In Advances in Artificial Life: Proceedings of the Sixth European Conference on Artificial Life (ECAL01), Springer Verlag, 2003.

A Visually-Based Evolvable Control Architecture for Agents in Interactive Entertainment Applications

Andrew Vardy

Computer Science Deptartment
Carleton University, Ottawa, Canada
avardy@scs.carleton.ca

Abstract. A visually-based evolvable control architecture for agents in interactive entertainment applications is presented. Agents process images of their local surroundings according to evolved image processing operators. A behaviour-based framework of *action rules* encapsulates image processing parameters and action parameters into discrete behavioural modules. These modules are interconnected and retain internal state through the dynamics of these internal connections. This novel control architecture has a wide behavioural range and is specified in an evolvable framework which allows agents for entertainment applications to be evolved as opposed to explicitly designed. The results of several demonstrations and experiments are presented to showcase the possibilities of this control architecture.

1 Introduction

We present an evolvable control architecture for computer-controlled agents in interactive entertainment applications. The proposed control architecture is intended to support a wide range of behaviour of the type often sought after by computer game designers and animators. The focus is on behaviour that is more reactive than deliberative. 'Pursuit', 'evasion', 'obstacle-avoidance', and 'wandering', are examples. The control architecture presented here has the capacity to implement these kinds of behaviours individually, in combination, and in sequence and it allows these multitudinous permutations of steering behaviours to evolve using a genetic algorithm (GA). The task to which we set an individual agent involves playing a game of survival against a hostile opponent agent which represents the human player in a video game. Artificial evolution is envisaged to take the place of explicit programming work via the *off-line* (i.e. pre-release) production of sophisticated behavioural controllers.

Current controllers apply a mix of rigid sequential behaviours along with randomized movement patterns to produce their character's behaviour [15, 16]. These mechanisms are difficult to program, inflexible, and non-adaptive. They rely heavily on the human tendency to anthropomorphise [16]. In order to provide a viable alternative to current game AI techniques our system must support

controllers that are challenging and interesting to human players. The only information available to an agent in our system is a ray-traced image of its immediate surroundings. It cannot see behind walls or consult global data structures and is therefore more intuitively understandable. Other intended attributes of this system are comprehensibility of evolved controllers and low computational burden. The control architecture described here is tailored towards games where the principal relationships existing between agents are spatial—as opposed to games where various internal and external resources need to be managed. Commercial games such as Quake, Tomb Raider, Baldur's Gate and numerous others are prime candidates for the kind of agent controllers described here. Note that most ostensibly three-dimensional games are actually fundamentally two-dimensional in nature.

The next section of this report provides an overview of the core concepts of the control architecture followed by a discussion of related work. We then go on to describe a number of refinements on the core architecture before presenting experimental results. A concluding section provides commentary on the results as well as future directions. Note that space limitations necessitate a highly selective and condensed style of coverage. Please see [14] for a more complete discussion.

1.1 System Overview

Agents occupying a cell in a two-dimensional grid apply evolved image processing operations on one-dimensional images of their local surroundings. Parameters for image processing are specific to each action rule. These parameters include the *color translation table* and the *convolution mask*. The action rule also includes parameters for action rule interconnection and a parameter that indicates what action the agent should take when it is active. The direction of action is determined by the image processed by the winning action rule. The pixel in this image with the highest response gives the direction. An agents entire genetic code is comprised by concatenating the codes for its constituent action rules. The code is comprised of integers, each in the range [-3, 3]. For the experiments below all agents have four action rules. See figure 1 for illustration of each of the following processes:

Image Generation 1-D *retinal images* are generated by tracing rays out from the agent to all points on the circumference of a rasterized octagon. The agent's field of vision is omni-directional and range-limited and exhibits such phenomena as occlusion and perspective.

Color Translation The values of various grid objects in the retinal image are used as an index into the color translation table. Application of this lookup table yields the pre-processed or translated image.

