized is simply a thresholding performed on the difference between the image and the background reference.

When a new video frame is available, the supervisor moves all the population members on it while keeping their positions unchanged.

Each artificial creature has a very simple feed-forward ANN as its “brain”. Through the input layer it senses environmental information while through the output layer it informs the supervisor about its next move. Note that each frame is interpreted as being toroidal. So, there are no restrictions imposed on an individual’s motion.

Figure 1 shows the structure of the network controlling a creature. The six input neurons are grouped into two sets. Four neurons form a minimalist retina that provides our creatures with vision. The sensing units are at the corners of a  $5 \times 5$ -pixel square, centered on the current position of the individual. The retina receives the following binary inputs:

$$I_j = \begin{cases} 1 & \text{if } |F(\mathbf{p}_j) - B(\mathbf{p}_j)| > \delta \\ 0 & \text{otherwise} \end{cases} \quad \forall j \in \{1, 2, 3, 4\} \quad (1)$$

where  $I_j$  is the input for the  $j$ -th sensing unit,  $\mathbf{p}_j$  represents the unit position inside the current frame  $F$  and in the background reference  $B$ . The parameter  $\delta$  is an attention threshold chosen by the user. The remaining two input neurons are used to provide the net with information about an individual’s position with respect to the rest of the population. They are coded according to two different states of the population (see Table 1): if all the individuals are still looking for a moving object, the inputs are both set to zero, otherwise the closest individual on the tracked object is taken as reference and the two inputs simply indicate the relative position of such an individual.

Inside the hidden layer there are only two neurons, both connected with all the inputs. Their activation function is the classical sigmoid

$$s(x) = \frac{1}{1 + e^x} \quad (2)$$

Finally, the output layer consists of two motor neurons connected with the units in the previous layer. Their purpose is to notify the supervisor about how to update the creature’s position at the next cycle. One neuron refers to horizontal movement while the other refers to vertical motion. The activation function used for the output layer is

$$o(x) = 2(W - 1) \cdot \text{round}(s(x) - 0.5) \quad (3)$$

where  $W$  is the width of the retina (five pixels in our case) and  $s(x)$  is once again the sigmoid function in Equation 2. As a result, the next position of an individual is always within a  $(2W - 1) \times (2W - 1)$ -pixel square centered on the present position.

### 3 Evolving Behaviours

Once the activation functions and the topology of an ANN have been chosen, outputs for a specific input rely exclusively on the values of connection weights. In the same way, the behaviour of our artificial creatures depends only on the connection weights inside their “brain”. We chose to represent them with real numbers. So, each individual will be described with a real vector. Unfortunately, with this representation the traditional genetic operators developed for binary strings can no longer be used to evolve the population. Evolutionary Strategies or Evolutionary Programming (EP) are the most suitable techniques to deal with real vectors [9].

When using an entire population as a swarm to track moving objects, it seems reasonable to think that it would be best to maintain a certain diversity between individual behaviours. This would permit the swarm to achieve better generalization and exploration capabilities [10]. At the same time, however, we want the swarm to have an efficient collective, swarm-type behaviour. This is usually obtained when each individual feels an attraction towards the center of the group and at the same time a repulsion from its neighbours [11].

As we explained above, to evolve our swarm we use a two-step evolution procedure. First we look for an individual that satisfies a small training set which exemplifies the correct behaviour in terms of attractions to and repulsion from other individuals, as we will explain in Section 3.1. When found, such an individual is cloned to form a new population further evolved to maximize behaviours differentiation, as we will describe in Section 3.2. This final population is then directly used in the tracking system. The final population size is a parameter chosen by the user.