Convolution The convolution mask for the active action rule is applied to the pre-processed image. It is an array of values swept along the image and multiplied

with image elements at corresponding positions, with the sums taken as the processed image [1]. Circular convolution is applied so that the original and processed images maintain the same length. The mask is also applied in reverse, generating an additional processed image (the image with the lower maximum is discarded). This allows masks to be orientation independent.

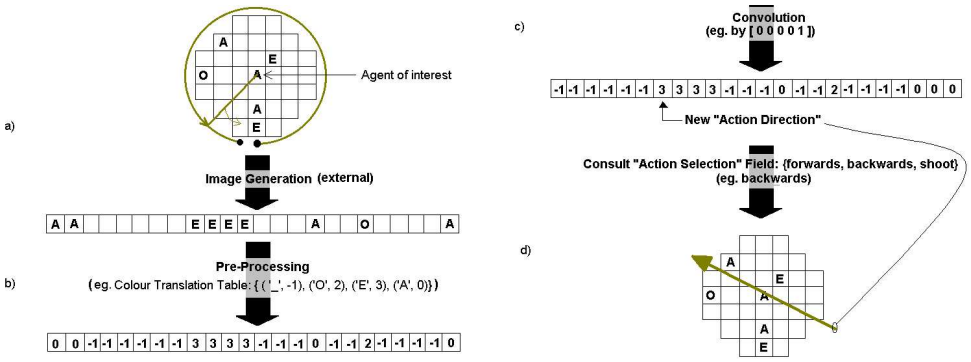


Fig. 1. Agent processing: (a) Image generation (obstacles=O, enemies=E, agents=A). (b) Colour translation. (c) Convolution. (d) Action direction selection and action selection.

Action Rule Selection The current retinal image is processed by all action rules to determine which rule should become active. The maximum value from each of these processed images must exceed a threshold value for the corresponding action rule to be further considered. Each action rule has an activation level which decays to zero at an evolvable rate. The currently active action rule has a weight field which is added to another action rule (possibly itself) selected according to a displacement field. An importance field is added to the activation level to yield the overall strength. The rule with the highest strength is then selected as active. An action role also contains an action selection field indicating one of {accelerate, decelerate, null, launch projectile}. A refractory period exists for firing projectiles such that a 'shooting' action rule is prevented from becoming active for a certain fixed period after the agent has launched a projectile.

Action Direction Selection Retinal images are generated such that their left-hand edge corresponds to the direction immediately beneath the agent. Therefore the highest valued pixel in the processed image corresponds uniquely to a particular action direction. The highest valued pixel location closest to the last action direction is chosen as the new action direction.

1.2 Related Work

Applications such as *Creatures* [9] allow the user to educate and assist virtual animals in their growth and exploration of a virtual world. See [9] for a review

of commercial artificial life related entertainment applications. Funes *et al.* describe a system whereby Internet users play an on-line game of 'Tron' against a community of automatic players adapting via genetic programming [8]. Reynolds [13] designs autonomous agents explicitly for behavioural animation and games. He presents techniques designed to navigate agents through their worlds in a "life-like and improvisational manner" using *steering behaviours* such as seeking, evasion, obstacle avoidance, and various others. This contrasts with our approach which is to find an evolvable control architecture within which all of these types of behaviour can find expression.

A study on visual processing in the fly [6] describes a hypothesized early layer in fly visual system where the entire visual field is processed by local operators known as Elementary Motion Detectors (EMD's), similar in spirit to the convolution masks used here. Also, a full-length convolution mask is essentially an image matcher, similar in concept to the idea of retinotopic images hypothesized for landmark identification and homing in insects [5]. This system adheres to a behavioral decomposition of behaviour—specifically, one whose behavioural modules are evolved [7] [11], rather than explicitly engineered [2].

The use of artificial evolution for image processing has been attempted in various forms by a number of researchers. Poli employed genetic programming to develop operators for image segmentation and feature detection using medical imagery as an example [12]. Of particular interest is work by Harris and Buxton who use genetic programming to evolve linear filters for edge detection [10]. The end product of their system is a symbolic function which is sampled to produce a convolution mask. Here we evolve the masks directly.