INPUTS						OUTPUTS			
I0	I1	I2	I3	cloLR	cloTB	minLR	maxLR	minTB	maxTB
//Retina Activations only in one of the four corners									
1	0	0	0	d-	d-	-4	-1	-4	-1
0	1	0	0	d-	d-	1	4	-4	-1
0	0	1	0	d-	d-	-4	-1	1	4
0	0	0	1	d-	d-	1	4	1	4
//Retina with no activations									
0	0	0	0	1	1	1	4	1	4
0	0	0	0	1	-1	1	4	-4	-1
0	0	0	0	-1	1	-4	-1	1	4
0	0	0	0	-1	-1	-4	-1	-4	-1
//Retina full activated									
1	1	1	1	1	1	1	4	1	4
1	1	1	1	1	-1	1	4	-4	-1
1	1	1	1	-1	1	-4	-1	1	4
1	1	1	1	-1	-1	-4	-1	-4	-1

**Fig. 2.** Training set utilized in the first evolution phase. The “d-” symbol means that both 1 and  $-1$  are allowed in that position.

### 3.1 Minimum Behaviour

To provide attractions and repulsions as well as some other basic behaviours, we use a special training set. Figure 2 shows the training set used in our experiments. For each example we specify a set of inputs and a corresponding desired behaviour (outputs). The inputs are specified numerically: 0 or 1 for the inputs I0-I3 representing the retina, and 1 or  $-1$  for the inputs cloLR and cloTB which specify the relative position of the closest individual. However, the system also allows the use of a don’t “care symbol” d- meaning either 1 or  $-1$ . Considering the “don’t care” symbol, the training set in Figure 2 presents twenty-four input patterns out of the possible sixty-four. Note that for the two outputs (left-right and top-bottom motion) we do not specify exact target values but minimum (minLR and minTB) and maximum (maxLR and maxTB) acceptable values.

As can be seen, the first set of examples in the training set requires that when only one out of the four sensing units in the retina is activated, the individual should move towards the quadrant corresponding to the activated unit, irrespective of where other individuals are. With the other two sets of examples we ask that, when the retina has no activations, or it is fully activated, the individual moves towards the quadrant of its nearest neighbour on the tracked object. The supervisor keeps track of which individual is on the tracked object. It normally moves creatures according to their outputs. However, when an individual is fully inside the tracked object, it does the opposite of what it is asked to do. As a result, the creature is moved away from the center of mass of the swarm. We made this design decision because if we asked the net to directly distinguish these two cases (creature all inside and creature all outside the tracked object), the training set would be much more difficult to learn, particularly with only two neurons in the hidden layer. So, a more computation-intensive supervisor has been preferred to bigger nets.

This first evolution phase is based on the classical evolution strategy. At every generation, each individual creates an offspring by a Gaussian perturbation.

Then, in the survivor selection phase, round-robin tournament competitions are performed among both parents and offsprings: the individuals with the greatest number of “wins” are selected to survive. To calculate the fitness of an individual all the examples in the training set are provided to the individual’s neural network, one by one. The corresponding outputs are checked against the ranges specified by `minLR`, `minTB`, `maxLR` and `maxTB` entries in each example and the fitness is simply the number of output patterns within the required ranges.

This first evolutionary phase ends when an individual with a fitness value equal to the size of the training set appears or when a maximum number of generations is reached, in which case evolution is aborted.

### 3.2 Population Differentiation

Once an individual satisfying our set of minimum requirements is found, the second phase starts. A new initial population is formed. This is made up of copies of the best individual found in the previous phase. Then a new evolution process starts. In this phase, our goal is to differentiate as much as possible the behaviours within the population. This corresponds to checking, for each of the possible input patterns, that the distribution of the outputs obtained with all the creatures is as uniform as possible. Obviously we also need to check whether the training set used in the previous evolution is still satisfied. Therefore, the distribution of the outputs corresponding to the inputs appearing in the basic rules will typically be uniform inside a sub-region of the output domain.