1.3 Refinements

Range Thresholds Early experiments revealed that the lack of range information was a severe impairment. A table of *range thresholds* was added as an image pre-processing stage. The distance of objects from the agent is thresholded according to an entry in this table corresponding to the object type. If the object's distance is greater than threshold then the pixel is translated as 'empty'. Figure 2 shows a trail of an agent run by only a single active 'wandering' action rule. The wander behaviour is simply achieved with a colour translation table that gives a high weight to 'empty', has a somewhat long and flat convolution mask (i.e. {11111111111111111111}), and a forwards action selection field. Wandering behaviour with and without a range threshold on 'obstacles' is shown. Clearly, the intended behaviour is better achieved with a range threshold.

Differential Processing The ability to predict the future direction of a moving target is provided by *differential processing*. If an agent's genotype specifies that this feature is active, then the current retinal image is pre-processed via range thresholding and colour translation as usual. The previous retinal image is then subjected to the same pre-processing operations and is subtracted from the current pre-processed retinal image. Figure 3 illustrates simple pursuit and predictive pursuit (with and without differential processing, respectively). A single

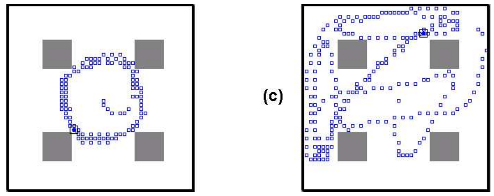


Fig. 2. Two 'wander' behaviours: (a) Without range thresholds; (b) Range thresholding obstacles at distance of 5. Grey cells and borders are 'obstacles'. Unfilled squares indicate past positions of agent. Final position indicated. Duration = 250 iterations.

forwards action rule implements pursuit. For simple pursuit the rule's convolution mask is simply '1'. For predictive pursuit the mask used is the 48-long string {-1, -1, -1, -1, -1, -1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}. Retinal image length is 224. The -1's match the previous position and the 1's match the current position while the 0's pad out the mask so that the best match occurs in the center—accelerating the agent towards the target's approximate future position. Differential processing supports other features such as predictive firing and movement detection.

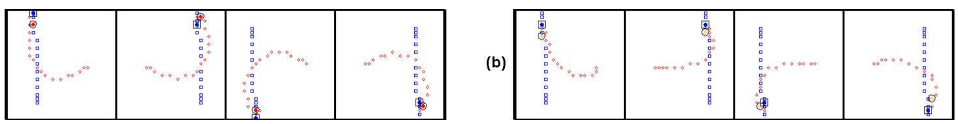


Fig. 3. Two 'pursuit' behaviours: (a) Simple pursuit without differential processing; (b) Predictive pursuit with differential processing. Unfilled diamonds indicate the path of the pursuer. Unfilled squares indicate the path of the evader. The evader's behaviour here is blind and unchanging.