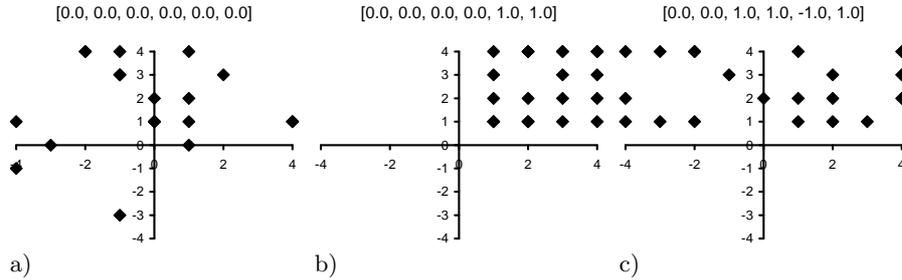
Like before, in each generation, every individual generates an offspring by means of Gaussian mutation. If the offspring satisfies the minimum requirements, it replaces its parent and the fitness of the entire population is re-evaluated. If fitness does not improve, the offspring is substituted with its original parent (thereby undoing the change) and the next offspring is generated and checked. The fitness function we minimize is

$$F = \sum_{i \in I} \sum_{m \in O} \sum_{n \in O} (1 - n_i(m, n))^2 \quad (4)$$

in which  $I$  is the set of all possible input patterns,  $O = \{-(W-1), \dots, (W-1)\}$  is the domain for the output of motor neurons and  $n_i(m, n)$  is the number of individuals that produce the output  $(m, n)$ , in the presence of input  $i$ .

This second evolutionary stage ends when a maximum number of generations, once again chosen by the user, is reached. The last population is saved and can then be used as a swarm for tracking experiments.

Figure 3 shows the responses to some input patterns produced by a typical population at the end of the second evolution phase. When all the individuals are still looking for the moving object, they receive an all-zero input. As one can see from the figure, however, despite the input being the same for everyone, each individual responds with a slightly different move. In Figure 3(a), all the responses of the population to the null input are plotted. When an individual is still in the search phase and there is at least one creature on the moving object, its retina does not sense anything, but the supervisor provides him with



**Fig. 3.** Responses of each of the twenty individuals in the Population to some input patterns: a) when all the population is in the search phase, each individual receives the null input; b) when an individual is outside the target and its closest neighbour on the object is below on the right; c) when an individual is on the object border with the bottom end of the retina activated. Note that the y coordinate on the image increases downwards and that some outputs could be overlapped.

the relative position of the closest neighbour on the object. As can be seen in Figure 3(b), every individual that is in this situation responds by moving towards the correct direction, but once again responses are differentiated. Finally, in Figure 3(c) all the possible responses for an individual with the bottom part of the retina activated can be seen. In these conditions the individual is probably on the object's upper border and it makes sense to move downward. It should be noted that this last situation is not included in our basic training set where only activations of the corners of the retina were included among the input patterns. Nonetheless, as a result of generalization and of behaviours differentiation, every response of the individual within the population is fully appropriate for the specific situation.

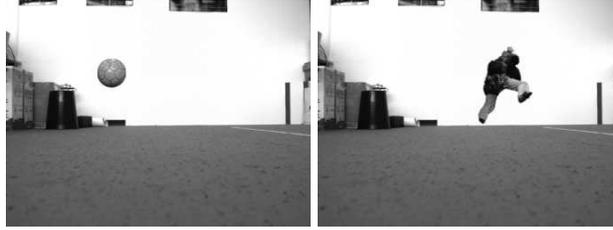
## 4 Results

The code for the system is written in Java(TM) using JINGPaC<sup>4</sup>, a package for Evolutionary Computation developed at the University of Parma.

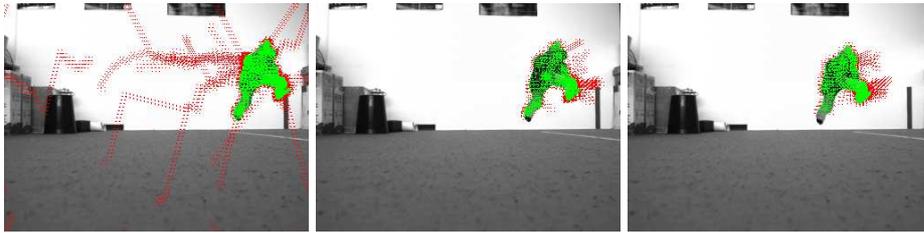
In the experiments, in the first evolution phase we used a population of 2,000 individuals and a maximum number of generations of 3,000. Tournament selection with tournament size 5 was adopted as our selection strategy. In the second phase, a population of 20 individuals was evolved for 40,000 generations.