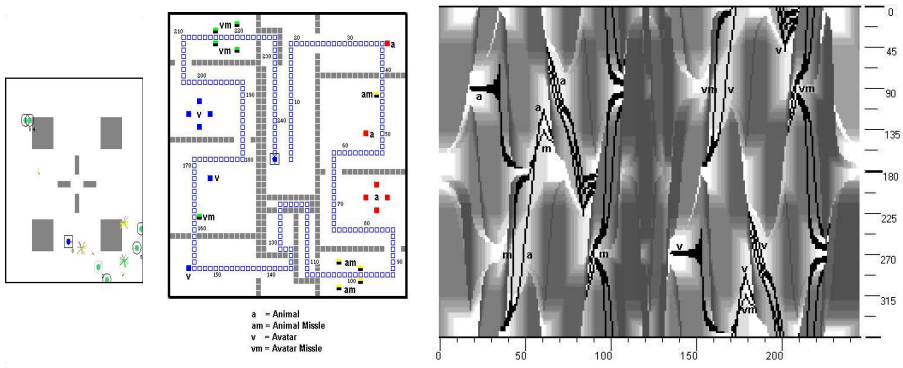
2 Experiments

A collection of agents, known as the *animals* is placed in a bounded arena together with a competing agent known as the *avatar* (representing a human player). Launched projectiles are damaging only to the opposing species. There are five types of objects in the world: obstacles, animals, animal missles, avatars, avatar missles. All objects occupy a full grid square. Properties of the physics model are: Velocity proportional to force; Viscosity; Random perfectly elastic collisions. The phase of agent 'thinking cycles' is randomized so that only half of the population will be thinking on any given iteration.

For evolutionary experiments the individual being tested is duplicated (six times for animals, once for the avatar) into the arena and evaluated via competition with the adversary species. For co-evolutionary experiments the avatar co-evolves with the animals. Each is evaluated against the best individual from the

previous generation, as in [4]. For single-species evolution the evolving animals are pitted against a hand-coded avatar. The fitness function rewards damage done to the opposing species, as well as high survival time for the losing species and low elimination time for the winning species (damage factor outweighs time factor) [14]. Fitness is averaged from three evaluations with half of the standard deviation subtracted to obtain the final score for an individual. A steady-state tournament selection style GA is applied with a tournament size of three for both reproduction and removal. Two-point crossover is used to create a single new population member which replaces an existing member. All population members are subjected to creep and jump mutation. Figure 4(a) illustrates the evaluation environment. The avatar begins its life in a randomized location near the center while the animals begin in randomized locations around the perimeter.

Visualization A particular environmental configuration known as *the course* has been designed to present an agent with a rich variety of stimuli, but in a standardized and regulated fashion. A *movie* was recorded of the retinal images presented to a dummy agent as it was moved throughout the course. Figure 4(b) illustrates the course and the path taken by the dummy recording agent as it passed through it. The recorded movie is presented in figure 4(c) as an array of vertical one-dimensional images flowing from left to right through time.



(a) Evaluation of animals (circles) vs. avatar (square).

(b) *The course* overlaid with the dummy recording agent's path through it. The trail is annotated with the frame number.

(c) Movie recorded by dummy agent. Frame number on horizontal axis. Vertical axis gives angular position in degrees where 0^o represents the direction directly beneath the agent. Black shows indicated type. Grey shows obstacles with grayscale proportional to distance from agent.

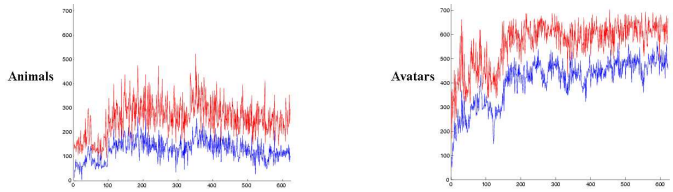
Fig. 4.

2.1 Results

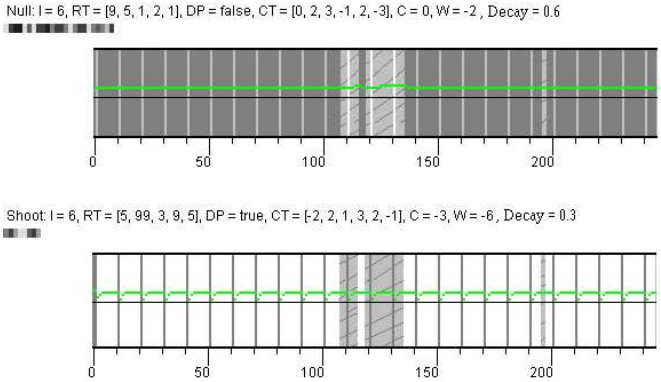
Currently, the only meaningful way of investigating the results of an evolutionary run with this system is qualitative description with the aid of the visualisation tool just presented. Some trials with human players were attempted but no conclusive results could be drawn—likely because of small sample size [14]. Therefore, there is only sufficient space to describe a single run. Figure 5(a) shows the fitness profile for evolved animals and avatars for the coevolutionary run, KA24. These figures reveal the ambiguity of measuring progress in coevolutionary simulations. It appears that some form of competitive coevolution is occurring, but this is not certain. Figures 5(b) and 5(c) shows the decoded action rules for the best members from the final generation (621) of animals and avatars, as well as the activity of those rules when presented with the movie. Figure 6 shows the sequence of processed images generated by the same animal and avatar when presented with the movie. Overlaid on the processed images are symbols to indicate the type of action rule being applied.

The animal depicted in Figures 5(b) and 6(c) exhibits no interesting dynamic between its two action rules. It effectively has only one rule because the first is a 'null' rule which does not even influence the other. Figure 6(a) shows that the animal's shooting rule aims to one side of the avatar. Strangely, its settings for colour translation are weighted more towards avatar missiles. By observing the animal in the evaluation environment it was noticed that the animal would actually fire in a predictive fashion. Differential processing enables the animal to fire at incoming avatars and avatar missiles at a point just ahead of and in the path of the target's current position. This did not hold for all conceivable orientations and velocities but predictive firing was generally evident.

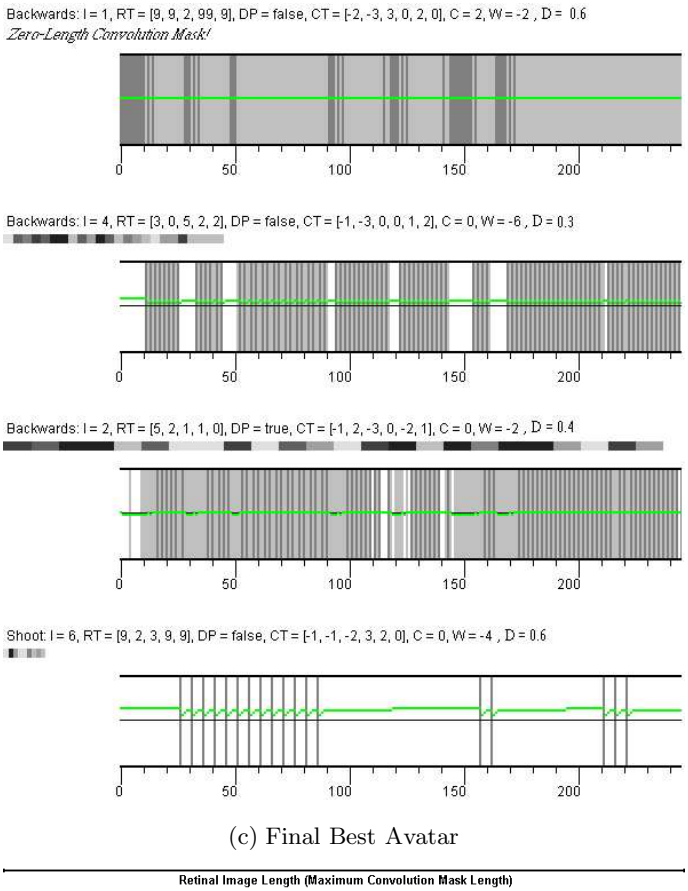
The avatar depicted in figures 5(c) and 6(b) exhibits interesting and complex behaviour. The first action rule has a zero length convolution mask. What this rule does is continue to move the avatar in the direction selected by the last rule. Presumably the advantage of this rule is, in the absence of other sensory input, to keep going in the current direction. In a closed arena this is a simple but effective strategy for covering a lot of space. The second and third action rules are entrained in a 'see-saw' dynamic. Both rules have self-inhibitory connections such that whenever either becomes active it allows the other to take over. Both respond to similar stimuli except that the third rule has a higher response to animals and avatars (animals and avatars present a strong stimulus around frames 50 and 150—see 4(c)). It is believed that the third rule's use of differential processing allows movement detection. The fourth rule, a shooter, has a fairly narrow convolution mask with a high center weight. It translates animals to the high weight of 2, as expected, but also translates avatar missles to the highest weight of 3. This is believed to be a way of externally representing state—if the avatar sees its own missles it will fire at them 'thinking' that it must have had something good to fire at in the first place. Matching the processed images in figure 6(b) with figure 4(c), as well as examining corresponding positions in figure5(c) shows that this avatar indeed tends to shoot directly at animals and avatar missles.