To test our approach we used two artificially-created videos. The picture of a room has been used as a fixed background and two objects have been artificially superimposed and moved over it. In Figure 4 it is possible to see two sample frames taken from our two video sequences. In the one on the left, an object with a regular shape, a soccer ball, is the target to track. In the one on the right, an object with an irregular shape (a climber with a rucksack on his back) has been used as target. Experiments have been performed on an Intel(R) Pentium(R) 4 CPU 2.80GHz setting the frame rate of the videos to 10 fps.

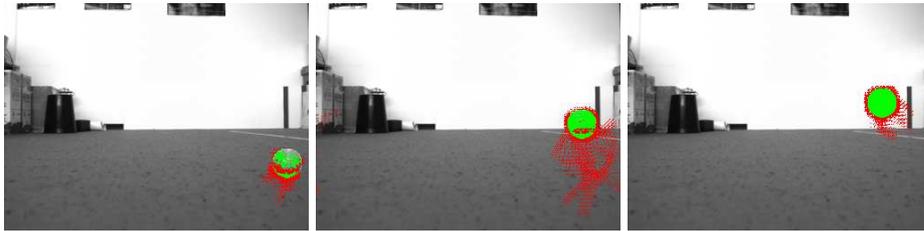
<sup>4</sup> Available at <http://jingpac.sourceforge.net/>



**Fig. 4.** Two sample frames extracted from the two videos used in the experiments. In the first sequence (left) a soccer ball moves around over a fixed background. In the picture on the right, the moving object is a climber with a rucksack on his back.



**Fig. 5.** The first three frames from the “climber” video. Green pixels represent the segmented part of the object while red points represent checked pixels which have not been marked as part of the object.



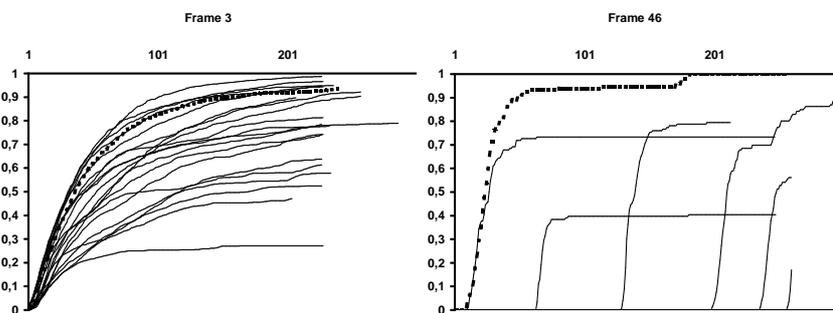
**Fig. 6.** Three frames from the middle of the “red ball” video. Green pixels represent the segmented part of the object while red points represent checked pixels which have not been marked as part of the object.

Figure 5 shows the first three frames of the “climber” video. On the left, the search behaviour of the population can be clearly observed, which eventually converges on the object. In subsequent frames, the man is tracked and almost completely segmented. It should be noticed that, due to colour similarity between his clothes and the background, it is sometimes not possible to identify part of the climber by simply thresholding the difference image.

In Figure 6 three frames extracted from the middle of the “red ball” sequence can be seen. Even if the target is moving fast, the swarm can easily track the ball. As can be seen, between the first and the second frame no parts of the ball

overlap: when the image is updated, no more individuals are on this target but, after a short initial search phase, the ball is found again and is fully segmented.