(a) Fitness profile for run KA24. Top trace is peak, bottom is mean.



(b) Final Best Animal



(c) Final Best Avatar

Retinal Image Length (Maximum Convolution Mask Length)

Fig. 5. Each row in (b) and (c) shows: The interpreted action rule; An image of the action rule's convolution mask, with lighter pixels corresponding to higher-valued entrys; A graph of the action rule's activity over the course of the movie. I = importance; RT = range thresholds; DP = differential processing; CT = colour translation; C = connection; W = weight; D = decay. RT array: {obstacle, avatar, avatar missile, animal, animal missile}. CT array: {empty, obstacle, avatar, avatar missle, animal, animal missle}. Rule activity (activation level + importance) is plotted as a light line. Dark grey bands indicate where the rule was active; Light grey, where it could be active; Hatched, where no action rule is active. Horizontal axis corresponds to movie frame number.

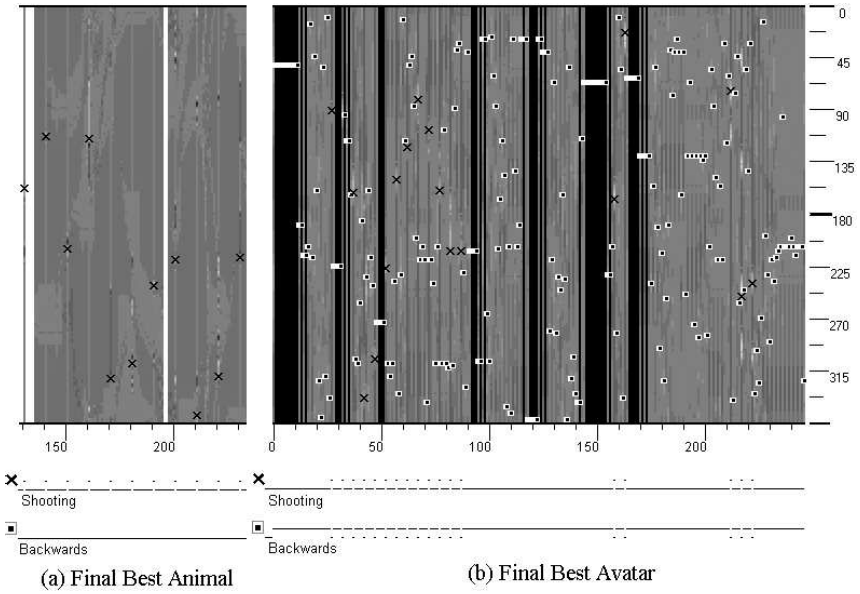


Fig. 6. Processed images of evolved agents presented with the movie. Higher grayscale values correspond to higher image value. Symbols are overlaid upon the processed images to indicate applied action: Black squares outline in white indicates backwards movement; 'X' indicates shooting. See Figure 4(c) for labelling of axes. Note that only a segment of the response for the animal is shown.

Note that all of the various runs conducted exhibited similar interesting examples of behaviour as described above. Also, the single-species runs could be said to be successful in that they showed steadily increasing fitness profiles.

2.2 Future Work

It will be necessary to devise and implement new means to analyze evolved agents. Some form of 'obstacle course' would allow isolated competences of agents to be evaluated objectively. However, even this may not reveal all of the tools and tricks that allow an evolved agent to succeed against its adversaries. An ethological approach of observing the agent in its evaluation environment can reveal more but is more difficult to carry out objectively. Ignoring the issue of mechanisms we could instead focus on raw performance by evaluating the agents in competition with humans and with controllers coded using other methods, such as neural networks and state machines. Again in terms of analysis, more investigation into the specific properties of evolved convolution masks is required (fourier domain analysis). Also, we require additional means to understand the coevolutionary process. Many such methods are described in [3]. Finally, the action rule arbitration scheme can be seen conceptually as being decoupled from the evolved image processing aspects of the system. More investigation into alternate action selection schemes is warranted.