To test our hypothesis about behaviour differentiation, we performed another experiment from which some statistical data were collected. Each of the twenty individual in the final population has been used to form a new population containing twenty exact clones of such an individual. Therefore, being formed by identical individuals, each population was representative of a single behaviour. To see if actually differentiated behaviours were more effective than single ones, the resulting twenty “clone” populations have been tested on both sequences. For each frame and for each life cycle, the percentage of segmented area of the moving object has been recorded. The results are shown in Figure 7 for two frames of the “red ball” sequence.



**Fig. 7.** Comparison between the performance of the final population (dashed line) and other populations without behaviour differentiation on two sample frames from the “red ball” sequence. The plots represent the fraction of target object correctly segmented versus life cycle number.

In most frames the differentiated population has been the best performer, even if in few frames some individuals performed better (see Figure 7, left). In any case this experiment showed that individuals with the same behaviour are not capable of continuously tracking the target. In fact, as can be observed on the right in Figure 7, in many frames single behaviours failed to find the object or have been capable to segment only a minimal part of it. Furthermore, behaviours that in few frames performed better than the final population, in other frames, failed to segment the target, while this never happened with a differentiated population.

## 5 Conclusions and Further Improvements

In this paper we have described a novel object tracking system. It is based on a population of artificial creatures controlled by ANNs. As image inhabitants, they go around looking for a moving object, indicating to a global supervisor which pixels to check: if these pixels satisfy certain segmentation criteria, they are

marked by the supervisor as part of a moving object. Presently only a simple thresholding algorithm is used to decide whether a pixel is part of a moving object or not, but in the future we want to improve the system including learning of object characteristics during tracking. This would probably allow to identify most objects, not just the object obtained after computing the image difference.

The results obtained in experiments with artificial video sequences proved the effectiveness of our approach. Two moving objects, one of which with a complex shape, have been successfully identified and tracked by our swarm. Furthermore, in most available frames, almost the whole object has been segmented. Future tests will use also more challenging real video sequences.

## Acknowledgements

This work is partially supported by the University of Parma, within the FIL project “Metodi di Swarm Intelligence per l’analisi di immagini”.

The authors gratefully acknowledge the support received by Matteo Sacchi and Federico Sassi, authors of JingPAC.

## References

1. Haralick, R., Shapiro, L.G.: Computer and Robot Vision. Addison Wesley, New York (1992)
2. Ballard, D.H., Brown, C.M.: Computer Vision. Prentice-Hall, Englewood Cliff, NJ (1982)
3. Cuevas, E., Zaldivar, D., Rojas, R.: Lvq color segmentation applied to face localization. In: 1st International Conference on Electrical and Electronics Engineering (ICEEE), Los Alamitos, IEEE Computer Society Press (2004) 142–146
4. Bengtsson, M.: A neural system as a dynamical model for early vision. *Neural Networks* **6** (1993) 313–325
5. Valli, G., Poli, R., Cagnoni, S., Coppini, G.: Neural networks and prior knowledge help the segmentation of medical images. *Journal of Computing and Information Technology (CIT)* **6**(2) (1998) 117–133
6. Kulkarni, A.D.: Artificial Neural Networks for Image Understanding. VNR Computer Library. Van Nostrand Reinhold, New York (1994)
7. Anton-Canalis, L., Hernandez-Tejera, M., Sanchez-Nielsen, E.: Particle swarms as video sequence inhabitants for object tracking in computer vision. In: ISDA '06: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, Washington, DC, USA, IEEE Computer Society (2006) 604–609
8. Poli, R., Valli, G.: Neural inhabitants of MR and echo images segment cardiac structures. In: Computers in Cardiology, London, IEEE Computer Society Press (1993) 193–196
9. Eiben, A., Smith, J.: Introduction to Evolutionary Computing. Springer-Verlag, Berlin, Heidelberg (2003)
10. Yao, X.: Evolving artificial neural networks. *Proceedings of the IEEE* **87**(9) (1999) 1423–1447
11. Couzin, I., Krause, J., Franks, N., Levin, S.: Effective leadership and decision making in animal groups on the move. *Nature* **433** (2005) 513–516