3 Conclusions

This report has detailed the design, motivations, and experimental results for a system that can be used to evolve agents for use in interactive entertainment applications. We have presented a novel visually-based control architecture that was specifically designed for its target application. As well as extending and analysing the current system we hope to explore using some of the same concepts for use in mobile robot navigation. Also, although this has been an engineering project it would be interesting to see how the system presented here could be adapted to serve as a model of visual processing in simple animals. While further investigation is certainly warranted, this work has uncovered an attractive new alternative control mechanism for agents in interactive entertainment.

Acknowledgments This work has been partially supported by NSERC Postgraduate Scholarship B - 232621 - 2000.

References

1. Bassmann, H., Besslich, P., "Ad Oculos", Thompson, 1995.
2. Brooks, R.A. "Intelligence Without Reason". In L. Steels and R. Brooks (Eds.) *The Artificial Life Route to Artificial Intelligence*, Erlbaum, pp. 25-81, 1995.
3. Cliff, D., Miller, G.F., "Tracking the Red Queen", In F. Foran *et. al.* (Eds.) *Third European Conference on Artificial Life*, Springer, pp. 200-218, 1995.
4. Cliff, D., Miller, G.F., "Co-evolution of Pursuit and Evasion II", In P. Maes *et. al.* (Eds.) *SAB4*, MIT Press, Cambridge MA, 1996.
5. Dill, M., Wolf, R., Heisenberg, M., "Visual Pattern Recognition in Drosphilia involves retinotopic matching", Nature Vol. 365, p. 751-753, 1993.
6. Egelhaaf, M., Borst, A. "Motion Computation and Visual Orientation in Flies", Comp. Biochem. Physiol, Vol. 104A, No. 4, p. 659-673, 1993.
7. Floreano, D. "Evolutionary Robotics in Artificial Life and Behavior Engineering", In T. Gomi (Ed.), *Evolutionary Robotics*, Ontario (Canada): AAI Books, 1998.
8. Funes, P., Sklar, E., Juilli, H., Pollack, J. "Animal-Animat Coevolution". In R. Pfeifer *et. al.* (Eds.) *SAB5*. MIT Press. p. 525-533, 1998.
9. Grand, S., Cliff, D. Malhotra, A. "Creatures: Artificial Life Autonomous Software Agents for Home Entertainment", CSRP 434, University of Sussex, 1996.
10. Harris, C., Buxton, B., "Evolving Edge Detectors", Research Note RN/96/3, UCL, Gower Street, London, WC1E 6BT, UK, Jan. 1996.
11. Harvey, I., Husbands, P., Cliff, D., Thompson, A., Jakobi, N. "Evolutionary Robotics", In *Robotics and Autonomous Systems*, v.20, p. 205-224, 1997.
12. Poli, R., "Genetic Programming for Feature Detection and Image Segmentation", In T. Fogarty (Ed.) *AISB'96 Workshop on EC*, Brighton UK, pp. 110-125, 1996.
13. Reynolds, C.W., "Steering Behaviors For Autonomous Characters", In *Proceedings of Game Developers Conference 1999*, Miller Freeman, San Francisco CA, 1999.
14. Vardy, A., "A Visually-Based Evolvable Control Architecture for Agents in Interactive Entertainment Applications", M.Sc. Dissertation, University of Sussex, 2000. (http://www.scs.carleton.ca/~avardy/vardy_msc_diss.pdf)
15. Lamothe, A., "Windows Game Programming for Dummies", IDG Books, 1998.
16. Woodcock, S., "Game AI: The State of the Industry", http://www.gamasutra.com/features/19990820/.